

2. MEMORIA DESCRIPTIVA

2.1 EL SISTEMA GNU/LINUX

2.1.1 Introducción a GNU/Linux

2.1.1.1 ¿Qué es GNU/Linux?

Linux es un sistema operativo compatible **Unix**. Dos características muy peculiares lo diferencian del resto de los sistemas que podemos encontrar en el mercado, la primera: es libre, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo; la segunda: el sistema viene acompañado del código fuente.

El **kernel** (también conocido como **núcleo**) es la parte fundamental del sistema operativo. Es el software responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora, o en forma más básica, es el **encargado de gestionar recursos**. A través de servicios de llamada al sistema. Como hay muchos programas y el acceso al hardware es limitado, el núcleo también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo. Este **núcleo**, escrito casi completamente en **C** con algunas extensiones de GNU C, fue desarrollado por el *hacker* finlandés *Linus Torvalds* en un intento por obtener un sistema operativo libre similar a **Unix** que funcionara con microprocesadores **Intel 80386**.



GNU/Linux es la denominación defendida por *Richard Stallman* y otros para el sistema operativo que utiliza el kernel Linux en conjunto con las aplicaciones de sistema creadas por el proyecto GNU.

Richard Stallman y voluntarios están intentando crear un sistema operativo libre con un funcionamiento similar al UNIX, recreando todos los componentes necesarios para tener un sistema operativo funcional que se convertiría en el sistema operativo GNU. En el comienzo de los años 1990, después de seis años, GNU tenía muchas herramientas importantes listas, como compiladores, depuradores, intérpretes de comando etc, excepto por el componente central: el núcleo. Con el surgimiento del kernel Linux, esta laguna fue llenada y surgió el sistema operativo formado por el kernel Linux en conjunto con las herramientas GNU. De esta manera, Stallman juzga que este sistema operativo es una "versión modificada" del sistema GNU y por lo tanto debe tener la denominación **GNU/Linux**. Esta denominación resolvería la confusión entre el núcleo y el sistema operativo completo a que puede llevar, y de hecho ha llevado, la denominación Linux en solitario. Stallman también espera que con el aporte del nombre GNU, se dé al proyecto GNU que él encabeza el reconocimiento que merece por haber creado las aplicaciones de sistema imprescindibles para ser un sistema operativo compatible con UNIX.

Hay que señalar que, al igual que es una simplificación denominar al sistema que usa el usuario final Linux, obviando las aplicaciones GNU que completan el sistema operativo, el conjunto linux+GNU representa solamente una parte (aunque importante) del software encontrado en una distribución Linux. Existe una gran cantidad de software original del sistema operativo BSD o producido independientemente de los proyectos GNU y Linux por otras personas u organizaciones, como por ejemplo Apache, el X Window System, Samba, GNOME, KDE, OpenOffice.org y miles de otros.

2.1.1.2 Características principales de Linux

Entre las características principales del sistema operativo Linux destacamos:

-Estabilidad: Los sistemas UNIX han sido reconocidos por su estabilidad. Habitualmente se ha usado UNIX en lugares donde la fiabilidad ha sido prioritaria, como en centrales nucleares, control de maquinaria, etc.

-Escalabilidad: La escalabilidad de un sistema operativo es la capacidad de adaptación al hardware sobre el que se ejecuta. Desde sus comienzos, Linux ha sido un sistema operativo con unos requerimientos muy bajos. Esto ha permitido crear servidores en hardware prácticamente obsoleto.

-Flexibilidad: Aunque en sus comienzos se orientó como sistema operativo para servidores, en la actualidad, se pueden realizar en Linux un amplio abanico de tareas alejadas de las funciones de servidor. Entre estas tareas destacamos: programación, ofimática, diseño, multimedia, etc.

-Multitarea: Es la capacidad de un sistema operativo de ejecutar al mismo tiempo varios programas. En la actualidad hay dos tipos de multitarea: una de ellas, multitarea **cooperativa**, muy utilizada en Windows, en la que los programas se ejecutan hasta que deciden que sea otro el que ocupe el procesador. Por el contrario, en Linux existe la multitarea **prioritaria** y en ella hay un proceso del sistema operativo (planificador de procesos) que se encarga de controlar que todos los programas tengan garantizado un tiempo mínimo de ejecución en el procesador.

-Multiusuario: Un sistema operativo multiusuario es aquél que permite que haya distintos usuarios, en un momento determinado, usando la máquina. De esta manera, estos usuarios pueden ejecutar distintos programas o incluso el mismo.

-Multiplataforma: Esta característica permite ejecutar Linux en distintas arquitecturas de procesadores. Las plataformas en las que en un principio se puede utilizar Linux son 386, 486, Pentium, Pentium Pro, Pentium II, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, ARM, MIPS, PowerPC y SPARC.

2.1.1.3 Ambitos de uso

Gracias a las características ofrecidas por Linux, podemos encontrar este sistema en áreas muy distintas del panorama informático, lo que da una idea del desarrollo que se ha obtenido con este sistema operativo.

-Servidores: tradicionalmente ha sido éste el campo en el que Linux ha alcanzado mayor éxito. En pocos años, ha pasado de ser una alternativa económica para pequeños servidores a ser el sistema operativo más robusto, flexible y escalable del mercado, superando ampliamente a otros sistemas operativos comerciales. Así pues, en la actualidad, los servidores web bajo Linux dominan Internet, grandes corporaciones están migrando sus sistemas de información de software propietario a Linux; empresas punteras como IBM, Sun, HP, etc., están enfocando gran parte de sus inversiones en tecnologías relacionadas con Linux y el código abierto.



-Escritorio: el segmento de mercado de los sistemas operativos de escritorio ha estado dominado hegemónicamente por Microsoft durante muchos años. Es precisamente en la orientación al escritorio donde Linux ha progresado, de forma más visible, en los últimos años. El desarrollo de los entornos KDE y GNOME junto con las mejoras implementadas en el sistema gráfico X Window, han permitido canalizar toda la potencia y fiabilidad de Linux a usuarios domésticos de una forma impensable hasta hace bien poco. Actualmente, bajo Linux se pueden desarrollar todas las tareas habituales de un PC de escritorio: ofimática, Internet, juegos, multimedia, etc.

-**Aplicaciones empotradas:** el mercado para pequeños dispositivos como teléfonos móviles, agendas, asistentes personales PDA, puntos de información, etc., es uno de los de mayor crecimiento actualmente. En la opinión de diversos analistas, el mercado para pequeños dispositivos será el más importante en pocos años superando al omnipresente PC de escritorio. La escalabilidad y flexibilidad de Linux, le permite adaptarse a las características limitadas de estos dispositivos en cuanto a memoria, capacidad de almacenamiento y velocidad de proceso. En la actualidad ya existen exitosos proyectos en el mercado como el popular vídeo digital Tivo.



Aplicaciones empotradas

-**Computación paralela:** la computación paralela (clustering) es, sin lugar a dudas, un área de expansión en los próximos años. El clustering es una técnica para incrementar la capacidad de cálculo consistente en usar varios ordenadores independientes, comunicándose a través de una red local de alta velocidad, de tal forma que se comporten, de cara al usuario, como un solo ordenador. Éste campo ha estado tradicionalmente reservado a los grandes centros de investigación. La computación paralela es aplicable a muchos campos, como pueden ser el tratamiento de vídeo, creación de imagen de síntesis, bases de datos paralelas, etc. Actualmente, Linux permite implementar distintos tipos de clustering, algunos de ellos, de forma sencilla y transparente al usuario.



2.1.1.4 Ventajas de Linux sobre otros Sistemas Operativos

Estas son las razones más importantes que hacen que Linux presente ventaja frente a otros sistemas operativos:

1. Software Libre

"Software Libre" es un concepto desarrollado en la década de los 80 por la **Free Software Foundation**. El software libre consiste, en oposición al conocido como "comercial" o "propietario", el software que le da al usuario libertad sobre su uso, modificación y redistribución, con las restricciones mínimas necesarias para garantizar esa libertad a otras personas (es decir, prohíbe que un usuario le restrinja la libertad a otro).

Las ventajas del software libre son muchas. En primer lugar, termina para siempre con los problemas de copyright y legalidad del uso del software. Ya que el software libre no tiene un dueño que guarda en secreto los detalles del software, no existe monopolio sobre un software dado, o su servicio técnico.

Como hemos comentado anteriormente, **GNU/Linux** es el término promovido por la Free Software Foundation (FSF), por su fundador Richard Stallman y por quienes los apoyan para lo que comúnmente se llama "Linux". Esto es, implementaciones del sistema operativo GNU de la FSF que utilizan el núcleo Linux. Por razones históricas y de otro tipo, la mayoría de la gente utiliza el término "Linux" para referirse al sistema completo, siendo una notable excepción **Debian GNU/Linux**.



El principal argumento para llamarlo "GNU/Linux" es que el núcleo de **Linus Torvalds** fue sólo la pequeña parte final de un (de otra manera) sistema completo, GNU, escrito y conformado a lo largo de muchos años con el objetivo explícito de crear un sistema operativo libre e integrado.

2. Estabilidad y eficiencia

Linux se destaca también por su estabilidad. Esto significa que es uno de los sistemas con menos fallos. Las ventajas de esto son obvias, tanto en seguridad de los datos, como en tiempo y costo invertido en resolución de problemas. La eficiencia de Linux también es superior a los otros sistemas operativos en el PC. Esto implica un aumento en la productividad, y una disminución en los costos, ya que pueden realizarse las mismas tareas con equipos menos sofisticados.

3. Basado en UNIX

Linux esta basado en el sistema UNIX. Esto significa que ya hay casi 30 años de experiencia y evolución en sistemas similares. UNIX y sus derivados son los usados en los sistemas de mayor envergadura, como Universidades de todo el mundo, redes gubernamentales y militares. Este sistema fue diseñado para ser capaz de cumplir misiones complejas y de gran escala, no solamente tareas elementales de escritorio. Otra ventaja de ser derivado de UNIX es que es una excelente herramienta de aprendizaje sobre este sistema, tanto para el usuario personal como para las instituciones educativas que quieran instruir sobre UNIX.

4. Usos posibles

Actualmente, Linux ya está en posición de competir con cualquier otro sistema operativo en lo que respecta a usos que pueden dársele. Sin embargo, hay algunas tareas que tienen enormes ventajas si son realizadas bajo UNIX.

En primer lugar, el uso de redes, lo que incluye a Internet. Cualquier usuario de Internet, desde aquel que lee su correo electrónico hasta el que quiere poner su propio proveedor de Internet, dispone de los mejores clientes y servidores de los distintos protocolos de red disponible. El hecho de que Linux es un derivado de UNIX le da más naturalidad en la interacción con Internet (que fue originalmente desarrollada en sistemas UNIX). Pero por redes no solo se entiende la Internet. También hay características avanzadas que permiten usar a Linux para manejar redes locales de forma muy eficiente, con excelentes sistemas de seguridad. Además pueden ponerse estaciones Linux conectadas con estaciones de otros sistemas operativos (Otros UNIXes, Windows NT, etc.).

En segundo lugar, la programación también es una de las tareas en las que se destaca Linux. La popularidad que Linux tiene entre los programadores e ingenieros se debe a las características avanzadas de este sistema que permiten desarrollar software más complejo. En menor medida, las aplicaciones para Linux también son mejores en otras áreas científicas.

El hecho de que Linux se destaque en estos ámbitos, no significa que sea peor en los otros. Existen excelentes interfaces gráficas, procesadores de texto, editores de imágenes, aplicaciones multimedia que funcionan en Linux.

2.1.1.5 Uso de Linux en entornos de Red

Como hemos comentado anteriormente, los sistemas UNIX han sobresalido especialmente en entornos de red. Así pues, numerosos premios entregados por publicaciones especializadas, han reconocido a Linux como el mejor sistema operativo de red del momento.

- **Protocolos de red soportados:** TCP/IP, TCP/IP v6, IPX/SPX, AppleTalk, ISDN (RDSI), T-1, T-3, X.25, Frame-Relay, PPP, SLIP, PLIP, Amateur Radio, ATM, etc

- **Hardware:** Linux dispone de soporte para, prácticamente, todo el hardware de red actual, incluyendo determinados dispositivos obsoletos. El soporte de hardware se extiende hasta los dispositivos para crear las nuevas redes inalámbricas 802.11.

- **Integración con distintos entornos:** Este sistema nos permite la integración con los siguientes entornos:

Entorno Windows: la aplicación **Samba** permite a un sistema Linux participar en una red Windows (grupo de trabajo o dominio) actuando tanto de servidor como de cliente. Como servidor, Samba permite que clientes Windows accedan a ficheros e impresoras tal como lo harían en un servidor Windows NT/2000/XP.

Entorno UNIX: al ser un clónico de UNIX, Linux se integra adecuadamente dentro de una red formada por sistemas UNIX, compartiendo ficheros e impresoras mediante el uso de los protocolos NFS o CODA, entre otros.

Entorno Novell: es posible configurar Linux para actuar dentro de una red Novell tanto de servidor como de cliente.

Entorno Apple: con el paquete "Netatalk", Linux puede proveer servicios de ficheros e impresión en una red local con otros ordenadores Macintosh. En la actualidad, están soportados los protocolos "AppleShare IP" y "classic Appletalk". Con Netatalk, los ordenadores Macintosh pueden acceder a los ficheros y las colas de impresión en el servidor Linux, como si se tratara de un dispositivo Appletalk estándar.



- **Servicios para Internet e Intranet:** Entre otros servicios para Internet/Intranet, dispone de los siguientes:

Correo electrónico: existen distintos servidores de email disponibles para Linux. Entre ellos se encuentran los conocidos servidores "sendmail" y "qmail".

Servidor Web: casi todas las distribuciones incluyen el servidor Apache, que es el servidor web más extendido en Internet. Más del 60% de las páginas que circulan por la Red son entregadas por un servidor Apache.

Servidor FTP: Linux soporta diversos servidores de este popular protocolo de transferencia de ficheros.

Servidor de News: podemos usar programas como "Inn" para crear un completo servidor de Usenet.

Servidor de DNS: existen distintos servidores de DNS para sistemas Linux como "bind" o "djbdns". Estos servidores nos permitirán usar la misma tecnología de resolución de nombres de Internet en nuestra red local.

Groupware: todas las suites de groupware punteras tienen su versión para Linux, actuando tanto de cliente como de servidor. Además de la comunicación con otros sistemas a través de red, Linux puede coexistir en la misma máquina junto a cualquier otro sistema operativo. Es posible acceder, desde Linux, a una enorme variedad de particiones que utilizan otros sistemas operativos como Windows, Solaris y Macintosh.

2.1.2 Distribuciones Linux

2.1.2.1 ¿Qué son las distribuciones?

Linux técnicamente es solamente el Kernel, no el S.O. de soporte. Para poder crear un producto más atractivo, las “**distribuciones**” de Linux son creadas cuando se relaciona el Kernel, manejadores, aplicaciones y muchos otros componentes necesarios para tener el producto final tal y como lo conocemos.

Por tanto, Una distribución es un sistema operativo completo creado por una empresa u organismo que usa una determinada versión del kernel Linux e incluye gran cantidad de software listo para usar.

Algunas compañías ofrecen distintas distribuciones que varían en la funcionalidad, orientación, soporte y cantidad de software que incluyen. Habitualmente, estas empresas permiten la descarga de la distribución desde Internet, mientras que, por otro lado, venden la versión en caja del mismo producto por los canales de comercialización habituales.

Una fuente frecuente de controversia surge debido a que los vendedores de distribución empaquetan código que no es **GPL** (*General Public License*) con el kernel de Linux y lo distribuyen de manera masiva.

En cualquier caso, cualquier persona puede obtener el kernel, complementarlo con algunos programas y crear su propia versión del sistema operativo. Este método, aunque ofrece una libertad total, resulta excesivamente complicado para la gran mayoría de usuarios, y sus ventajas quedan oscurecidas por la enorme dificultad que conlleva.

Las **ventajas** de usar una distribución, en lugar de crearla nosotros mismos, se describen a continuación:

- **Seguridad:** todas las empresas mantienen una actualización continua de los paquetes afectados por vulnerabilidades. Actualizar periódicamente los paquetes inseguros es una de las tareas principales de un administrador de sistemas.

- **Herramientas de configuración:** es norma común entre las distintas distribuciones incluir programas que permiten realizar de forma sencilla las tareas relacionadas con la administración del sistema.

- **Software probado y actualizado:** la enorme cantidad de software disponible para Linux hace complicada la tarea de crear un sistema desde cero sin que aparezcan conflictos entre distintos programas. Una de las prioridades de los creadores de distribuciones es lograr que todo el software funcione sin problemas.

- **Documentación:** además de la documentación sobre Linux existente en forma de libros y sitios web, cada distribución proporciona unos manuales específicos sobre su producto. Esto facilita mucho las tareas de explotación del sistema dado que la información que proporcionan estos manuales es muy específica y fácilmente aplicable.

- **Últimas versiones del software:** la creación de software para Linux sigue un ritmo vertiginoso, con nuevas versiones apareciendo cada pocos meses. Resulta mucho más sencillo el usar los paquetes actualizados que nos proporciona la empresa creadora de nuestra distribución que instalar el software directamente desde las fuentes.

- **Actualizaciones de versiones anteriores:** es habitual, en todas las distribuciones, que se pueda actualizar el sistema actualmente instalado a una versión más moderna sin tener que formatear el disco duro. Esto nos permite tener un sistema completamente actualizado sin tener que eliminar información cada vez que queramos actualizarnos a una nueva versión de nuestra distribución.

Aunque las distribuciones de Linux solucionan muchos problemas, también presentan algunos **inconvenientes** como son los siguientes:

- **Falta de optimización:** aunque Linux se caracteriza por ser un sistema rápido, siempre es posible optimizarlo para lograr mayores prestaciones. Habitualmente, las distribuciones se crean para ser compatibles con la mayor variedad de hardware posible a costa de una pérdida de prestaciones. Por ejemplo, la gran mayoría de las distribuciones se compilan para el procesador 386 de Intel. De esta forma, se aumenta la compatibilidad dado que la mayoría de procesadores actuales son compatibles con el 386. Por otro lado, compilando para procesadores antiguos no se aprovechan las características avanzadas de los micros actuales. Es posible compilar programas de nuestra distribución para aumentar las prestaciones pero, en algunos casos, el proceso puede ser complicado para un usuario inexperto.

- **Incompatibilidades entre distintas distribuciones:** aunque todas las distribuciones parten de la misma base, presentan pequeñas incompatibilidades que hacen que, instalar un paquete de una distribución en otra, pueda resultar problemático.

- **Dificultad para añadir software no preparado para la distribución:** el sistema de gestión de software de la distribución guarda toda la información relevante de cada fichero instalado en el sistema. Esta información es, entre otra, la fecha de la instalación, si ha sido modificado, las dependencias existentes con otros ficheros, etc. Generalmente, el software no nativo de la distribución puede presentar problemas a la hora de instalarse dado que no está bajo la supervisión del sistema de gestión de paquetes.

2.1.2.2 Contenido de las Distribuciones

Aunque la esencia de Linux es el Kernel, se necesita mucho más para convertirlo en un sistema operativo completo. A continuación citamos algunos elementos clave de una típica distribución Linux:

- **Boot manager:** el boot manager o administrador de inicio es una utilidad que reside en el sector de inicio maestro (MBR), que el BIOS de una computadora carga al inicio. Si se agrega Linux a un ordenador con Windows y otro sistema operativo, el administrador de inicio permite seleccionar el sistema operativo que se quiere ejecutar. Aunque hay administradores de inicio de otros fabricantes, Linux incluye su propio administrador de inicio, conocido como GRUB o bien LILO (Linux Loader), que por lo general se instala por omisión.
- **Interfaz de usuario:** En el núcleo, Linux es un derivado de Unix, y su interfaz por omisión es una línea de comandos. Para hacer más accesible a Linux, hay varios entornos de escritorio gráficos que pueden semejarlo al SO Windows o Macintosh. Algunos ejemplos son KDE (Kool Desktop Environment) (que imitó en sus inicios a CDE (Common Desktop Environment)) y Gnome (GNU Network Object Model).
- **Sistema X Window:** al igual que GDI (Graphics Device Interface) de Windows, X window es el subsistema gráfico que soporta una interfaz de usuario gráfica. Usualmente se utiliza la plataforma X.org para sostener interfaces gráficas que es un fork de Xfree86, surgido a raíz del cambio de licencia que este proyecto sufrió en la versión 4.4 y que lo hacía incompatible con la GPL.
- **Servicio de Internet:** Linux tiene soporte nativo para TCP/IP y para todos los protocolos de Internet comunes, como DNS (Domain Name System), HTTP (Hypertext Transfer Protocol), SMTP (Simple Mail Transfer Protocol), FTP (File Transport Protocol), NNTP (Network News Transfer Protocol)...
Gran parte de las de las distribuciones incluyen Apache, el popular servidor web de código público, así como visualizadores web, clientes de correo electrónico, lectores de noticias y otras aplicaciones de Internet.
- **Servicios de impresión y de archivos:** estos servicios permiten a un sistema acceder a los recursos de red y compartir archivos e impresoras con otros usuarios. Entre los servicios típicos se incluyen NFS (Network File System) para compartir archivos con otros sistemas basados en Unix; y Samba, que hace que los sistemas Linux actúen como servidores Windows NT y Appletalk.

- **Aplicaciones:** Linux aún no goza del mismo soporte de aplicaciones que Windows. Pero gran parte de las distribuciones incluyen, literalmente cientos de aplicaciones, y si se desean buscar, se encontrará una buena variedad de herramientas. Además de editores básicos de texto como Crisp y emacs, hay diversas aplicaciones y suites de productividad. GIMP (GNU Image Manipulation Program) es un clon de Adobe Photoshop.
- **Administradores de Paquetes:** Los administradores de paquetes son herramientas diseñadas para instalar, desinstalar y actualizar aplicaciones en sistemas Linux. El *Debian Package Management System* y el *Red Hat Package Manager* (RPM) son administradores de paquetes bien conocidos.
- **Herramientas del Programador:** puesto que Linux es un favorito de los programadores, no sorprende que las distribuciones incluyan muchas herramientas de programación. Entre éstas se incluyen algunas para administrar y crear programas a partir de código fuente, depuradores como gdb, lenguajes de programación y lenguajes de scripting como Phyton y TCL.
- **Bibliotecas:** Linux incluye un conjunto central de rutinas de bibliotecas que son módulos ejecutables que realizan funciones que soportan a otras aplicaciones. Glibc es la biblioteca estándar de C GNU. En los sistemas Linux se instala con el nombre de libcX (siendo X el número de versión).



2.1.2.3 Las distribuciones más importantes

Como resultado del código abierto de Linux y de su libre distribución un buen número de compañías y organizaciones han desarrollado su propia distribución de este sistema operativo. Cada una de estas distribuciones tiene sus propias características, que las hacen más adecuadas para determinados tipos de sistemas informáticos.

DEBIAN

Es una de las distribuciones más antiguas (1993). El proyecto Debian es una comunidad de desarrolladores que han hecho causa común para crear un sistema operativo libre y sin ataduras comerciales. Esto tiene como contrapartida que las versiones de Debian suelen ser lentas en aparecer, ya que al no haber detrás ninguna compañía comercial, no se sacan versiones hasta que no se han comprobado exhaustivamente todos los posibles fallos. Destacan de esta distribución su probada estabilidad, la excelente documentación y la enorme cantidad de aplicaciones que incluye.



KNOPPIX

Esta distribución basada en Debian, aporta la novedad de poder ser ejecutada directamente desde el CD (Live CD), de manera que los usuarios noveles pueden probar un sistema Linux sin necesidad de instalarlo en su ordenador. Posteriormente, pueden hacerlo si lo desean. Además, se caracteriza por detectar automáticamente casi todo el hardware que tengamos en nuestro equipo. Por otra parte, incorpora la última versión del escritorio KDE y OpenOffice.org.



GUADALINEX

Este es el CD Live creado por la Junta de Andalucía para su uso en dicha comunidad autónoma. Está basada en Debian y Linex (distribución Live CD de la Junta de Extremadura), con un sistema de instalación fácil de usar que incluye QtParted, un interfaz gráfico para crear particiones. Al igual que Knoppix, si posteriormente deseamos instalarlo en nuestro disco duro, podemos hacer uso del icono que a tal efecto aparece en el escritorio Gnome de esta distribución.



RED HAT

Red Hat es actualmente el líder mundial en desarrollo, instalación y gestión de Linux y soluciones de código abierto para aplicaciones que cubren desde sistemas empotrados hasta servidores web seguros. Dispone de innumerables herramientas que facilitan la configuración de todos los servicios y las posibilidades del sistema. A partir de la versión 9, Red Hat decidió separar el producto desarrollando la distribución para uso empresarial (Red Hat Enterprise Edition) y colaborando en otra para uso doméstico, Fedora Core, junto con la comunidad GNU.



MANDRAKE

Mandrake Linux es una distribución creada en 1998, basada en Red Hat, con el objetivo de hacer Linux más fácil de usar para todo el mundo. La empresa creadora, MandrakeSoft, integró los entornos gráficos más potentes junto con unas herramientas propias de configuración y pronto alcanzó fama por establecer el estándar en facilidad de uso e instalación. Actualmente, MandrakeSoft ofrece todo el poder de Linux destinado tanto a usuarios principiantes, profesionales y servidores.



SUSE

SuSE Linux es una distribución altamente profesional dirigida principalmente al mercado empresarial. Destaca de esta distribución, sus herramientas propietarias de configuración que proporcionan a los administradores de sistemas un control enorme bajo una interfaz muy sencilla.



SLACKWARE

La distribución Slackware Linux es la más veterana de las existentes ya que es la primera que tuvo repercusión internacional. Es una distribución muy actualizada y que ofrece, desde su creación, una instalación y un entorno de trabajo basado en la sencillez. Apenas dispone de herramientas de configuración y todo su mantenimiento debe hacerse al nivel más bajo, es decir, editando los ficheros de configuración de las distintas aplicaciones. Es la distribución más parecida al sistema UNIX original.



GENTOO

Gentoo Linux es una distribución rápida, versátil y abierta enfocada a desarrolladores, profesionales y usuarios avanzados. Gentoo utiliza un sistema de gestión de software, heredado de los sistemas UNIX BSD, denominado "ports". A diferencia de otras distribuciones, Gentoo no ofrece software ya compilado y listo para instalar sino que, el proceso de instalación de nuevo software consiste en descargar el paquete de Internet y compilarlo siguiendo unas reglas que hayamos definido en nuestro sistema. Todo el proceso de instalación de nuevos paquetes está supervisado por un avanzado sistema de gestión denominado "Portage".



2.1.2.4 Comparativa de algunas distribuciones Linux

A continuación mostramos una tabla comparativa extraída de *Wikipedia.org* donde se muestran las distribuciones más importantes, las empresas u organizaciones que hay detrás, la fecha de la primera distribución y otros datos de interés.

	Empresa	Fecha de la primer P.R.	Predecesor	Ultima versión estable	Precio (€)	Licencia	Público	País
Debian GNU/Linux	Proyecto Debian	Agosto de 1993	N/A	4.0 (Etch)	Gratis	Cualquier DFSG	Desktop, workstation, server	Mundial
Fedora Core	Proyecto Fedora	Noviembre de 2003	Fedora Linux, Red Hat Linux	Ver 7. Junio 2007	Gratis	GPL	Workstation, server, público	EEUU
Gentoo	Fundación Gentoo	Marzo de 2002	Enoch	2006.1	Gratis	GPL	Workstation, Server, público	Mundial
Mandriva Linux	Mandriva	Julio de 1998	Mandrake Linux, Conectiva	2007.1 Spring	Gratis	GPL	Desktop, Workstation, server	Mundial
Rxart	Pixart	Octubre de 2001	Rxart Linux	3.0	16	GPL	Workstation, server	Argentina
Slackware Linux	Patrick Volkerding	Julio de 1993	SLS	11.0	Gratis	GPL	Workstation, Server, público	EEUU
SUSE Linux	Novell, Opensuse	Marzo de 1994	Jurix	10.3	Descarga gratuita	GPL	Workstation, Server, Desktop, Público	Mundial
Ubuntu	Canonical Ltd.	Octubre de 2004	Debian	7.04 (Feisty Fawn)	Gratis LiveCD	GPL	Desktop, Workstation, Server	Mundial

2.1.2.5 Distribución para nuestro proyecto: Debian ¿Por qué Debian?

Para la realización de este proyecto, optamos por el uso de la distribución **Debian** ya que no hay detrás de ella una compañía comercial. Es una de las distribuciones más libres que existen por la manera en que esta organizada su comunidad. Con más de 900 desarrolladores oficiales, y más de 18.000 paquetes, Debian es uno de los **más grandes proyectos** de Software Libre en existencia hoy.

Los usuarios de Debian deciden utilizarla por su **calidad**, está entre las distribuciones GNU/Linux más seguras, estables y robustas. Además, es recomendable conocerla, ya que las distribuciones LIVE CD la usan como base.

Los desarrolladores de Debian no incorporan un paquete hasta que no esté infinitamente probado, al no tener exigencias comerciales, esto hace de Debian una distribución diferente.

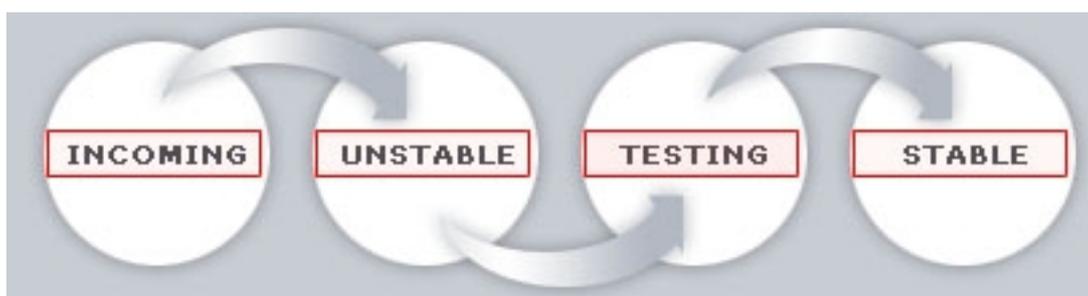
Una característica importante de la distribución Debian es su útil **sistema de paquetes** (DEB) para facilitar la instalación, desinstalación y actualización del software. Un paquete es un conjunto de componentes compilados, necesarios para hacer que un programa funcione. Mediante el sistema de paquetes, otros desarrolladores pueden contribuir con software propio a la distribución. A día de hoy, con casi 18.000 paquetes disponibles y habiendo sido portada con éxito a otras plataformas, la distribución Debian es por el momento la distribución más grande.

Como desventaja podemos citar que es difícil de instalar, pues la instalación se realiza en modo texto a diferencia de otras distribuciones con interfaces gráficos más amigables y automatizados, esto implica conocer a fondo la máquina donde se instala, esa dificultad se recompensa en personalización, flexibilidad y configuración óptima del sistema. Sin embargo, esto ocurría con la versión estable (Sarge) en el momento de ejecución de este proyecto, pues actualmente a la finalización del mismo la versión Etch de Debian ya dispone de interfaz gráfico durante la instalación del sistema.



2.1.2.6 Modelo de Desarrollo de Debian: Estable, Prueba, Inestable

Ya hemos comentado antes que la comunidad Debian no publica una nueva versión hasta que no ha comprobado exhaustivamente los errores que puedan presentarse. Esto hace que la sucesión de versiones no sea muy rápida, como en otras distribuciones.



Los nombres de las versiones de Debian son tomados de la película Toy Story. Hasta la fecha ha habido ocho versiones. Las versiones principales son las siguientes:

En todo momento, se dispone de una versión **stable**, que es la recomendada para uso en servidores, ya que el software que se utiliza en esta versión ha sido comprobado exhaustivamente en busca de errores. Para uso en una estación de trabajo no sería del todo recomendable ya que los programas que incorpora están obsoletos en su mayor parte. Al comienzo de este proyecto, la versión estable es la **3.1 Sarge** (Desde el 6 de junio de 2005); si bien a fecha de hoy es la **4.0 Etch** (Desde el 8 de abril de 2007).

A continuación, tendríamos la que sería la siguiente versión a publicar como stable, cuando se hayan solucionado gran parte de los errores encontrados. Se denomina **testing** (la actual se llama **lenny**) y contiene software más actualizado que la versión estable. Esta versión sería más idónea para una estación de trabajo.

Por último encontramos la versión **unstable** (la cuál siempre recibe el nombre de **Sid**), en la que se encuentran las últimas versiones de los programas. Por esta razón, puede incorporar muchos fallos. Aunque no debe preocuparse, ya que el modelo de desarrollo de Linux hace que en cuestión de horas se hayan solucionado.

El proceso que sigue cualquier software desde que su autor lo crea hasta que llega a la distribución estable es el siguiente:

- a. Cuando un colaborador “sube” un programa al sitio web de la distribución, éste es almacenado en el directorio incoming.
- b. Una vez al día se trasladan de aquí a la distribución unstable y permanecen allí, estando disponibles para pruebas e incluso para ser descargados por los usuarios.
- c. Ahí permanecerán durante cierto tiempo hasta que se decida su traslado a la rama testing. Así se va construyendo la distribución testing, que será la próxima versión estable.
- d. Cuando la distribución de prueba testing está suficientemente madura, se “congela” (distribución frozen): no se aceptan programas nuevos y los desarrolladores sólo dedican su tiempo a corregir fallos.
- e. Una vez que los errores están por debajo de un máximo exigible, la distribución congelada se transforma en la nueva distribución estable, pasando la anterior estable a obsoleta.

2.1.3 Instalación de la versión estable de Debian

2.1.3.1 Obtener Debian GNU/Linux desde Internet

Debian proporciona muchas opciones antes y durante el proceso de instalación. Es posible hacer instalaciones a través de una red, con o sin cd-rom, variando el número de particiones, etc...

Si poseemos una versión oficial de Debian Linux, dispondremos de la versión impresa de la "Guía de instalación", pero si hemos obtenido nuestra distribución descargándola de Internet o a través de otros medios, podremos consultar la guía de instalación junto a otros documentos en la siguiente dirección: <http://www.debian.org/releases/stable/installmanual>.

Debian soporta 11 arquitecturas distintas (Intel x86, Motorota, etc). Nosotros nos centraremos en la instalación sobre una arquitectura basada en Intel x86. De todas formas, podemos consultar <http://www.debian.org/ports/> para más detalles.

En nuestro caso particular, como nuestro PC dispone de conexión a Internet y como nos interesa tener un sistema operativo estable con las aplicaciones mínimas necesarias, utilizaremos "la instalación en red". Esta instalación consiste en que descargaremos una única imagen ISO de algo menos de 180 Mb que posteriormente grabaremos en un CD y que contiene los datos mínimos para iniciar la instalación.

La descarga la realizaremos desde la siguiente dirección de Internet:

<http://www.debian.org/CD/netinst/#netinst-stable>

Dicho CD es autoarrancable y posee el software necesario para comenzar la instalación de Debian, así como un sistema operativo muy básico, pudiendo descargar el resto de paquetes que queramos desde Internet.

Podemos consultar el manual de instalación de Debian en cualquier momento en la siguiente url:

<http://www.debian.org/releases/stable/i386/index.html.es>

2.1.3.2 Pasos previos a la instalación de Debian

- **Requisitos mínimos de memoria y espacio en disco:** Debemos tener al menos 32MB de memoria y 110MB espacio de disco duro. Para un sistema mínimo basado en consola (todos los paquetes estándar), se requiere 250 MB. Si queremos instalar una cantidad razonable de software, incluyendo el sistema de ventanas X, y algunos programas y bibliotecas de desarrollo, necesitaremos al menos 400 MB. Para una instalación más o menos completa de un sistema de escritorio, se necesitarán entre 1 y 4 gigabytes.

- **Copia de seguridad:** En el proceso de instalación no es preciso eliminar la información existente de otros sistemas operativos. En cualquier caso, es una buena costumbre siempre que se vaya a realizar una acción potencialmente peligrosa como instalar un nuevo sistema operativo, realizar una copia de seguridad de la información del disco duro.

Los métodos para efectuar una copia de seguridad dependen del sistema operativo que se esté utilizando y de los medios físicos de los que dispongamos.

En nuestro caso, no realizaremos copia de seguridad alguna, ya que vamos a tener una máquina dedicada, y suponemos que carecemos de información relevante en el disco antes de la instalación.

- **Dispositivo de arranque:** El dispositivo de arranque permite a Debian comenzar el proceso de instalación. Los dispositivos más habituales serán el propio CD autoarrancable de Debian o un disquete de arranque.

Si nuestro ordenador permite el arranque desde CD-ROM, probablemente debamos modificar, en la BIOS, el orden de arranque de los dispositivos. Dependiendo del tipo y fabricante de la BIOS, el procedimiento para modificar el orden de arranque puede variar. Por lo tanto, se recomienda consultar el manual de la placa base para averiguar cuáles son los pasos a seguir.

- **Método de Instalación:** Para realizar la copia de archivos, el programa instalador necesita localizar en qué soporte se encuentran. Entre los soportes permitidos por Debian encontramos:
 - CD-Rom: necesita un ordenador que permita arrancar desde la unidad de CD-Rom o ejecutamos el programa `\install\boot.bat` contenido en el CD.
 - Hard Drive: los datos se leen desde el disco duro, por lo que las imágenes ISO de los CDs deben estar alojadas en él. Se necesitará un disquete de arranque.
 - Imagen NFS: necesita tener alojadas las imágenes ISO de los CDs de instalación de Debian en un servidor NFS.
 - FTP: el proceso es similar a la instalación desde un servidor NFS cambiando únicamente el tipo de servidor.
 - http: similar a la instalación desde FTP únicamente cambiando el protocolo por el que se reciben los datos.

- **Particionamiento del disco duro:** Las particiones nos permiten dividir el espacio disponible en un disco duro con el fin de instalar varios sistemas operativos u organizar la información dentro de un mismo sistema operativo. En una arquitectura PC existen los siguientes tipos de particiones:
 - Particiones primarias: permiten almacenar sistemas operativos. Pueden existir cuatro particiones primarias en cada disco duro.
 - Particiones extendidas: Una partición extendida es un contenedor donde podemos crear otras particiones. No son utilizables por sí solas. Solamente puede existir una partición extendida en cada disco duro.
 - Particiones lógicas: son las que podremos crear dentro de la partición extendida.

- **Partición swap:** Consiste en utilizar una parte del disco duro para almacenar los fragmentos de la memoria RAM que se utilicen con menor frecuencia. Esto permite a la máquina soportar un uso de la memoria elevado durante algunos instantes.

Para su instalación, Linux requiere del uso de al menos una partición de swap que, para un equipo destinado para uso personal, será suficiente definirla con un tamaño de 128MB.

Este esquema de funcionamiento tiene un inconveniente claro y es que el disco duro es mucho más lento que la memoria principal.

- **Denominación de discos y particiones:** La nomenclatura en sistemas Linux consiste en identificar cada disco y cada partición dentro de éste por una cadena que sigue el siguiente formato:

- En primer lugar, aparece la cadena "hd" para discos IDE y "sd" para discos SCSI, SATA ó USB.
- A continuación, una letra (a,b,c,d, etc.) identifica el número de disco.
- Finalmente, un número identifica a la partición.

La tercera partición del segundo disco duro IDE será:

hdb3

La séptima partición del tercer disco duro SCSI o SATA será:

sdс7

2.1.3.3 Pasos a seguir durante la instalación de Debian

- **Arranque del sistema de instalación:** Una vez que hemos grabado la imagen del CD de instalación y habilitamos la BIOS de nuestro ordenador para que ejecute “*arranque desde CD*” como primera opción, insertamos nuestro CD y reiniciamos la máquina.

- **Métodos y parámetros de arranque del instalador:** A continuación, aparecerá la pantalla de bienvenida de Debian en la que encontramos atajos de teclado para obtener ayuda. En la última línea de pantalla, aparece la palabra “boot” y un cursor parpadeante, esperando alguna orden. En este momento, la instalación acepta “**parámetros de arranque**”, que son los parámetros del núcleo de Linux que generalmente se utilizan para asegurar la correcta gestión de los periféricos. Si por cualquier causa, Linux no autodetectara el hardware de nuestro equipo, podríamos especificar explícitamente mediante estos parámetros el tipo y la ubicación de nuestro hardware para que el sistema lo reconociera. Aunque en la mayoría de los casos el núcleo puede auto-detectar toda la información necesaria sobre los periféricos.

Algunos de los parámetros más interesantes que reconoce el sistema de instalación son los siguientes:

debconf/priority: El valor de este parámetro define la prioridad de los mensajes que se mostrarán durante la instalación. No se mostrará ningún mensaje de menor prioridad a la aquí definida. Usaremos **debconf/priority=low**, en este caso, se mostrarán todos los mensajes (esto es equivalente al método de arranque *experto*).

DEBIAN_FRONTEND: Este parámetro de arranque controla el tipo de interfaz de usuario que utilizará el instalador. Utilizaremos la interfaz predeterminada, que es **DEBIAN_FRONTEND=newt**

BOOT_DEBUG: Si se establece este parámetro a dos se conseguirá que el proceso de arranque del instalador genere registros más detallados. Si se establece a tres se obtendrá un intérprete de línea de órdenes en puntos estratégicos del proceso de arranque.

netcfg/disable_dhcp: Por omisión, el debian-installer configura automáticamente la conexión de red a través de DHCP. No se podrán revisar ni cambiar los valores obtenidos de esta forma, si la prueba tiene éxito. Sin embargo, podremos realizar la configuración de forma manual en el caso de que falle la solicitud DHCP.

Por otro lado, hay distintos “**métodos de arranque**” disponibles. Ellos son:

linux: arranca la instalación, es el método por defecto.

expert: arranca la instalación en modo experto, para tener un control máximo.

linux26: arranca la instalación utilizando una versión 2.6 del Kernel.

expert26: arranca la instalación en modo experto utilizando una versión 2.6 del Kernel.

Utilizaremos expert26 para comenzar la instalación en modo experto con un núcleo estable de **versión 2.6**. Al iniciar el instalador en modo “expert” para hacer funcionar o detectar el hardware es necesario indicar opciones a los módulos del núcleo conforme se instalen. El modo experto da control total del debian-installer.

-**Funcionamiento del instalador:** el instalador de Debian está compuesto por un conjunto de componentes de propósito específico para realizar cada tarea de la instalación. Cada componente realiza una tarea, formulando al usuario las preguntas que sean necesarias para realizar su trabajo. Se asignan prioridades a cada una de las preguntas, fijando su prioridad al arrancar el instalador.

Cuando se realiza una instalación estándar, solamente se formularán las preguntas esenciales (prioridad alta). Esto tiene como consecuencia un proceso de instalación altamente automatizado y con poca interacción del usuario. Los componentes son ejecutados automáticamente en una secuencia predeterminada. Los componentes a ejecutar dependerán del método de instalación que use y de su hardware. El instalador usará los valores predeterminados para las preguntas que no son formuladas. Cuando exista un problema, veremos el error en pantalla, y es posible que se muestre el menú del instalador para que elijamos de éste alguna acción alternativa.

Las pantallas del instalador están basadas en caracteres. El ratón no está operativo en este entorno. Las teclas con la flecha **arriba** y **abajo** mueven entre los distintos elementos disponibles en una lista desplazable, y también desplazan a la lista en sí. Con la **barra espaciadora** se marcan elementos, como en el caso de una casilla. Y pulsando **Enter** se activan las opciones elegidas.

Los mensajes de error son redireccionados a la tercera consola. Se puede acceder a ésta pulsando **Alt Izq-F3**. También puede encontrar los mensajes de error en `/var/log/messages`. Este registro se copia a `/var/log/debian-installer/messages` en el nuevo sistema una vez finalizada la instalación. Durante el proceso de instalación se puede encontrar otros mensajes en `/var/log/`, y en `/var/log/debian-installer/` después de que el ordenador haya sido iniciado con el sistema instalado.

A continuación, se realizan los siguientes pasos:

1. Elección del idioma para la instalación, país y configuración de la distribución del teclado.
2. Detección del hardware para cargar los módulos adecuados.
3. Cargar componentes del instalador desde el cd.
4. Detección del hardware de red y configuración de la misma.
5. Particionado del disco.
6. Instalación del sistema base.
7. Instalación del cargador de arranque GRUB en disco duro.
8. Terminar la instalación

-Consideraciones adicionales sobre el particionado del disco: Como mínimo, Debian necesita una partición que contiene el sistema de ficheros raíz (cuyo punto de montaje es `/`). La mayor parte de las personas también consideran imprescindible una partición de intercambio separada. Como ya hemos comentado con anterioridad, el intercambio (swap) es una zona de disco que el sistema operativo utiliza como almacenamiento para memoria virtual.

Durante la instalación, se nos pregunta si queremos realizar una única partición (para sistemas de escritorio), o tener un sistema con varias particiones (espacial para sistemas multiusuarios y servidores). De esta forma, al tener varias particiones separadas para los distintos directorios de nuestro sistema, la corrupción de una de ellas no destruye el sistema completo.

En nuestro caso particular, para dotar de mayor seguridad, elegiremos dicha opción, creándonos un sistema con varias particiones. Nos ha quedado lo siguiente:

Tenemos un disco duro de 4.3 GB y las particiones han quedado:		
- Partición primaria:	205.6MB	ext3 montada en /
- Partición lógica:	230.3 MB	intercambio
- Partición lógica:	1.6 GB	ext3 montada en /usr
- Partición lógica:	1.5 GB	ext3 montada en /home
- Partición lógica:	674.4 MB	ext3 montada en /var
- Partición lógica:	123.3 MB	ext3 montada en /tmp

- Configuración del sistema base Debian: Si todo ha ido bien al reiniciar el sistema, nos encontraremos con una ventana de bienvenida. El sistema está ya instalado, pero necesita una configuración base como fecha y hora, creación usuarios e instalación de programas.

Seguiremos los siguientes pasos:

1. Pre-configurar los parámetros relacionados con el idioma.
2. Configurar zona horaria.
3. Configurar usuarios y contraseñas.
4. Establecer el nombre del sistema.
5. Configurar apt.
6. Seleccionar e instalar paquetes.

Podremos volver a ejecutar este programa de configuración en cualquier momento mediante **base-config**.

- **Gestión de usuarios y grupos:** Como comentamos anteriormente, Linux es un sistema operativo multiusuario, y debido a esta característica todos los accesos al sistema se deben realizar usando una cuenta de usuario. Cuando se crea una cuenta de usuario el administrador le asigna un identificador UID que es un número que la máquina utiliza internamente para identificar a los usuarios. Habrá que decidir qué usuarios podrán acceder al sistema, de qué forma y bajo qué permisos.

Nuestro dispositivo de comunicaciones, en principio, es una máquina que permite el intercambio de información entre estaciones de trabajo en una red. Por lo tanto, en esta configuración no vamos a tener usuarios que accedan a nuestro dispositivo para ejecutar programas o comandos. Es por ello que sólo vamos a permitir el acceso al sistema a un único usuario para que éste pueda realizar tareas de configuración, mantenimiento y monitorización de la máquina. Dicho usuario es el administrador de la red.

Para tal función utilizaremos la cuenta de **root**. La cuenta root es la cuenta del usuario que tiene privilegios especiales para poder administrar el sistema a través de él. La utilización de la cuenta root puede ser peligrosa, dado que proporciona poder sin límites, cualquier error en un comando puede tener consecuencias catastróficas. Cuando queramos ejecutar algún comando complejo destructivo debemos probarlo primero de forma no destructiva. Sólo se debería utilizar "root" para las tareas que impliquen la administración del sistema. Por tanto, crearemos además una cuenta de usuario que el administrador de la red utilizará para trabajar en un modo funcional básico que le permitirá realizar tareas simples como la monitorización y obtención de datos y logs. Con esta medida de seguridad reduciremos la probabilidad de que se puedan producir errores de cambios importantes en la configuración del sistema.

- **Apt:** comentaremos el uso de la herramienta apt con sumo detalle en el siguiente apartado.

- **Entrada al sistema:** Una vez realizados todos estos pasos ya tendremos el sistema operativo instalado y listo para trabajar. Para entrar al sistema debemos escribir el nombre del usuario válido creado durante la instalación o bien escribir "root" si queremos acceder como administrador del sistema. A continuación se nos pedirá la contraseña de dicho usuario y, si es correcta, ya estaremos dentro del sistema. Al proceso de identificación ante el sistema se le conoce como **login**.

Además, si hemos iniciado sesión en el sistema bajo un usuario normal, podemos pasar a ser "root" usando el comando "su". Una vez introducido, el sistema nos preguntará la contraseña de root, y si es correcta ya podremos usar sus privilegios. Observaremos cómo el mensaje que presenta el sistema mientras espera órdenes (prompt) cambia del símbolo "\$" a "#".

Para volver a ser usuario, podemos introducir la orden "exit" o pulsar las teclas Control + D. De nuevo cambiará el prompt para reflejar la pérdida de privilegios.



2.1.3.4 Instalación y gestión de paquetes en Debian

Las tareas de administración de software han sido tradicionalmente muy complejas y propensas a fallos. Hace unos años, el administrador de un sistema Linux obtenía los programas en código fuente y debía instalarlos a mano, cuidando de que no se produjese un conflicto con alguna aplicación ya instalada. Aunque la instalación de software desde las fuentes es adecuada en algunas circunstancias, no es una práctica cómoda y puede resultar compleja para usuarios sin muchos conocimientos.

Los usuarios tenían que compilar cada programa que quisieran usar en su sistema GNU/Linux. El software se distribuía como ficheros de extensión **.tar.gz**, llamados *tarballs*. Son archivos empaquetados con tar y comprimidos con gzip. Se pueden descomprimir con "gzip -d fichero.tar.gz" o desempaquetar con "tar -xzf archivo.tar.gz".

Cuando Debian fue creado, fue imperante que el sistema incluyera un programa que se encargara de manejar la paquetería instalada en la computadora. Con la creación de los sistemas de paquetes RPM y DEB, la situación ha cambiado radicalmente. Un **paquete** es un archivo que contiene una o varias aplicaciones junto con información relativa a esas aplicaciones. Actualmente, la instalación de una

aplicación puede ser tan sencilla como seleccionarla en un menú, mientras que un programa automáticamente la obtiene de un CD o Internet, comprueba las dependencias y la instala en nuestro sistema. Habitualmente, usaremos aplicaciones preparadas para nuestra distribución, y es que una de las ventajas de utilizar una distribución conocida es la gran cantidad de software listo para instalar que proporciona. Además, todo este software ha sido probado de tal forma que funcione correctamente instalado en la misma máquina.

Es muy común en Linux que una aplicación dependa de otras, ya sea mediante la línea de comandos o utilizando librerías. Esta relación provoca, especialmente en aplicaciones grandes, que instalar un programa nos obligue a instalar otros muchos de los que depende. Este problema se agrava cuando la aplicación no sólo depende de otras, sino que también depende de una determinada versión de otra aplicación. El problema de las dependencias entre aplicaciones ha sido, en parte, solucionado con el uso de paquetes.

Los paquetes de software utilizados por la distribución Debian se reconocen por la extensión **.deb**. Para la gestión de los paquetes en Debian existen dos tipos de herramientas:

- **De bajo nivel:** El único programa que encontramos aquí es **dpkg**, el instalador de paquetes individuales .deb. Su único objetivo es instalar, actualizar, eliminar un solo paquete. El aspecto negativo de esta herramienta es que no resuelve dependencias: si intentamos instalar un programa que depende de otros, no lo instalará.

- **De alto nivel:** En esta categoría encontramos más programas, algunos de ellos para X Window. Estos gestores sí resuelven las dependencias, de tal manera que si intentamos instalar un programa que dependa de otros, primero instalará los otros y luego el que nosotros pedimos, automáticamente. Ejemplos de este tipo de programas:

- **Apt (Advanced Packaging Tool)**
- **Dselect**
- **Aptitude**
- **Synaptic**

Apt, dselect y aptitude son programas de texto para la consola mientras que Synaptic funciona bajo X Window.

Centrémosnos en el uso de la herramienta **apt**. Para poder manipular paquetes mediante APT necesitamos conocer el nombre de dicho paquete. Para esto podemos usar los comandos apt-cache que buscan en la base de datos de paquetes disponibles.

Los comandos más usados de este tipo son:

Comando	Función
apt-cache search <nombre>	Busca en la lista de paquetes disponibles para instalar, aquellos que están relacionados con <nombre>
apt-cache show <paquete>	Muestra información sobre el paquete especificado.

Una vez que ya sabemos qué queremos instalar, podemos usar los comandos apt-get. No tenemos que preocuparnos de problemas de dependencias ya que apt es lo suficientemente inteligente como para gestionarlas.

Comando	Función
apt-get install <paquete1> <paquete2> ...	Instala los paquetes cuyo nombre le indiquemos
apt-get remove <paquete1> <paquete2>	Borra los paquetes que le indicamos. No elimina los ficheros de configuración.
apt-get --purge remove <paquete1> <paquete2>..	Borra el paquete completamente, incluyendo los ficheros de configuración.

Comando	Función
apt-get update	Utiliza las fuentes definidas en sources.list para buscar paquetes nuevos.
apt-get upgrade	Actualiza aquellos programas de los que existen nuevas versiones.
apt-get dist-upgrade	Nos permite actualizarnos de distribución (estable, inestable, testing)

Normalmente apt buscará los paquetes en los repositorios que tenemos especificados en el archivo de repositorios de paquetes y programas. Los repositorios no son otra cosa que depósitos donde están almacenados todos los programas, binarios y códigos fuente. Estos depósitos están alojados en servidores de Internet a los que podemos conectarnos y descargarnos el software de manera automática utilizando apt.

El archivo `/etc/apt/sources.list` contiene las entradas de los repositorios en los que busca apt. Escribiremos las siguientes líneas para obtener los paquetes del repositorio de la versión estable de Debian:

```
deb http://ftp.debian.org/debian/ stable main
#deb-src http://ftp.debian.org/debian/ stable main
```

Las entradas que empiezan con deb-src son para obtener el código fuente de los programas. Normalmente no la utilizaremos y la comentaremos precediéndola del carácter '#', si necesitáramos descargar el código fuente de algún programa lo descomentaríamos.

Acto seguido actualizamos la lista de paquetes:

```
apt-get update
```

Y con esto ya tenemos el sistema listo para instalar paquetes del repositorio oficial de Debian utilizando `apt-get install`.

Como ya comentamos en los objetivos del proyecto, nuestro dispositivo debería ser lo más compacto posible; esto es que, en la medida de lo posible, instalaremos únicamente los paquetes necesarios con el fin de optimizar al máximo los recursos de la máquina. Para llevar a cabo tal objetivo sería conveniente ir eliminando aquellos paquetes que no vamos a necesitar a la vez que vamos construyendo nuestro dispositivo. Vamos a introducir algunas herramientas que nos ayudarán a conseguir esta misión.

En primer lugar, podemos ver la totalidad de los paquetes que tenemos instalados mediante:

```
dpkg --get-selections
```

Para eliminar por completo un paquete, incluso sus ficheros de configuración, lo haremos con:

```
dpkg --purge <paquete>
```

El problema que encontramos es que a priori es difícil saber de la lista de paquetes cuáles no vamos a necesitar o bien cuáles de ellos no tienen dependencias con otros paquetes. Por ejemplo, podríamos eliminar un paquete del cual dependiesen otros que fueran parte de una aplicación imprescindible para nuestro dispositivo de red. Para evitar problemas de este tipo utilizaremos la siguiente herramienta:

```
apt-get install deborphan
```

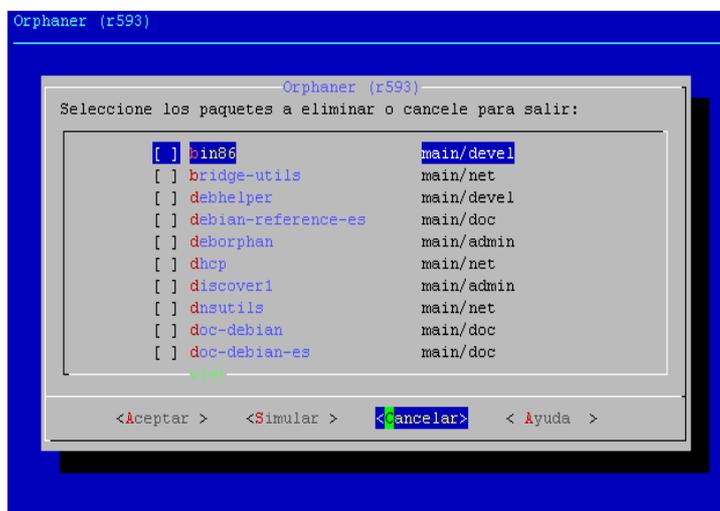
Deborphan lista únicamente aquellos paquetes que a su vez, no tienen otros que dependan de ellos.

```
deborphan
```

De esta manera obtenemos una lista de paquetes que si sabemos con seguridad que no necesitamos podríamos eliminar sin afectar a otras herramientas. Deborphán muestra por defecto sólo aquellos paquetes correspondientes a librerías (archivos lib*); sin embargo, con la opción -a muestra todos los que no tengan dependencias o “huérfanos”.

Orphaner es una interfaz más amigable para deborphán. Desde ella podemos directamente seleccionar de la lista los paquetes que deseamos eliminar.

orphaner



También podemos obtener información sobre la funcionalidad de cada uno presionando la opción de Ayuda.

A medida que vamos eliminando paquetes sin dependencias, la lista que generan estas herramientas se actualiza con nuevos que dependían exclusivamente de los que acabamos de eliminar. Si a su vez vamos eliminando más paquetes, van apareciendo otros que estaban por debajo; así vamos “podando nuestro árbol” de paquetes.

En la siguiente tabla mostramos cómo nos queda el espacio utilizado en disco tras realizar una instalación limpia y mínima del sistema base (en esta prueba es la Debian etch), antes de realizar ninguna eliminación de paquetes:

Sistema ficheros	Usados	Montado en
/dev/sda1	76M	/
tmpfs	-	intercambio
/dev/sda9	18M	/home
/dev/sda8	5.6M	/tmp
/dev/sda5	143M	/usr
/dev/sda6	66M	/var

Este es un buen momento para desinstalar los programas que no usaremos. La herramienta *aptitude* es un frontend de apt en el que podemos ver de una manera amigable la lista de paquetes que tenemos instalados en nuestro sistema separados en distintas categorías, esto nos permitirá evaluar si no vamos a necesitar determinados paquetes y los eliminaremos con el objetivo de optimizar el espacio en disco y hacer más simple la administración y mantenimiento del sistema.

Para ejecutar la interfaz basta hacer:

aptitude

A continuación presentamos una captura de la misma:

```
Acciones  Deshacer  Paquete  Solucionador  Buscar  Opciones  Vistas  Ayuda
C-T: Menú ? : Ayuda q : Salir u : Actualizar g : Descarga/Instala/Elimina Paqs
aptitude 0.4.4
--- Actualizaciones de seguridad
--\ Paquetes instalados
--- admin - Utilidades de administración (instalación de programas, gestión de
--- devel - Utilidades y programas para desarrollo de programas
--- doc - Documentación y programas especializados para ver documentación
--- editors - Editores y procesadores de texto
--- libs - Colección de rutinas de programas
--- misc - Programas varios
--- net - Programas para conectarse y proporcionar varios servicios
--- oldlibs - Bibliotecas obsoletas

Los paquetes en la sección «devel» se usan para escribir programas y trabajar
sobre programas ya existentes. Los que no sean programadores y no compilen sus
propios programas probablemente no necesiten muchos programas de esta sección.

Incluye compiladores, herramientas de depuración, editores para programadores,
herramientas para procesar código fuente y otras cosas relacionadas con el
desarrollo de programas.
```

En esta instalación particular mínima del sistema base que hemos realizado, aparentemente no hay muchos paquetes que veamos de manera obvia que no vayamos a utilizar posteriormente. Proseguiremos el proceso con la herramienta `deborphan` comentada anteriormente y eliminaremos los paquetes que no tengan dependencias con una línea como la siguiente:

```
deborphan | xargs apt-get remove -y
```

Donde `xargs` lee la salida de `deborphan` y se los pasa como argumentos a `apt-get remove`.

Una vez llegados aquí deberíamos tener nuestro dispositivo lo más compacto posible; a partir de este momento ya tenemos nuestro sistema operativo Debian preparado para ir instalando y configurando únicamente las herramientas de red con las que vamos a trabajar.

2.1.4 El Kernel Linux

2.1.4.1 Numeración y Versiones del Kernel

El Kernel es el software que se sitúa entre el hardware y los procesos de nivel de usuario. De él dependen las características, fiabilidad y estabilidad del conjunto. Las distintas distribuciones de Linux se diferencian en el software que proporcionan ya que el Kernel es común.

El Kernel sigue una estricta numeración entre las distintas versiones. El patrón seguido actualmente por la numeración de las versiones del Kernel de Linux es el siguiente:

A.B.C[D] (Ej.: 2.2.1, 2.4.13 ó 2.6.12.3)

El número **A** denota la versión del kernel. Es el que cambia con menor frecuencia y sólo lo hace cuando se produce un gran cambio en el código o en el concepto del kernel. Históricamente sólo ha sido modificado dos veces: en 1994 (versión 1.0) y en 1996 (versión 2.0).

El número **B** denota la mayor revisión del kernel. Antes de la serie de Linux 2.6.X, los números pares indicaban la versión "estable" lanzada. Por ejemplo, una para uso de fabricación, como el 1.2, 2.4 ó 2.6. Los números impares, en cambio, como la serie 2.5.X, son versiones de desarrollo, es decir que no son consideradas de producción. En los kernels de desarrollo se prueban características, drivers y funciones que, más tarde, pasan a los kernels estables.



Comenzando con la serie Linux 2.6.x, no hay gran diferencia entre los números pares o impares con respecto a las nuevas herramientas desarrolladas en la misma serie del kernel. Linux Torvalds dictaminó que éste será el modelo en el futuro.

El número **C** indica una versión menor en el kernel. En la forma anterior de versiones con tres números, esto fue cambiado cuando se implementaron en el kernel los parches de seguridad, bugfixes, nuevas características o drivers. Con la nueva política, sólo es cambiado cuando se introducen nuevos drivers o características; los cambios menores se reflejan en el número **D**.

El número **D** se produjo cuando un grave error, que requería de un arreglo inmediato, se encontró en el código NFS de la versión 2.6.8. Sin embargo, no habían otros cambios como para lanzar una nueva revisión (la cual hubiera sido 2.6.9). Entonces se lanzó la versión 2.6.8.1 con el error arreglado como único cambio. Con 2.6.11 esto fue adoptado como la nueva política de versiones. Bug-fixes y parches de seguridad son actualmente manejados por el cuarto número dejando los cambios mayores para el número **C**.

También, algunas veces después de las versiones puede haber algunas letras como "rc1" ó "mm2". El "rc" se refiere a release candidate e indica un lanzamiento no oficial. Otras letras usualmente (pero no siempre) hacen referencia a las iniciales de la persona. Esto indica una bifurcación en el desarrollo del kernel realizado por esa persona, por ejemplo ck se refiere a Con Kolivas, ac a Alan Cox, mientras que mm se refiere a Andrew Morton.

En la dirección web www.kernel.org podemos encontrar las últimas versiones de Kernel disponibles.

2.1.4.2 Obtención de un kernel específico para nuestro dispositivo

La estructura del sistema operativo GNU/Linux funciona de tal manera que el kernel puede ser reemplazado por otro de diferente versión sin ningún tipo de problemas. En dicho Kernel está especificado el Hardware que poseemos, desde el procesador o procesadores que tenemos hasta la tarjeta de sonido, de red, sistema de ficheros soportados, protocolos de conexión a Internet, etc.

Al instalar nuestro sistema Debian, tenemos un kernel pre-configurado y pre-compilado, lo cual puede llevarnos a que algún componente de nuestro hardware que poseamos no funcionase o no lo hiciera correctamente, además de tener una sobrecarga de módulos inútiles para nosotros.

En esta sección vamos a aprender a configurar y compilar nuestra propia imagen del kernel para hacerlo a nuestra justa medida.

En primer lugar nos descargaremos de www.kernel.org la última versión estable, que en el momento que fue escrito este documento era la 2.6.22 y las guardaremos en `/usr/src/`.

```
cd /usr/src/  
wget ftp://ftp.kernel.org/pub/linux/kernel/v2.6/linux-2.6.22.tar.gz
```

Y lo descomprimiremos y desempaquetaremos con:

```
tar xvzf linux-2.6.22.tar.gz
```

Borramos el archivo tar.gz, entramos dentro del directorio creado y ya estamos preparados para dar el siguiente paso.

Como dijimos anteriormente, una de las razones por la cual compilar el kernel es para ajustarlo y optimizarlo a nuestro hardware. Esto se hace por medio de la configuración, un proceso con una gran cantidad de opciones. Antes de comenzar, es necesario tomar en cuenta lo siguiente: muchas de las características propias del kernel pueden ser compiladas dentro del mismo o como módulos. La diferencia radica principalmente en que, mientras más modularizado sea nuestro kernel, más pequeño será el tamaño de la imagen, lo que por ende nos lleva a una mejor utilización de la memoria.

Por otro lado, la utilización de módulos tiene otra gran ventaja: un kernel dinámico hace más fácil la tarea de agregar o eliminar hardware, puesto que solamente es necesario compilar e instalar o eliminar el módulo correspondiente, respectivamente, y no reconfigurar todas las opciones del sistema y compilar un nuevo kernel. Además, los módulos son cargados en memoria y eliminados de ella en demanda, haciendo nuevamente un manejo más eficiente de los recursos.

La configuración del kernel se lleva a cabo a través de un programa interactivo que muestra todas las posibles opciones. En este sentido, existen tres posibilidades: un sencillo programa de consola que consta de una serie de preguntas (no recomendado, muy tedioso y no da la posibilidad de retroceder), otro es un programa al igual de consola pero con una interfaz basada en ncurses lo que lo hace mucho más amigable para el usuario, y por último un programa similar pero para X11 (no recomendado puesto que puede presentar problemas de estabilidad). Nosotros utilizaremos la interfaz basada en ncurses y la ejecutaremos de la siguiente manera:

```
make menuconfig
```

Habiendo necesitado para ello instalar los paquetes *build-essential* y *libncurses5-dev*. La interfaz tiene un aspecto como muestra la siguiente captura:

```
.config - Linux Kernel v2.6.22 Configuration  
-----  
Linux Kernel Configuration  
Arrow keys navigate the menu. <Enter> selects submenus ---.  
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,  
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>  
for Search. Legend: [*] built-in [ ] excluded <M> module < >  
-----  
[*] Code maturity level options --->  
|   General setup --->  
|   Loadable module support --->  
|   Block layer --->  
|   Processor type and features --->  
|   Power management options (ACPI, APM) --->  
|   Bus options (PCI, PCMCIA, EISA, MCA, ISA) --->  
|   Executable file formats --->  
|   Networking --->  
|   Device Drivers --->  
|   File systems --->  
+-----+  
|   <Select>  <Exit>  <Help>  |  
+-----+
```

Desde aquí damos soporte al hardware, a los protocolos, etc. Más tarde, esta configuración será compilada en una imagen con la que iniciaremos el sistema. La leyenda de teclas para la utilización del menuconfig es como sigue:

Pulsando “enter” entramos en el menú o submenús.

Pulsando “Y” incluimos lo que seleccionemos como parte de la imagen del Kernel.

Pulsando “M” lo incluiremos como módulo, esto es, se compila aparte y se cargará si queremos o no y no ocupará espacio en la imagen del kernel.

Pulsando “N” lo excluirémos de la futura imagen.

A continuación mostramos los distintos apartados que se nos presentan en el menú de configuración y comentaremos algunas modificaciones que vamos a ir haciendo:

Sección	Comentarios
Code maturity level options	Al seleccionar esta opción se nos preguntará durante el proceso de configuración si queremos incluir en nuestro kernel características muy nuevas y/o experimentales, incrementando la cantidad de posibilidades. Como nuestro objetivo es tener una máquina estable, no activaremos esta opción.
General setup	Opciones generales de configuración: activaremos soporte para redes, hardware PCI, entre otras.
Loadable module support	Opciones para hacer uso de módulos.
Processor type and features	En esta sección se indica el tipo de procesador y sus características. Seleccionaremos la familia de la CPU (586) y otros parámetros como la cantidad de memoria física, emulación matemática, MTRR, SMP, etc...
Power Management Options	Algunas opciones sobre ahorro de energía.
Networking support	Opciones de red: soporte para firewall, IP aliasing, masquerading, entre otros. Eliminaremos soporte para todos aquellos protocolos y tecnologías que no sean IP que no vamos a utilizar (ATM, Appletalk, IPX, Bluetooth, infrarrojos, etc...)
Device Drivers	En esta sección especificaremos si tenemos o no dispositivos SCSI, IDE, SATA, unidades FLOPPY, puertos USB, unidades de CD-Rom, soporte para RDSI. Desactivaremos todo aquello que no necesitemos. Como también soporte para dispositivos de red que no sean Ethernet PCI (Token Ring, USB Network Adapters , PCMCIA, SLIP, PPP, WAN devices, etc...)
File Systems	Como estamos usando EXT3, activaremos sólo soporte para este tipo de ficheros. Deshabilitando otros como NFS, JFS, así como compatibilidades con ficheros de windows, pues no usaremos aplicaciones como Samba.
Kernel Hacking	Esta opción es generalmente para desarrolladores de drivers o administradores que buscan problemas en el kernel.
Security options	Opciones de Selinux, que son implementaciones de directivas de seguridad en el kernel de Linux.
Cryptographic options	Soporte para multitud de protocolos de encriptación.

Una vez finalizada la configuración del kernel (y estando seguros de que todo está como corresponde), debemos salir del programa y guardar los cambios.

Hecho esto, procederemos a generar las dependencias y limpiar los objetos residentes anteriores que puedan andar por ahí:

```
make dep && make clean
```

A continuación tecleamos:

```
make zImage
```

Este es el proceso más largo, la compilación del kernel. Esto creará una imagen del kernel compilada. Si al finalizar este proceso obtenemos un mensaje diciendo que el sistema es demasiado grande tenemos la posibilidad de compilar la imagen en un formato comprimido reemplazando el comando anterior por:

```
make bzImage
```

Esto creará la imagen compilada comprimida en bzip2, por lo cual necesitaremos tener instalado bzip2

Si todo ha ido bien, tendremos nuestra imagen del kernel bajo el directorio `/usr/src/linux-2.6.22/arch/i386/boot/` llamada `bzImage`. Ahora nos disponemos a compilar e instalar los módulos de nuestro nuevo kernel:

```
make modules && make modules_install
```

Por último, instalamos de forma definitiva el nuevo kernel en el sistema mediante:

```
make install
```

Que es equivalente a copiar los archivos compilados en un lugar donde las pueda encontrar nuestro gestor de arranque (GRUB):

```
cp /usr/src/linux-2.6.22/arch/i386/boot/bzImage /boot/vmlinuz-2.6.22
cp /usr/src/linux-2.6.22/System.map /boot/System.map-2.6.22
```

GRUB (GRand Unified Bootloader) es el primer programa que se ejecuta al arrancar un ordenador. Él tiene la tarea de cargar y de transferir el control al núcleo del sistema operativo. A continuación configuraremos GRUB para arrancar con la nueva imagen del kernel. GRUB posee una interfaz de menú desde la que podemos seleccionar una entrada para elegir entre un sistema operativo o un kernel de linux con el que arrancar. Para modificar este menú tendremos que editar el archivo de `/boot/menu.lst` e insertar las líneas:

```
title      Debian GNU/Linux 2.6.22
root       (hd0,0)
kernel     /boot/vmlinuz-2.6.22 root = /dev/hda1
```

Donde `title` es el nombre que aparecerá en el menú.

`root (<tipo-dispositivo><numero-dispositivo>,<particion>)` - Configura la partición raíz para GRUB, tal como `(hd0,0)` y monta la partición.

`kernel <ruta kernel> <opcion-1>...<opcion-N>` Especifica el archivo del kernel a cargar cuando se cargue el sistema operativo. La opción `root=/dev/hda1` especifica el dispositivo en el que se ubica la partición `root` para el sistema.

A continuación, reiniciamos nuestra máquina:

```
reboot
```

Y ya podemos elegir cargar el nuevo kernel compilado. Si en este proceso obtuviéramos errores de tipo *kernel panic*, deberíamos volver a la configuración del kernel, refinar la selección de módulos y volver a repetir el proceso. Si por el contrario todo ha ido bien y no nos ha aparecido ningún mensaje de error, ya tendremos listo nuestro dispositivo para empezar a trabajar.

