



Manual Gentoo Linux 2005.1 x86

Gentoo Foundation, Inc.

Este es el Manual Gentoo, un esfuerzo para centralizar la información acerca de Gentoo/Linux.

Contents:

1. Instalando Gentoo	5
2. Trabajando con Gentoo	91
3. Trabajando con Portage	133
4. Configuración de Redes en Gentoo	159

Authors

[Sven Vermeulen](mailto:swift@gentoo.org) [swift@gentoo.org]

Autor

[Daniel Robbins](mailto:drobbins@gentoo.org) [drobbins@gentoo.org]

Autor

[Jerry Alexandratos](mailto:jerry@gentoo.org) [jerry@gentoo.org]

Autor

[Tavis Ormandy](mailto:taviso@gentoo.org) [taviso@gentoo.org]

Desarrollador Gentoo Alpha

[Guy Martin](mailto:gmssoft@gentoo.org) [gmssoft@gentoo.org]

Desarrollador Gentoo HPPA

[Joe Kallar](mailto:blademan@gentoo.org) [blademan@gentoo.org]

Desarrollador Gentoo SPARC

Pierre-Henri Jondot

Editor

[Rajiv Manglani](mailto:rajiv@gentoo.org) [rajiv@gentoo.org]

Editor

[Stoyan Zhekov](mailto:zhware@gentoo.org) [zhware@gentoo.org]

Editor

[Colin Morey](mailto:peitolm@gentoo.org) [peitolm@gentoo.org]

Editor

[Carl Anderson](mailto:carl@gentoo.org) [carl@gentoo.org]

Editor

[Zack Gilburd](mailto:klasikahl@gentoo.org) [klasikahl@gentoo.org]

Editor

[Benny Chuang](mailto:bennyc@gentoo.org) [bennyc@gentoo.org]

Editor

[Joshua Kinard](mailto:kumba@gentoo.org) [kumba@gentoo.org]

Editor

[Grant Goodyear](mailto:g2boojum@gentoo.org) [g2boojum@gentoo.org]

Revisor

[Donnie Berkholz](mailto:spyderous@gentoo.org) [spyderous@gentoo.org]

Revisor

[Lars Weiler](mailto:pylon@gentoo.org) [pylon@gentoo.org]

Contribuidor

[John Christian Stoddart](mailto:chiquire@gentoo.org) [chiquire@gentoo.org]

Editor-Es

[Roy Marples](mailto:uberlord@gentoo.org) [uberlord@gentoo.org]

Autor

[Chris Houser](mailto:chouser@gentoo.org) [chouser@gentoo.org]

Autor

[Seemant Kulleen](mailto:seemant@gentoo.org) [seemant@gentoo.org]

Desarrollador Gentoo x86

[Jason Huebel](mailto:jhuebel@gentoo.org) [jhuebel@gentoo.org]

Desarrollador Gentoo AMD64

[Pieter Van den Abeele](mailto:pvdabeele@gentoo.org) [pvdabeele@gentoo.org]

Desarrollador Gentoo PPC

[John P. Davis](mailto:zhen@gentoo.org) [zhen@gentoo.org]

Editor

[Eric Stockbridge](mailto:stocke2@gentoo.org) [stocke2@gentoo.org]

Editor

[Jungmin Seo](mailto:seo@gentoo.org) [seo@gentoo.org]

Editor

[Jared Hudson](mailto:jhhudso@gentoo.org) [jhhudso@gentoo.org]

Editor

[Jorge Paulo](mailto:peesh@gentoo.org) [peesh@gentoo.org]

Editor

[Jon Portnoy](mailto:avenj@gentoo.org) [avenj@gentoo.org]

Editor

[Jack Morgan](mailto:jmorgan@gentoo.org) [jmorgan@gentoo.org]

Editor

[Erwin](mailto:erwin@gentoo.org) [erwin@gentoo.org]

Editor

[Tobias Scherbaum](mailto:dertobi123@gentoo.org) [dertobi123@gentoo.org]

Editor

[Gerald J. Normandin Jr.](mailto:gerrynjr@gentoo.org) [gerrynjr@gentoo.org]

Revisor

[Ken Nowack](mailto:antifa@gentoo.org) [antifa@gentoo.org]

Revisor

[José Alberto Suárez López](mailto:bass@gentoo.org) [bass@gentoo.org]

Editor-Es

[José Luis Rivero](mailto:chiquire@gentoo.org) [chiquire@gentoo.org]

Editor Adjunto-Es

Part 1

Instalando Gentoo

En esta parte aprenderá cómo instalar Gentoo en su sistema.

Contents:

1.1.	Acerca de la instalación Gentoo Linux	7
1.1.1.	Introducción	7
1.1.2.	Instalación rápida utilizando la Plataforma de Referencia Gentoo	10
1.2.	Arrancando desde el CD Universal de instalación	12
1.2.1.	Requisitos Hardware	12
1.2.2.	El CD de instalación Universal de Gentoo	12
1.2.3.	Descargar, Grabar y Arrancar un CD de instalación de Gentoo	14
1.3.	Configuración de su red	20
1.3.1.	¿Necesita conexión a la red?	20
1.3.2.	Auto Detección de Red	21
1.3.3.	Configuración Automática de Red	22
1.3.4.	Configuración Manual de la Red	24
1.4.	Preparando los discos	31
1.4.1.	Introducción a Dispositivos de Bloque	31
1.4.2.	Diseñando un Esquema de Particionamiento	32
1.4.3.	Utilizando fdisk para particionar su disco	34
1.4.4.	Creación de Sistemas de Ficheros	39
1.4.5.	Montaje	41
1.5.	Instalando los archivos de instalación Gentoo	43
1.5.1.	Instalando el Stage comprimido (tarball)	43
1.5.2.	Por defecto: utilizar un stage3 del LiveCD	44
1.5.3.	Alternativa: utilizando un stage procedente de Internet	45
1.5.4.	Instalando Portage	46
1.5.5.	Configurando las opciones de compilación	47
1.6.	Instalando el sistema base de Gentoo	51
1.6.1.	Chrooting	51
1.6.2.	Configurando la variable USE	52

1.7.	Configurando el kernel	54
1.7.1.	Zona Horaria	54
1.7.2.	Instalar las Fuentes	54
1.7.3.	Predeterminado: Configuración Manual	55
1.7.4.	Alternativa: Usar genkernel	59
1.7.5.	Configurar los módulos del kernel	60
1.8.	Configurando su sistema	62
1.8.1.	Información del Sistema de Ficheros	62
1.8.2.	Información de red	64
1.8.3.	Información del sistema	68
1.9.	Instalando herramientas necesarias para el sistema	71
1.9.1.	Bitácora del Sistema	71
1.9.2.	Opcional: Demonio Cron	72
1.9.3.	Opcional: Indexar Archivos	72
1.9.4.	Herramientas del Sistema de Archivos	73
1.9.5.	Herramientas de Red	73
1.10.	Configurando el gestor de arranque	75
1.10.1.	Realizando su elección	75
1.10.2.	Por defecto: Usando GRUB	76
1.10.3.	Alternativa: Utilizando LILO	81
1.10.4.	Reiniciando el Sistema	84
1.11.	Finalizando su instalación Gentoo	85
1.11.1.	Administración del Usuario	85
1.11.2.	Opcional: Instalar Paquetes PRG	86
1.12.	¿Y ahora qué?	88
1.12.1.	Documentación	88
1.12.2.	Gentoo en línea	88
1.12.3.	Cambios en Gentoo desde 2005.1	89

Acerca de la instalación Gentoo Linux

“Los usuarios que no estén familiarizados con Gentoo no siempre saben que Gentoo representa la posibilidad de escoger.”

1: Introducción

Primero de todo, *bienvenido/a* a Gentoo. Está a punto de entrar en un mundo de flexibilidad y rendimiento. Gentoo es la flexibilidad en sí. Cuando instalas Gentoo, esto queda claro varias veces, puedes elegir cuánto quieres compilar tu mismo, cómo instalar Gentoo, que gestor de registro prefieres, etc.

Gentoo es una metadistribución moderna, rápida, con un diseño limpio y flexible. Gentoo está hecha alrededor del software libre y no oculta a sus usuarios qué hay bajo la alfombra. Portage, el sistema de mantenimiento de paquetes que usa Gentoo, está escrito en Python, por lo que el código fuente es fácil de visualizar y modificar. El sistema de paquetes de Gentoo se basa en el código fuente (aunque también soporta paquetes precompilados) y para configurar Gentoo se utilizan archivos de texto plano. En otras palabras, abierto a cualquiera.

Es muy importante que entienda que la *flexibilidad* es lo que hace que Gentoo funcione. Intentamos no forzarle a entrar en algo que no le guste. Si cree en algún momento que lo estamos haciendo, por favor, [envíe su opinión](http://bugs.gentoo.org) [http://bugs.gentoo.org].

. . . : ¿Cómo está estructurada la instalación?

La instalación de Gentoo puede verse como un procedimiento de 10 pasos, los correspondientes a los capítulos 2 a 11. Cada paso da como resultado un cierto estado:

- Tras el paso 1, te encontrarás en un entorno funcional preparado para instalar Gentoo
- Después del paso 2, la conexión a Internet estará funcionando y lista para instalar Gentoo.
- Tras el paso 3, sus discos duros estarán preparados para alojar tu instalación de Gentoo
- Tras el paso 4, el entorno de instalación estará preparado y se encontrará

dentro de un entorno chroot.

- Después del paso 5, los paquetes principales, que son los mismos en toda instalación de Gentoo, estarán instalados
- Tras el paso 6, el kernel Linux estará compilado.
- Después del paso 7, la mayoría de los archivos de configuración de tu sistema Gentoo estarán preparados
- Tras el paso 8, las herramientas de sistema necesarios (podrá elegirlos de una hermosa lista) están instaladas.
- Al finalizar el paso 9, el gestor de arranque elegido estará instalado y configurado y usted está dentro de su nueva instalación de Gentoo.
- Tras el paso 10, se encontrará dentro de su nueva Gentoo.

Cuando se le pide una elección especial, intentamos explicar lo mejor posible los pros y contras. También propondremos una opción por defecto, identificada con "Por defecto: " en el título. Las otras posibilidades se titulan "Alternativa: ". Pero *no* crea que la opción por defecto es la que recomendamos. Es la que pensamos que la mayoría de usuarios van a utilizar.

Algunas veces puedes seguir un paso opcional. Estos pasos son marcados como "Opcional: " y no son necesarios para instalar Gentoo. Sin embargo, algunos pasos opcionales dependen de una decisión tomada previamente. Le informaremos cuando se dé el caso, tanto cuando tome la decisión, como cuando se describa el paso opcional.

... : ¿De qué opciones dispongo?

Puedes instalar Gentoo de diferentes formas. Puedes descargar e instalar uno de nuestros CDs de instalación, desde otra distribución, desde un CD de arranque (como Knoppix), desde un arranque por red, desde un disquete de arranque, etc.

Este documento cubre la instalación utilizando el CD de instalación universal, un CD autoarrancable que contiene todo lo que necesita para conseguir instalar y ejecutar Gentoo Linux. De manera opcional, se puede utilizar nuestro CD de paquetes para instalar un sistema completo en cuestión de minutos después de haber instalado el sistema base de Gentoo.

Si necesita ayuda para otros tipos de instalación, por favor lea nuestra [Guía de Instalación Alternativa](http://www.gentoo.org/doc/es/altinstall.xml) [http://www.gentoo.org/doc/es/altinstall.xml]. También tenemos el documento [Trucos y consejos de Instalación de Gentoo \(en inglés\)](http://www.gentoo.org/doc/es/gentoo-x86-tipsntricks.xml) [http://www.gentoo.org/doc/es/gentoo-x86-tipsntricks.xml] que podría resultar útil. Si nota que las instrucciones de instalación son demasiado detalladas, no dude en utilizar nuestra Referencia Rápida para Instalación la cual se encuentra entre nuestros [recursos de documentación oficiales](http://www.gentoo.org/doc/es/index.xml) [http://www.gentoo.org/doc/es/index.xml] si su arquitectura tiene este documento disponible.

También existen otras posibilidades: puede compilar el sistema completo desde el principio o utilizar un entorno pre-compilado para tener el sistema listo en poco tiempo. Y, por supuesto, también hay soluciones intermedias con las cuales no se compila todo el sistema pero se empieza desde un sistema bastante completo.

...: ¿Problemas?

Si tienes algún problema con la instalación (o con el documento de instalación), por favor, compruebe el error en desde nuestro [Proyecto de ingeniería de versiones Gentoo](http://www.gentoo.org/proj/en/releing/) [http://www.gentoo.org/proj/en/releing/], visite nuestro [Sistema de seguimiento de errores](http://bugs.gentoo.org) [http://bugs.gentoo.org] y compruebe si el error es conocido. Si no lo es, por favor cree un informe sobre él para que podamos encargarnos de él. No tenga miedo de los desarrolladores que están asignados a sus informes, normalmente no se comen a nadie.

Acuérdese que, a pesar de que el documento que está leyendo es específico de la arquitectura, esté también contiene referencias a otras arquitecturas. Esto es así porque el manual de Gentoo tiene partes extensas de código que es común para todas las arquitecturas (para evitar duplicar esfuerzos y el desgaste de los recursos de desarrollo). Intentaremos reducir esto al mínimo para evitar la confusión.

Si no estás seguro que el problema es de usuario (algún error que ha cometido al despistarte y no leer la documentación cuidadosamente) o un problema de software (algún error que hemos cometido despistándonos al probar la instalación y/o documentación) eres libre de entrar en #gentoo-es en irc.freenode.net. Por supuesto, eres bienvenido de todas formas :)

Si no se está seguro si el problema es un problema del propio usuario (algún error que se ha cometido por no leer con cuidado la documentación) o un problema con el software (un error que hemos cometido por no probar la documentación/instalación con cuidado), tómese la libertad de entrar #gentoo-es en la red irc.freenode.net. Por supuesto, siempre será bienvenido de todas formas

:)

Si tiene cualquier pregunta concerniente a Gentoo, eche un vistazo a las [Preguntas de Uso Frecuente](http://www.gentoo.org/doc/es/faq.xml) [http://www.gentoo.org/doc/es/faq.xml], disponibles en la [Documentación de Gentoo](http://www.gentoo.org/doc/es/) [http://www.gentoo.org/doc/es/]. También puede mirar los [FAQs](http://forums.gentoo.org/viewforum.php?f=40) [http://forums.gentoo.org/viewforum.php?f=40] en nuestros [foros](http://forums.gentoo.org) [http://forums.gentoo.org]. Si no encuentras la respuesta aquí, pregunta en #gentoo-es, nuestro canal IRC en irc.freenode.net. Sí, algunos de nosotros somos frikis que aún usan el IRC :-)

2: Instalación rápida utilizando la Plataforma de Referencia Gentoo

La Plataforma de Referencia de Gentoo, de ahora en adelante la conoceremos por su abreviatura inglesa GRP (Gentoo Reference Platform), es una imagen de los paquetes precompilados que los usuarios (esto le implica a usted) pueden instalar durante la instalación de Gentoo para acelerar el proceso de instalación. La GRP contiene todos los paquetes necesarios para tener una instalación de Gentoo completamente funcional. No son todos los necesarios que hacen falta para tener una instalación base preparada en nada de tiempo, pero todas las compilaciones grandes (como puedan ser KDE, xorg-x11, GNOME, OpenOffice, Mozilla, ...) están disponibles como paquetes GRP.

Sin embargo, estos paquetes precompilados no son mantenidos durante la toda la vida de la distribución de Gentoo. Son simplemente imágenes creadas con cada distribución de Gentoo y hacen posible tener un sistema funcional en poco tiempo. Puede actualizar el sistema en segundo plano mientras trabaja con su entorno Gentoo.

. . . : Cómo maneja Portage los Paquetes GRP

Su árbol de Portage - la colección de *ebuilds* (los archivos que contienen toda la información sobre un paquete, como es su descripción, página oficial, las URLs de código fuente, instrucciones de compilación, dependencias, etc.) - debe sincronizarse con los GRP: las versiones de los ebuilds disponibles y su correspondiente paquete GRP deben coincidir.

Por esta razón solamente se puede beneficiar de los paquetes GRP que Gentoo suministra mientras se está llevando a cabo la instalación actual. GRP no está

disponible para los interesados en llevar a cabo una instalación utilizando las últimas versiones de los paquetes disponibles.

... : ¿Puedo utilizar GRP?

No se proporcionan paquetes GRP para todas las arquitecturas. Esto no significa que GRP no esté soportado por el resto de arquitecturas, pero significa que no disponemos de todos los recursos necesarios para compilar y probar los paquetes GRP.

Por el momento proporcionamos paquetes GRP para las siguientes arquitecturas:

- La arquitectura **x86** (x86, pentium3, pentium4, athlon-xp) Nota: Los paquetes GRP de x86 (por ejemplo packages-x86-2005.0.iso) están disponibles en nuestros servidores espejo, mientras que pentium3, pentium4, athlon-xp están sólo disponibles a través de bittorrent.
- La arquitectura **amd64** (amd64)
- La arquitectura **sparc** (sparc64)
- La arquitectura **ppc** (G3, G4, G5)
- La arquitectura **alpha**

Si su arquitectura (o subarquitectura) no está en la lista, no podrá utilizar la opción GRP durante la instalación.

Ahora que ha terminado la introducción, continuemos con [Arrancando el CD de instalación Universal](#).

Arrancando desde el CD Universal de instalación

“Utilizando nuestro CD de instalación Universal puede arrancar su sistema desde un entorno de ejecución que le permite instalar Gentoo.”

1: Requisitos Hardware

Antes de empezar, enumeraremos los requerimientos de hardware necesarios para instalar con éxito Gentoo en su máquina.

. . . : Requisitos Hardware

CPU	i486 o superior
Memoria	64 MB
Espacio en disco	1.5 GB (excluyendo el espacio de intercambio (swap))
Espacio de intercambio	Al menos 256 MB

2: El CD de instalación Universal de Gentoo

Gentoo Linux puede instalarse usando uno de los tres archivos *stage*. Un archivo *stage* (fase) es un tarball (un tipo de archivo comprimido) que contiene un entorno mínimo.

- Un archivo *stage1* contiene poco más que un compilador, Portage (el sistema de gestión de software de Gentoo) y un par de paquetes más de los que dependen el compilador y Portage.
- Un archivo *stage2* contiene un sistema llamado *bootstrapped*, un entorno mínimo desde el cual puede construir otras aplicaciones necesarias que harán de Gentoo un sistema completo.
- Un archivo *stage3* contiene un sistema pre-compilado mínimo el cual es completamente funcional. Solamente le falta unas cuantas aplicaciones que el usuario de Gentoo necesita elegir las que quiere instalar.

En este documento optaremos por una instalación stage3. Si quiere llevar a cabo una instalación de Gentoo desde el stage1 o stage2, por favor utilice las instrucciones del [Manual de Gentoo](http://www.gentoo.org/doc/en/handbook/handbook-x86.xml) [http://www.gentoo.org/doc/en/handbook/handbook-x86.xml]. Necesitará una conexión a Internet.

. . . : El CD de instalación Universal de Gentoo

Un CD de instalación de Gentoo es un CD arrancable que contiene un entorno Gentoo auto-contenido. Permite arrancar Linux desde el CD. Durante el proceso de arranque se detecta el hardware y se cargan los controladores apropiados. Estos CDs son mantenidos por los desarrolladores de Gentoo.

Actualmente tenemos dos CDs de instalación disponibles:

- El CD de instalación Universal contiene todo aquello que necesita para instalar Gentoo. Proporciona archivos stage3 para las arquitecturas comunes, código fuente para aplicaciones extra sobre las que necesita elegir, y por supuesto, las instrucciones de instalación para su arquitectura.
- El CD de instalación Minimal solamente contiene un entorno mínimo que le permite arrancar y configurar la red para que pueda conectarse a Internet. No contiene ningún archivo adicional y no puede utilizarse para éste proceso de instalación (recordemos que estamos en las instrucciones para llevar a cabo la instalación sin conexión a la red).

Gentoo también proporciona un CD de paquetes (Package CD). Este CD no es de instalación pero contiene una fuente importante de recursos que puede ser utilizados durante la instalación de su sistema Gentoo. Contiene paquetes precompilados (conocidos como GRP) que permiten una rápida y fácil instalación de aplicaciones adicionales (como pueden ser (OpenOffice.org, KDE, GNOME, ...) inmediatamente después de la instalación de Gentoo y justo antes de actualizar su árbol Portage.

El empleo del CD de paquetes está recogido en este documento más adelante

3: Descargar, Grabar y Arrancar un CD de instalación de Gentoo

Ha elegido usar un CD de instalación Gentoo. Empezaremos por descargar y grabar el CD que eligió. Ya hablamos anteriormente de los distintos CD de instalación disponibles, pero ¿dónde podemos encontrarlos?

Puede descargar cualquiera de los CD de instalación (y si quiere, también un CD de Paquetes) desde uno de nuestros [espejos](http://www.gentoo.org/main/en/mirrors.xml) [http://www.gentoo.org/main/en/mirrors.xml]. Los CDs de instalación se localizan en el directorio releases/x86/2005.1/livecd; los CDs de Paquetes se encuentran en el directorio releases/x86/2005.1/packagecd.

Los CDs de paquetes para las arquitecturas i686, athlon-xp, pentium3 y pentium4 están disponible vía [BitTorrent](http://torrents.gentoo.org) [http://torrents.gentoo.org].

Dentro del directorio encontrará los llamados archivos ISO. Estos son imágenes de CD completas que puede grabar en un CD-R.

En caso que quiera saber si el archivo que descargó está corrupto, puede verificar la suma de control MD5 de su archivo y compararla con la que nosotros proporcionamos (una como install-x86-minimal-2004.3.iso.md5). Puede revisar las sumas de control con [md5sum](http://www.etree.org/md5com.html) de Linux o con [md5sum](http://www.etree.org/md5com.html) [http://www.etree.org/md5com.html] en Windows.

Otra manera de verificar la validez de la descarga es usar GnuPG para revisar la firma criptográfica que proporcionamos (el nombre del archivo termina con .asc). Descargue el archivo de firma y obtenga la clave pública:

Code Listing 1: Obtener la clave pública

```
$ gpg --keyserver pgp.mit.edu --recv-keys 17072058
```

Ahora verificamos la firma:

Code Listing 2: Verificar la firma criptográfica

```
$ gpg --verify <archivo de firma> <archivo iso>
```

Para grabar el o los ISOs descargados, debe seleccionar grabar/quemar sin formato. Cómo hacerlo depende del programa. Aquí hablaremos de [cdrecord](#) y [K3B](#); puede encontrar mayor información en las [Preguntas de Uso Frecuentes](#) [<http://www.gentoo.org/doc/es/faq.xml#isoburning>] sobre Gentoo.

- Con [cdrecord](#), simplemente escriba `cdrecord dev=/dev/hdc` (reemplace `/dev/hdc` con la ruta de su unidad CD-RW).
- Con [K3B](#), seleccione [Herramientas](#) > [CD](#) > [Grabar Imagen ISO](#). Luego busque el archivo ISO dentro del área 'Imagen a Grabar'. Por último presione el botón comenzar.

... : Arrancar el/los CDs de instalación

Important: Lea completamente esta subsección antes de continuar, probablemente no tendrá la oportunidad de leerla mientras esté haciendo cosas luego.

Una vez que ha grabado sus CDs de instalación, es hora de arrancarlos. Retire todos los CD de sus unidades, reinicie el sistema y entre en el BIOS. Esto se logra por lo general oprimiendo la tecla Supr, F1 o Esc, dependiendo del BIOS. Dentro del BIOS, cambie el orden de arranque para que intente arrancar primero desde el CD-ROM. Regularmente se encuentra dentro de "CMOS Setup". Si no lo hace, su sistema arrancará desde el disco duro e ignorará el CD-ROM.

Coloque el CD de instalación en la unidad CD-ROM y reinicio. Deberá ver un símbolo de espera de órdenes. En esta pantalla puede oprimir la tecla Enter para empezar el proceso de arranque con las opciones predefinidas, o arrancar el CD de instalación con opciones personales especificando un kernel seguido por las opciones de arranque y luego oprimir la tecla Enter.

¿Especificamos un kernel? Sí, ya que proporcionamos varios en nuestros CD de instalación. El predeterminado es [gentoo](#). Otros núcleos son para hardware específico que necesitan las variantes [-nofb](#) que tienen desactivado el framebuffer.

Más adelante encontrará una breve descripción de los kernels disponibles:

<i>Kernel</i>	<i>Descripción</i>
gentoo	El Kernel 2.6 predeterminado con soporte para múltiples CPUs
gentoo-nofb	Igual que gentoo pero sin soporte para framebuffer
memtest86	Para verificar su memoria RAM para errores

También puede pasarle opciones al kernel. Estas representan configuraciones

opcionales que puede activar o desactivar. La siguiente lista es la misma que se muestra al presionar F2 en la pantalla de arranque.

Code Listing 3: Opciones que puede pasarle al kernel de su elección

```
- agpgart      carga agpart (úsela si tiene problemas con los gráficos)
- acpi=on     carga el soporte para el firmware ACPI
- ide=nodma   fuerza la desactivación para dispositivos IDE que no funcionen
correctamente con DMA
- doscsi      busca dispositivos scsi (bloquea algunas tarjetas ethernet)
- dopcmcia    inicia el servicio pcmcia para PCMCIA cdroms
- nofirewire  desactiva los modules firewire en initrd (para cdroms firewire,
etc)
- nokeymap    desactiva la selección de teclado para teclados distintos a los
"us"
- docache     guarda la parte de ejecución completa del cd en RAM, permite
montar
              y desmontar /mnt/cdrom para utilizar otro.
- nodetect    provoca que no se ejecuten hwsetup/kudzu y hotplug
- nousb      desactiva el módulo usb cargado por initrd, desactiva hotplug
- nodhcp      no inicia automáticamente dhcp cuando se detecta una nic
- nohotplug   desactiva la carga del servicio hotplug
- noapic      desactiva apic (pruebelo si tiene problemas con el hardware
nics,scsi,etc)
- noevms2     desactiva la carga de los módulos para EVMS2
- nolvm2     desactiva la carga de los módulos para LVM2
- hdx=stroke  le permite particionar el disco duro incluso si su BIOS no puede
manejar
              grandes unidades.
- noload=module1[,module2[,...]]
              desactiva la carga de modulos específicos en el kernel
```

Ahora arranque su CD, seleccione un kernel (si no está contento con el kernel predeterminado [gentoo](#)) y las opciones de arranque. Por ejemplo, aquí le mostramos como arrancar el kernel [gentoo](#), con el parámetro [dopcmcia](#):

Code Listing 4: Arrancando un CD de instalación

```
boot: gentoo dopcmcia
```

Después verá otra pantalla de arranque y una barra de progreso. Si está instalando Gentoo en un sistema con un teclado no americano, asegúrese de pulsar Alt-F1 para cambiar al modo "verbose" y seguir el cursor. Si no se hace ninguna selección en 10 segundos, la opción predefinida (teclado americano) será la utilizada y el proceso de arranque continuará. Cuando se complete el proceso de arranque, automáticamente se registrará en el Gentoo Linux "Live" como "root", el super usuario. Tendrá el indicador del root ("#") en la consola actual y también podrá cambiarse a otras consolas presionando Alt-F2, Alt-F3 y Alt-F4. Para regresar a la consola en la que empezó presione Alt-F1.

Continúe ahora Configurando Hardware Adicional.

... : Configurando Hardware Adicional

Cuando arranca el CD de instalación, trata de detectar todos los dispositivos de hardware y cargar los módulos apropiados en el kernel para soportar este hardware. En la gran mayoría de casos, hace muy buen trabajo. Sin embargo, en algunos casos, puede que no cargue automáticamente los módulos del kernel necesarios. Si falla la auto-detección PCI con algún hardware de su sistema, trate de cargar el módulo apropiado manualmente.

En el siguiente ejemplo trataremos de cargar el módulo [8139too](#) (que da soporte a ciertos tipos de interfaces de red):

Code Listing 5: Cargar módulos del kernel

```
# modprobe 8139too
```

Si necesita soporte PCMCIA, debe ejecutar el script de inicio de [pcmcia](#):

Code Listing 6: Ejecutando el script de inicio de PCMCIA

```
# /etc/init.d/pcmcia start
```

... : Opcional: Afinando el Desempeño del Disco Duro

Si usted es un usuario avanzado, tal vez quiera afinar el desempeño del disco duro IDE usando [hdparm](#). Con la opción `-tT` puede probar el desempeño de su disco (ejecútelo varias veces para tener una impresión más precisa):

Code Listing 7: Verificar el desempeño del disco duro

```
# hdparm -tT /dev/hda
```

Para afinarlo siga uno de los siguientes ejemplos (o haga sus propios experimentos) los cuales usan `/dev/hda` como disco (sustitúyalo por el disco en cuestión):

Code Listing 8: Afinar el desempeño del disco duro

```
Activar el DMA:  
# hdparm -d 1 /dev/hda  
Activar las opciones de optimización seguras:  
# hdparm -d 1 -A 1 -m 16 -u 1 -a 64 /dev/hda
```

... : Opcional: Cuentas de Usuarios

Si planea darle a otras personas acceso a su entorno de instalación o si quiere platicar con alguien usando [irssi](#) sin privilegios de root (por razones de seguridad), hará falta crear las cuentas de usuario necesarias y cambiar la contraseña de root.

Para cambiar la contraseña de root, use [passwd](#):

Code Listing 9: Cambiar la contraseña de root

```
# passwd  
New password: (Escriba su nueva contraseña)  
Re-enter password: (Escríbala de nuevo)
```

Para crea una cuenta de usuario, primero escribimos sus credenciales, y luego su contraseña. Se usan [useradd](#) y [passwd](#) para estas tareas. En el siguiente ejemplo, crearemos un usuario llamado "john".

Code Listing 10: Crear una cuenta de usuario

```
# useradd john  
# passwd john  
New password: (Escriba la contraseña de john)  
Re-enter password: (Escriba otra vez la contraseña de john)
```

Puede cambiar su identificador de usuario desde root para usar la del nuevo usuario usando [su](#):

Code Listing 11: Cambiar de usuario

```
# su - john
```

... : Opcional: Ver la Documentación mientras Instala

Si quiere ver el Manual Gentoo (ya sea desde el CD o en línea) durante la instalación, asegúrese de haber creado una cuenta de usuario (consulte [Opcional: Cuentas de Usuarios](#)). Luego presione [Alt-F2](#) para ir a una nueva consola e ingrese al sistema.

Si quiere ver la documentación en el CD puede ejecutar inmediatamente [links2](#) para leerla:

Code Listing 12: Ver la Documentación en CD

```
# links2 /mnt/cdrom/docs/handbook/html/index.html
```

Sin embargo, es preferible que use el Manual Gentoo en línea ya que es más reciente que el Manual en el CD. También puede verlo usando [links2](#), pero solo después de haber completado el capítulo *Configurar su Red* (de otro modo no podrá usar Internet para ver el documento):

Code Listing 13: Ver la Documentación en Línea

```
# links2 http://www.gentoo.org/doc/es/handbook/handbook-x86.xml
```

Puede regresar a la consola original presionando [Alt-F1](#).

. . . : Opcional: Iniciar el Demonio SSH

Si quiere permitirle a otros usuarios el acceso a su máquina durante la instalación de Gentoo (quizá porque esos usuarios le vayan a ayudar, o incluso lo hagan por usted), necesitará crear una cuenta de usuario para ellos e incluso darles la contraseña de root (*sólo si confía totalmente* en ese usuario).

Para arrancar el demonio SSH, ejecute el siguiente comando:

Code Listing 14: Iniciar el demonio SSH

```
# /etc/init.d/sshd start
```

Para poder usar `sshd`, primero necesita configurar su red. Continúe en el capítulo [Configurar su Red](#).

Configuración de su red

“Si desea instalar Gentoo a través de Internet, debe configurar la red.”

1: ¿Necesita conexión a la red?

Normalmente, no necesitará una conexión activa para instalar Gentoo utilizando el CD de instalación Universal. Sin embargo, hay ciertas circunstancias en las que quizá quiera utilizar una conexión a Internet activa:

- Los archivos stage3 que se almacenan en el CD de instalación Universal no coinciden con su arquitectura y necesita descargar el archivo stage3 correcto
- Necesita instalar una aplicación de red específica que le permita conectarse a Internet la cual no está disponible en el CD de instalación Universal pero está soportada por el CD de instalación (e.j. puede conectarse a Internet utilizando el CD de instalación pero las fuentes necesarias no están disponibles en el CD de instalación)
- Quiere asistencia remota durante la instalación (utilizando SSH o a través de conversaciones por IRC)

... : ¿Necesito conexión de red?

Para saber si el archivo stage3 de su arquitectura está disponible, eche un vistazo a `/mnt/cdrom/stages` y compruebe si uno de los stages corresponde a su arquitectura. Si no, todavía puede optar por un stage3 de una arquitectura compatible con la suya.

Si, por el contrario, quiere utilizar un fichero stage3 optimizado para su arquitectura y no está disponible, entonces necesitará una conexión de red funcionando para descargar el stage3 apropiado.

Entonces, si no necesita una conexión a la red, sáltese el resto de este capítulo y continúe con [Preparando los Discos](#). De lo contrario, continúe con las secciones de configuración de red más abajo.

2: Auto Detección de Red

Si su sistema está conectado a una red Ethernet con un servidor DHCP, es muy probable que la configuración de red se haya detectado automáticamente. En ese caso, debería ser capaz de trabajar con los comandos que hacen uso de la red y que están en el CD de instalación como [ssh](#), [scp](#), [ping](#), [irssi](#), [wget](#) y [links](#), entre otros.

Si la red ya ha sido configurada el comando `/sbin/ifconfig` debería listar algunas interfaces de red además de lo, como `eth0`:

Code Listing 1: `/sbin/ifconfig` para una tarjeta de red funcionando

```
# /sbin/ifconfig
(...)
eth0      Link encap:Ethernet  HWaddr 00:50:BA:8F:61:7A
          inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::50:ba8f:617a/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1498792 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1284980 errors:0 dropped:0 overruns:0 carrier:0
          collisions:1984 txqueuelen:100
          RX bytes:485691215 (463.1 Mb)  TX bytes:123951388 (118.2 Mb)
          Interrupt:11 Base address:0xe800
```

... : Opcional: Configurar Proxys

Si accede a Internet a través de un proxy, podría necesitar configurar la información del proxy durante la instalación. Es muy sencillo definir un proxy: tan solo necesita definir la variable que contiene la información del mismo.

En la mayoría de los casos, usted puede definir las variables usando simplemente el host del servidor. Por ejemplo, asumimos que el proxy se llama [proxy.gentoo.org](#) y el puerto es el [8080](#).

Code Listing 2: Definiendo los servidores proxy

```
(Si el proxy filtra el tráfico HTTP)
# export http_proxy="http://proxy.gentoo.org:8080"
(Si el proxy filtra el tráfico FTP)
# export ftp_proxy="ftp://proxy.gentoo.org:8080"
(Si el proxy filtra el tráfico RSYNC)
# export rsync_proxy="rsync://proxy.gentoo.org:8080"
```

Si su proxy requiere un nombre de usuario y una contraseña, debería usar la siguiente sintaxis para la variable:

Code Listing 3: Añadiendo usuario/contraseña a la variable del proxy

```
http://usuario:contraseña@servidor
```

. . . : Probando la red

Usted podría intentar hacer ping hacia el servidor DNS de su ISP, (que se encuentra en `/etc/resolv.conf`) y un sitio web a su elección, con la intención de asegurarse que sus paquetes llegan a la red, la resolución de nombres DNS esta funcionando correctamente, etc.

Code Listing 4: Más testeo de red

```
# ping -c 3 www.yahoo.com
```

¿Está su conexión funcionando? En ese caso, puede saltarse el resto de esta sección y continuar con [Preparando los discos](#). De lo contrario, siga leyendo.

3: Configuración Automática de Red

Si la red no funciona inmediatamente, algunos medios de instalación le permitirán usar [net-setup](#) (para redes estándar o sin cables), [adsl-setup](#) (para usuarios de ADSL) o [pptp](#) (para usuarios de PPTP solamente disponible para x86).

Si su medio de instalación no contiene ninguna de estas herramientas, continúe con [Configuración Manual de la Red](#).

- Los usuarios de redes estándar deberían continuar con [Por defecto: Usando net-setup](#)

- Los usuarios de ADSL deberían continuar con [Alternativa: Usando RP-PPPoE](#)
- Los usuarios de PPTP deberían continuar con [Alternativa: Usando PPTP](#)

... : Por defecto: Usando net-setup

El sistema más simple de configurar una red, si no se consiguió hacerlo automáticamente, es ejecutar el script [net-setup](#):

Code Listing 5: Ejecutando el script net-setup

```
# net-setup eth0
```

[net-setup](#) le hará algunas preguntas sobre su entorno de red. Cuando lo haya completado, debería disponer de una conexión de red funcionando. Pruebe su conexión de red como se especificó anteriormente. Si los resultados son positivos, ¡felicidades! Ahora está listo para instalar Gentoo. Sáltese el resto de esta sección y continúe con [Preparando los discos](#).

Si su conexión de red sigue sin funcionar, continúe con [Configuración Manual de la Red](#).

... : Alternativa: Usando RP-PPPoE

Asumiendo que requiere PPPoE para conectar a Internet, el CD de instalación (en cualquier versión) ha sido pensado para facilitarle el trabajo incluyendo [rp-pppoe](#). Use el script [adsl-setup](#) proporcionado para configurar su conexión. Se le pedirá el dispositivo de red que está conectado a su módem adsl, su nombre de usuario y su contraseña, las IPs de sus servidores DNS y si requiere un firewall básico o no.

Code Listing 6: Usando rp-pppoe

```
# adsl-setup  
# adsl-start
```

Si algo va mal, asegúrese que ha tecleado correctamente su nombre de usuario y su contraseña mirando `/etc/ppp/pap-secrets` o `/etc/ppp/chap-secrets` y asegúrese que está haciendo uso del dispositivo de red correcto. Si su dispositivo de red no existe, deberá cargar los módulos de red apropiados. En ese caso, debería continuar con [Configuración manual de la Red](#) donde explicaremos como cargar los módulos de red apropiados.

Si todo funcionó, continúe con [Preparando los Discos](#).

. . . : Alternativa: Usando PPTP

Note: PPTP solamente está disponible para x86.

Si requiere soporte PPTP, puede usar [pptpclient](#) que se incluye en nuestros CDs de instalación. Pero primero debe asegurarse que su configuración es correcta. Edite `/etc/ppp/pap-secrets` o `/etc/ppp/chap-secrets` ya que contiene la combinación correcta de usuario/contraseña:

Code Listing 7: Editando `/etc/ppp/chap-secrets`

```
# nano -w /etc/ppp/chap-secrets
```

Ajuste ahora `/etc/ppp/options.pptp` si es necesario:

Code Listing 8: Editando `/etc/ppp/options.pptp`

```
# nano -w /etc/ppp/options.pptp
```

Cuando todo esté listo, tan sólo ejecute [pptp](#) (junto con las opciones que podría haber configurado en `options.pptp`) para conectar al servidor:

Code Listing 9: Conectando a un servidor por línea telefónica

```
# pptp <server ip>
```

Ahora continúe con [Preparando los discos](#).

4: Configuración Manual de la Red

Cuando el CD de instalación arranca, intenta detectar todos sus dispositivos hardware y carga los módulos del kernel (controladores) apropiados para darles soporte. En la gran mayoría de los casos, hace un muy buen trabajo. No obstante, en algunos casos, puede no cargar automáticamente los módulos del kernel que necesita.

Si [net-setup](#) o [adsl-setup](#) fallaron, entonces puede asumir sin ningún riesgo que su tarjeta de red no se encontró en el acto. Esto significa que tendrá que cargar los módulos del kernel apropiados manualmente.

Para descubrir que módulos del kernel le proporcionamos para la red, use [ls](#):

Code Listing 10: Buscando los módulos apropiados

```
# ls /lib/modules/`uname -r`/kernel/drivers/net
```

Si encuentra un driver para su tarjeta de red, use [modprobe](#) para cargar el módulo del kernel:

Code Listing 11: Usando modprobe para cargar el módulo kernel

```
(Como ejemplo, cargaremos el módulo pcnet32)  
# modprobe pcnet32
```

Para confirmar si su tarjeta de red se detecta ahora, use [ifconfig](#). Una red detectada debería desembocar en algo como esto:

Code Listing 12: Comprobando la disponibilidad de su tarjeta de red, éxito

```
# ifconfig eth0  
eth0      Link encap:Ethernet  HWaddr FE:FD:00:00:00:00  
          BROADCAST NOARP MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Si de todas formas recibe el siguiente error, la tarjeta de red no está detectada:

Code Listing 13: Comprobando si la disponibilidad de su tarjeta de red fracasó

```
# ifconfig eth0  
eth0: error fetching interface information: Device not found
```

Si tiene múltiples tarjetas de red en su sistema éstas recibirán el nombre *eth0*, *eth1*, etc. Asegúrese que la tarjeta de red que quiere utilizar funciona correctamente y recuerde emplear el nombre correcto a lo largo de este documento. Asumiremos que la tarjeta de red *eth0* va a ser la utilizada.

Asumiendo que usted dispone ahora de una tarjeta de red red detectada, puede reintentar [net-setup](#) o [adsl-setup](#) otra vez (que deberían funcionar ahora), pero para los duros de entre ustedes, les explicaremos como configurar su red manualmente.

Seleccione una de las siguientes secciones basándose en su configuración de red.

- [Usando DHCP](#) para adjudicación de IP automática
- [Preparando el Acceso Inalámbrico \(Wireless\)](#) si tiene una tarjeta inalámbrica.
- [Entendiendo la terminología de red](#) explica lo que necesita conocer sobre

- redes.
- Utilizando ifconfig y route explica como configurar su red manualmente.

... : Usando DHCP

DHCP (Protocolo de Configuración Dinámica de Host) hace posible recibir automáticamente su información de red (Dirección IP, máscara de red, dirección de broadcast, pasarela, servidores de nombres etc.). Esto sólo funciona si usted dispone de un servidor DHCP en su red (o si su proveedor le ofrece servicio DHCP). Para conseguir que su interfaz de red reciba esta información automáticamente use `dhcpcd`:

Code Listing 14: Usando dhcpcd

```
# dhcpcd eth0
Algunos administradores de red requieren que utilice el
nombre del equipo y el dominio que proporciona el servidor DHCP.
Si es el caso, utilice
# dhcpcd -HD eth0
```

Si esto funciona (pruebe hacer `ping` a algún servidor en Internet, como [Google](http://www.google.com) [<http://www.google.com>]), entonces lo tiene todo configurado y listo para continuar. Sáltese el resto de esta sección y continúe con Preparando los Discos.

... : Preparando el Acceso Inalámbrico (Wireless)

Note: Soporte para `iwconfig` solamente está disponible en los CDs de instalación de las arquitecturas x86, amd64 y ppc. Si el suyo no lo tiene, todavía puede conseguir que sus extensiones funcionen siguiendo las instrucciones del [proyecto linux-wlan-ng](http://ftp.linux-wlan.org/pub/linux-wlan-ng/README) [[ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/README](http://ftp.linux-wlan.org/pub/linux-wlan-ng/README)].

Si está empleando una tarjeta inalámbrica (802.11), quizá necesite configurar sus opciones antes de ir más allá. Para revisar la configuración inalámbrica actual de su tarjeta, puede utilizar `iwconfig`. Ejecutando `iwconfig` debería mostrar algo como esto:

Code Listing 15: Mostrando la configuración inalámbrica actual

```
# iwconfig eth0
eth0      IEEE 802.11-DS  ESSID:"GentooNode"
          Mode:Managed  Frequency:2.442GHz  Access Point: 00:09:5B:11:CC:F2
          Bit Rate:11Mb/s  Tx-Power=20 dBm   Sensitivity=0/65535
          Retry limit:16   RTS thr:off    Fragment thr:off
          Power Management:off
          Link Quality:25/10  Signal level:-51 dBm  Noise level:-102 dBm
          Rx invalid nwid:5901 Rx invalid crypt:0 Rx invalid frag:0 Tx
          excessive retries:237 Invalid misc:350282 Missed beacon:84
```

Note: El nombre de dispositivo de algunas tarjetas de red inalámbricas puede ser `wlan0` o `ra0` en lugar de `eth0`. Ejecute `iwconfig` sin más parámetros para determinar cual es el nombre de dispositivo correcto.

La mayoría de usuarios, solamente tendrá que modificar dos configuraciones, el ESSID (Nombre de red inalámbrica) o la clave WEP. Si el ESSID y la dirección del Punto de Acceso mostradas son correctas y el Punto de Acceso y usted mismo no están utilizando WEP, su red inalámbrica está funcionando. Si necesita cambiar su ESSID, o añadir una clave WEP, puede utilizar los siguientes comandos:

Code Listing 16: Cambiando ESSID y/o añadiendo una clave WEP

```
(Esto asigna el nombre de "GentooNode" a nuestra red)
# iwconfig eth0 essid GentooNode

(Esto asigna una clave WEP hexadecimal)
# iwconfig eth0 key 1234123412341234abcd

(Esto asigna una clave ASCII , añadiendo al principio "s:")
# iwconfig eth0 key s:some-password
```

Puede volver a comprobar la configuración inalámbrica utilizando `iwconfig`. Una vez que tenga la conexión funcionando, puede continuar configurando las opciones de red de nivel IP como se describe en la siguiente sección (Entendiendo la terminología de red) o utilizar la herramienta `net-setup` como hemos descrito anteriormente.

. . . : Entendiendo la terminología de red

Note: Si conoce su dirección IP, dirección de broadcast, máscara de red y los servidores de nombres, entonces puede saltarse esta subsección y continuar con Usando ifconfig y route.

Si todo lo anterior falla, tendrá que configurar su red manualmente. No es demasiado complicado. Sin embargo, necesita familiarizarse con alguna

terminología de red para ser capaz de configurar su red satisfactoriamente. Después de leer esto, conocerá que es una *pasarela*, para que sirve una *máscara de red*, como se forma una dirección de *broadcast* y porqué necesita *servidores de nombres*.

En una red los hosts están identificados por su *dirección IP* (dirección del Protocolo de Internet). Tal dirección es una combinación de cuatro números entre 0 y 255. Bien, como mínimo así es como las percibimos. En realidad, tal dirección IP consiste de 32 bits (unos y ceros). Vamos a ver un ejemplo:

Code Listing 17: Ejemplo de una dirección IP

```
IP Address (numbers):  192.168.0.2
IP Address (bits):    11000000 10101000 00000000 00000010
                      -----
                      192      168      0      2
```

Cada dirección IP es única para ese host, tan lejos como redes accesibles estén a su disposición (por ejemplo, todos los hosts que usted es capaz de conectar, deben tener direcciones IP únicas). Para ser capaz de hacer la distinción entre hosts dentro de una red, y hosts fuera de una red, la dirección IP está dividida en dos partes: la parte de *red* y la parte de *host*.

La separación esta anotada en la *máscara de red*, una conjunto de unos seguidos de un conjunto de ceros. La parte de la IP que quedan enmascarados sobre los unos es la parte de red, la otra es la parte de host. Como es normal, la máscara de red, puede ser anotada como una dirección IP.

Code Listing 18: Ejemplo de una separación red/host

```
IP-address:   192      168      0      2
              11000000 10101000 00000000 00000010
Netmask:     11111111 11111111 11111111 00000000
              255      255      255      0
+-----+-----+-----+-----+
              Network      Host
```

En otras palabras, 192.168.0.14 es parte de nuestra red de ejemplo, pero 192.168.1.2 no lo es.

La dirección de *broadcast* es una dirección IP con la misma parte de red que su red, pero con solo unos como parte de host. Cada host en su red escucha esta dirección IP. Esto verdaderamente sirve para la transmisión de paquetes.

Code Listing 19: Dirección de Broadcast

```

IP-address:   192      168      0        2
              11000000 10101000 00000000 00000010
Broadcast:   11000000 10101000 00000000 11111111
              192      168      0        255
+-----+-----+-----+-----+
              Network      Host

```

Para ser capaz de navegar por Internet, debería conocer qué host comparte la conexión a Internet. Este host se llama la *pasarela*. Puesto que es un host estándar, tiene direcciones IP estándar (por ejemplo 192.168.0.1).

Anteriormente afirmamos que cada host tiene su propia dirección IP. Para ser capaz de alcanzar este host por un nombre (en vez de la dirección IP) necesitara un servicio que traduzca un nombre (como *dev.gentoo.org*) a una dirección IP (como *64.5.62.82*). Tal servicio se le conoce como servicio de nombres. Para usar tal servicio, debe definir los *servidores de nombres* necesarios en `/etc/resolv.conf`.

En algunos casos, su pasarela también sirve como servidor de nombres. De otro modo, tendrá que introducir los servidores de nombres facilitados por su ISP.

Para resumir, necesitará la siguiente información antes de continuar:

<i>Elemento de Red</i>	<i>Ejemplo</i>
Su dirección IP	192.168.0.2
Máscara de Red	255.255.255.0
Broadcast	192.168.0.255
Pasarela	192.168.0.1
Servidor(es) de nombres	195.130.130.5, 195.130.130.133

... : Usando ifconfig y route

Configurar su red consiste en tres pasos. Primero, nos asignamos una dirección IP usando `ifconfig`. Entonces configuraremos el ruteo hacia la pasarela usando `route`. Por último, acabaremos situando las IPs de los servidores de nombres en `/etc/resolv.conf`.

Para asignar una dirección IP, necesitara su dirección IP, la dirección de broadcast y la mascara de red. Entonces ejecute el siguiente comando, substituyendo `#{IP_ADDR}` con su dirección IP, `#{BROADCAST}` con su dirección de broadcast y `#{NETMASK}` con su máscara de red:

Code Listing 20: Usando ifconfig

```
# ifconfig eth0 ${IP_ADDR} broadcast ${BROADCAST} netmask ${NETMASK} up
```

Ahora configure la ruta usando [route](#). Sustituya `${GATEWAY}` con la dirección IP de su pasarela:

Code Listing 21: Usando route

```
# route add default gw ${GATEWAY}
```

Ahora abra `/etc/resolv.conf` con su editor favorito (en nuestro ejemplo, usaremos [nano](#)):

Code Listing 22: Creando `/etc/resolv.conf`

```
# nano -w /etc/resolv.conf
```

Ahora complete con su(s) servidor(es) de nombres usando la siguiente plantilla. Asegúrese que sustituye `${NAMESERVER1}` y `${NAMESERVER2}` con las direcciones apropiadas de servidor de nombres:

Code Listing 23: Plantilla `/etc/resolv.conf`

```
nameserver ${NAMESERVER1}
nameserver ${NAMESERVER2}
```

Eso es. Ahora compruebe su red haciendo [ping](#) a algún servidor de Internet (como [Google](http://www.google.com) [<http://www.google.com>]). Si funciona, felicitaciones entonces. Ahora está listo para instalar Gentoo. Continúe con [Preparando los Discos](#).

Preparando los discos

“Para poder instalar Gentoo, debe crear las particiones necesarias. Este capítulo describe cómo particionar un disco para uso futuro.”

1: Introducción a Dispositivos de Bloque

Examinaremos de forma detallada los aspectos de Gentoo Linux así como Linux en general que tengan que ver con discos, sistemas de ficheros de Linux, particiones y dispositivos de bloque. Una vez esté familiarizado con las entrañas de los discos y sistemas de ficheros, le guiaremos a través del proceso de creación de particiones y sistemas de ficheros de tu instalación Gentoo Linux.

Para empezar, explicaremos el término *dispositivos de bloque*. Quizás el dispositivo de bloque más conocido es el que representa la primera unidad IDE llamada `/dev/hda` en un sistema Linux. Si tu máquina utiliza discos SCSI entonces el primer disco duro se denominará `/dev/sda`.

Los dispositivos de bloque mencionados anteriormente representan una interfaz abstracta de disco. Las aplicaciones pueden hacer uso de estas interfaces para interactuar con el disco duro de la máquina sin importar el tipo de unidad que tienes: IDE, SCSI, o cualquier otra. La aplicación puede simplemente dirigirse al almacenamiento en el disco como a una serie de bloques de acceso aleatorio de 512-bytes situados de forma contigua.

. . . : Particiones

Aunque teóricamente es posible utilizar el disco duro completo para albergar la instalación Linux, esto casi nunca se hace. En su lugar, los dispositivos de bloque enteros se dividen en partes más manejables y pequeñas. En los sistemas x86 éstas se llaman *particiones*.

Particiones se dividen en tres tipos: *primarias*, *extendidas* y *lógicas*.

La partición *primaria* es aquella que almacena su información en el MBR (registro principal de arranque). Ya que el MBR puede almacenar hasta 512 bytes, sólo pueden definirse cuatro particiones primarias (por ejemplo, desde `/dev/hda1`

hasta /dev/hda4).

Una partición *extendida* es una partición primaria especial (entendemos que la partición extendida debe ser una de las cuatro posibles particiones primarias) la cual contiene más particiones. Al principio no existía este tipo de partición, pero como cuatro primarias eran muy pocas, se diseñó para extender el esquema de particionamiento sin perder la compatibilidad inversa.

Una partición *lógica* es aquella que está dentro de la partición extendida. En otras palabras, estas particiones no se definen dentro del MBR, sino que se declaran dentro de la partición extendida.

. . . : Almacenamiento Avanzado

Los CDs de instalación proporcionan soporte para EVMS (en inglés, Enterprise Volume Management System) manejo empresarial de volúmenes o LVM2 (en inglés, Logical Volume Management) manejo lógico de volúmenes, incrementando la flexibilidad ofrecida por el esquema de particionamiento. Durante la instalación las instrucciones principalmente se enfocan sobre particiones "habituales", pero es bueno saber que EVMS y LVM2 también están soportadas.

2: Diseñando un Esquema de Particionamiento

Si no está interesado en diseñar un esquema de particionamiento particular para tu sistema, puede hacer uso del esquema que utilizaremos en este manual:

<i>Partición</i>	<i>Sistema de Ficheros</i>	<i>Tamaño</i>	<i>Descripción</i>
/dev/hda1	ext2	32M	Partición de arranque
/dev/hda2	(swap)	512M	Partición de intercambio
/dev/hda3	ext3	El resto del disco	Partición de raíz

Si está interesado en conocer el tamaño que debería tener una partición, o incluso cuantas particiones necesita, continúe leyendo. En caso contrario, siga con el particionamiento del disco leyendo [Utilizando fdisk para particionar su disco](#).

... : ¿Cuántas? y ¿de qué tamaño?

El número de particiones que necesita depende mucho de su entorno particular. Por ejemplo, si la máquina tiene muchos usuarios, lo más probable es que desee tener `/home` en una partición separada para facilitar las tareas de copia de respaldo y aumentar la seguridad. Si está instalando Gentoo para funcionar como servidor de correo, debe tener `/var` sobre una partición separada ya que es allí dónde se almacena todo el correo. Asimismo, una buena elección de sistema de ficheros optimizará el rendimiento del equipo. Las estaciones de juegos deben disponer de una partición `/opt` ya que la mayoría de juegos se instalan en ese directorio. Las razones para todas estas recomendaciones son similares a aquellas que hemos mencionado para el caso de `/home`: seguridad y salvaguarda de datos. Muy recomendable mantener `/usr` con un tamaño considerable: no solamente contiene la mayor parte de las aplicaciones sino que además el propio árbol de Portage ocupa sobre unos 500 Mbytes sin contar las fuentes que guarda dentro de él.

Como puede ver, todo depende de lo que quiera conseguir. Tener particiones o volúmenes separados tiene las siguientes ventajas:

- Puede elegir el mejor sistema de ficheros para cada partición o volumen
- El equipo en su totalidad no puede quedar sin espacio si una herramienta o aplicación está escribiendo datos de forma continua al volumen o partición
- Si es el caso, el tiempo dedicado a las comprobaciones de integridad de sistemas de fichero se reduce ya que las comprobaciones pueden ser llevadas a cabo en paralelo (sin embargo esta ventaja es mayor con múltiples discos que con múltiples particiones)
- La seguridad puede ser aumentada montando algunas de las particiones en modo sólo lectura, `nosuid` (los `setuid` bits se ignoran), `noexec` (los bits de ejecución se ignoran), etc.

Pero, tener múltiples particiones tiene una gran desventaja: si la configuración no es la adecuada, puede acabar teniendo mucho espacio libre en una de las particiones y quedarse sin espacio en otras. Además, existe un límite de 15 particiones para SCSI y SATA.

Como ejemplo de un esquema de particionamiento utilizaremos un disco duro de 20 Gb de un portátil para fines de demostración (incluye servidor web, servidor de correo, gnome, etc.):

Code Listing 1: Ejemplo de particionamiento

```
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/hda5       ext3      509M  132M  351M  28% /
/dev/hda2       ext3      5.0G  3.0G  1.8G  63% /home
/dev/hda7       ext3      7.9G  6.2G  1.3G  83% /usr
/dev/hda8       ext3     1011M  483M  477M  51% /opt
/dev/hda9       ext3      2.0G  607M  1.3G  32% /var
/dev/hda1       ext2       51M   17M   31M  36% /boot
/dev/hda6       swap      516M   12M  504M   2% <not mounted>
(Espacio sin particionar para uso futuro: 2 Gb)
```

/usr parece estar bastante llena (83%), pero una vez que todo el software esté instalado no tiende a llenarse más. La gente puede pensar que el espacio que asignamos a /var es excesivo, sin embargo hay que saber que Gentoo compila todas las aplicaciones en /var/tmp/portage, por lo tanto se debe tener al menos 1 Gb de espacio libre en /var si no se planea compilar aplicaciones grandes, en caso contrario se recomienda tener 3 Gb libres, si compilar KDE u OpenOffice.org no es demasiado para usted.

3: Utilizando fdisk para particionar su disco

Las siguientes instrucciones explican como particionar el disco duro según el esquema descrito anteriormente:

<i>Partición</i>	<i>Descripción</i>
/dev/hda1	Partición de arranque (boot)
/dev/hda2	Partición de intercambio (swap)
/dev/hda3	Partición de raíz (root)

Cambie el esquema de particionamiento según su propio criterio.

... : Examinando el Esquema de Particionamiento Actual

fdisk es una herramienta potente y bastante popular que permite dividir el disco en particiones. Arranca **fdisk** sobre tu unidad de disco (en nuestro ejemplo usamos el dispositivo de disco /dev/hda):

Code Listing 2: Iniciando fdisk

```
# fdisk /dev/hda
```

Una vez que `fdisk` esté en ejecución, el programa ofrecerá el siguiente símbolo de comandos:

Code Listing 3: Símbolo de espera de órdenes de `fdisk`

```
Command (m for help):
```

Teclee `p` para mostrar el esquema de particionamiento actual:

Code Listing 4: Un ejemplo sobre configuración de particiones

```
Command (m for help): p
```

```
Disk /dev/hda: 240 heads, 63 sectors, 2184 cylinders  
Units = cylinders of 15120 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	14	105808+	83	Linux
/dev/hda2		15	49	264600	82	Linux swap
/dev/hda3		50	70	158760	83	Linux
/dev/hda4		71	2184	15981840	5	Extended
/dev/hda5		71	209	1050808+	83	Linux
/dev/hda6		210	348	1050808+	83	Linux
/dev/hda7		349	626	2101648+	83	Linux
/dev/hda8		627	904	2101648+	83	Linux
/dev/hda9		905	2184	9676768+	83	Linux

```
Command (m for help):
```

Este disco en particular está configurado para albergar siete sistemas de ficheros Linux, cada uno con su correspondiente partición con la etiqueta "Linux", así como una partición de intercambio (swap) que aparece con la etiqueta "Linux swap".

. . . : Eliminación de todas las Particiones

Primero eliminaremos todas las particiones existentes en el disco. Teclee `d` para eliminar una partición, seguido por `intro`. Por ejemplo, para borrar una partición existente en `/dev/hda1`:

Code Listing 5: Eliminando una partición

```
>  
Command (m for help): d  
Partition number (1-4): 1
```

La partición ha sido marcada para su borrado. Ya no aparecerá si teclea `p`, pero no será eliminada hasta que guarde los cambios realizados. Si comete una equivocación y desea abortar los cambios realizados, teclee `q` inmediatamente y pulse `intro`; las particiones no serán eliminadas.

Ahora, asumiendo que intenta eliminar todas las particiones existentes del disco duro, debe teclear **p** en forma repetida para ver el listado de particiones y pulsar **d** junto con el número de la partición para borrarlas. Finalmente, acabará teniendo una tabla de particiones vacía:

Code Listing 6: Una tabla de particiones vacía

```
Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot      Start          End      Blocks   Id  System

Command (m for help):
```

Ahora que la tabla de particiones en memoria del sistema está vacía, estamos preparados para crear nuevas particiones. Utilizaremos el esquema por defecto, tal como hemos acordado anteriormente. ¡Claro está, que no debe seguir estas instrucciones al pie de la letra si no desea tener una tabla de particiones exactamente igual que la nuestra!

. . . : Creación de una Partición de Arranque

En primer lugar debemos crear una pequeña partición de arranque. Teclee **n** para crear esta nueva partición, y luego **p** para seleccionar una partición primaria, siguiendo por **1** para elegirla como primera partición primaria. Cuando el sistema solicite introducir el primer cilindro, pulse **intro** y cuando pida definir el valor del último cilindro, teclee **+32M** para crear una partición de tamaño 32 Mb.:

Code Listing 7: Creando una partición de arranque

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-3876, default 1): (Hit Enter)
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-3876, default 3876): +32M
```

Si ahora teclea **p**, debe ver la siguiente partición en la tabla:

Code Listing 8: Una partición creada

```

Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot      Start         End      Blocks   Id  System
/dev/hda1          1           14     105808+   83  Linux

```

Necesitamos hacer que esta partición sea arrancable. Teclee **a** para marcar esta partición como arrancable. Si introduce **p** de nuevo, verá que el ***** ha aparecido en la columna "Boot".

. . . : Creación de una Partición de Intercambio

Vamos a crear ahora la partición de intercambio. Para hacerlo, teclee **n** para crear una nueva partición, y luego **p** para comunicar a fdisk que debe ser una partición primaria. Entonces teclee **2** para crear la segunda partición primaria, `/dev/hda2` en nuestro caso. Cuando el sistema solicite introducir el valor del primer cilindro, pulse **intro** y cuando solicite introducir el valor del último, teclee **+512M** para crear una partición de 512 Mb. Cuando lo haya hecho, teclee **t** para establecer el tipo de partición, **2** para seleccionar la partición que acaba de crear y entonces **82** para fijar el tipo "Linux Swap". Una vez completados estos pasos la introducción de **p** visualizará la tabla de particiones que debe ser similar a ésta.

Code Listing 9: Listado de particiones después de la creación de la partición de intercambio

```

Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot      Start         End      Blocks   Id  System
/dev/hda1 *        1           14     105808+   83  Linux
/dev/hda2          15           81     506520    82  Linux swap

```

... : Creación de la Partición de Raíz

En el último lugar, creamos la partición de raíz. Introduzca **n** para crear la nueva partición, **p** para marcarla como partición primaria. A continuación teclee **3** para crear la tercera partición primaria, `/dev/hda3`, según nuestro ejemplo. Al solicitar la introducción del valor del primer cilindro de la partición pulsamos **intro**, mientras que cuando el sistema solicite que introduzcamos el valor del último cilindro, también le damos a **intro** para crear una partición que ocupe todo el espacio restante en el disco. Tras completar todos estos pasos, introducimos **p** para ver la tabla de particiones que debe parecer mucho a la siguiente:

Code Listing 10: Listado de particiones después de crear la partición de raíz

```
Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot      Start         End      Blocks   Id  System
/dev/hda1 *         1          14        105808+   83  Linux
/dev/hda2           15          81        506520   82  Linux swap
/dev/hda3           82         3876     28690200  83  Linux
```

... : Almacenamiento de la Tabla de Particiones

Para guardar el esquema de particionamiento y salir del **fdisk** tecleamos **w**.

Code Listing 11: Guardado y salida del fdisk

```
Command (m for help): w
```

Ahora que las particiones están creadas, puede proseguir con la [Creación de Sistemas de Ficheros](#).

4: Creación de Sistemas de Ficheros

Ahora que ya tiene creadas las particiones, debe formatearlas para poder tener un sistema de ficheros. Si no le importa el tipo de sistema de ficheros que desee utilizar y está conforme con nuestra elección por defecto, continúe con la sección [Creación de Sistema de Ficheros en una Partición](#). En caso contrario, siga leyendo para ver qué sistemas de ficheros puede utilizar ...

... : ¿Sistemas de Ficheros??

El kernel de Linux soporta varios sistemas de ficheros. Explicaremos ext2, ext3, ReiserFS, XFS y JFS por ser los más utilizados en sistemas Linux.

ext2 es un sistema de ficheros Linux probado, pero no dispone de soporte para transacciones, lo que significa que las comprobaciones rutinarias al arrancar pueden tardar bastante tiempo. Ahora, hay muchas opciones alternativas, sistemas de ficheros de nueva generación con soporte para transacciones cuya integridad puede ser verificada con mayor rapidez, por lo que gozan de mayor popularidad. Los sistemas de ficheros transaccionales previenen retrasos durante el reinicio del equipo, incluso cuando el sistema de ficheros está en un estado inconsistente.

ext3 es la versión transaccional de ext2, que proporciona soporte para una rápida recuperación además de otros modos mejorados de funcionamiento como registro completo y ordenado de datos. Ext3 es un buen sistema de ficheros además de fiable. Posee una opción adicional para indización basada en árboles B que proporciona un alto rendimiento en casi todas las situaciones. Se puede activar esta opción añadiendo `-O dir_index` al comando `mke2fs`. En resumen, ext3 es un excelente sistema de ficheros.

ReiserFS es un sistema de ficheros B*-tree (basado en árboles balanceados) que tiene un gran rendimiento y que sobrepasa con creces a ext2 y ext3 cuando se trate de trabajar con ficheros pequeños (archivos menores de 4kb.), a veces diez o quince veces mejor. ReiserFS es extremadamente escalable y soporta transaccionalidad. Desde la versión 2.4.18+, ReiserFS es sólido y estable para su uso en casos genéricos así como en casos extremos cuando es necesario trabajar por ejemplo con sistemas de ficheros grandes, utilizar múltiples ficheros pequeños

o manejar archivos grandes y directorios con miles y miles de ficheros.

XFS es un sistema de ficheros transaccional el cual viene con un juego de características robustas y está optimizado para ser escalable. Recomendamos el uso de este sistema de ficheros para aquellas plataformas Linux que dispongan de dispositivos de almacenamiento SCSI de alto rendimiento y/o almacenamientos de canal de fibra (en inglés, fiber channel) con sistema de alimentación ininterrumpida. XFS realiza un almacenamiento temporal agresivo de datos en tránsito en RAM, pues aquellas aplicaciones con defectos de diseño (de las cuales hay muchas) que no toman precauciones necesarias durante la escritura de datos al disco pueden perderlos en caso de que el sistema se apague de forma inesperada.

JFS de IBM es un sistema de ficheros de alto rendimiento con soporte transaccional. Sólo recientemente ha entrado en fase de producción por lo tanto en este momento todavía no hay datos suficientes para opinar de forma favorable o negativa sobre su estabilidad.

. . . : Creación de Sistema de Ficheros en una Partición

Para crear un sistema de ficheros en una partición o volumen existen herramientas específicas para cada sistema de ficheros:

<i>Sistema de Ficheros</i>	<i>Comando de Creación</i>
ext2	mke2fs
ext3	mke2fs -j
reiserfs	mkreiserfs
xf	mkfs.xfs
jfs	mkfs.jfs

Por ejemplo, para formatear la partición de arranque (/dev/hda1 según el ejemplo) en formato ext2 y la partición de raíz (/dev/hda3 según el ejemplo) en formato ext3, utilizaría los siguientes comandos:

Code Listing 12: Creación de un sistema de ficheros en una partición

```
# mke2fs /dev/hda1
# mke2fs -j /dev/hda3
```

Y ahora, puede crear sistemas de fichero sobre sus particiones o volúmenes lógicos recién creados.

. . . : Activando la Partición de Intercambio

`mkswap` es el comando usado para inicializar particiones swap:

Code Listing 13: Inicialización de una partición de intercambio

```
# mkswap /dev/hda2
```

Para activar la partición, usa el comando `swapon`:

Code Listing 14: Activación de una partición de intercambio

```
# swapon /dev/hda2
```

Crea y activa tu partición de intercambio ahora.

5: Montaje

Ahora que las particiones están inicializadas y albergan sistemas de ficheros, es la hora de montarlas. Utiliza el comando `mount`. No olvides de crear puntos de montaje necesarios para cada partición que has creado. Como ejemplo montamos la partición de raíz y de arranque:

Warning: Debido a un error en el paquete `e2fsprogs`, necesita utilizar explícitamente la opción `mount -t ext3` si utiliza sistemas de ficheros `ext3`.

Code Listing 15: Montaje de particiones

```
# mount /dev/hda3 /mnt/gentoo
(Para particiones ext3:)
# mount -t ext3 /dev/hda3 /mnt/gentoo

# mkdir /mnt/gentoo/boot
# mount /dev/hda1 /mnt/gentoo/boot
```

Note: Si quiere que su `/tmp` resida sobre una partición diferente, asegúrese de cambiar los permisos después de montarla: `chmod 1777 /mnt/gentoo/tmp`. Lo mismo debe ser aplicado a `/var/tmp`.

También necesitamos montar el sistema de ficheros `proc` (la interfaz virtual del kernel) en `/proc`. Pero primero necesitamos situar nuestros ficheros en las

particiones.

Continué con Instalación de Ficheros de Instalación de Gentoo.

Instalando los archivos de instalación Gentoo

“Las instalaciones de Gentoo funcionan a con los archivos llamados "stage". En este capítulo describimos cómo se extrae un archivo stage y configuramos Portage.”

1: Instalando el Stage comprimido (tarball)

Antes de continuar debes revisar la fecha y la hora y actualizarlos. ¡Un reloj mal configurado puede traer resultados extraños a futuro!

Para comprobar la fecha/hora actual, ejecute el comando `date`:

Code Listing 1: Verificando la fecha/hora

```
# date
Fri Mar 29 16:21:18 CEST 2005
```

Si la fecha/hora está equivocada, actualícela con el comando `date MMDDhhmmAAAA`, con la siguiente sintaxis (**M**es, **D**ía, **h**ora, **m**inuto, **A**ño). Por ejemplo, para colocar la fecha y hora a las 16:21 horas del 29 de Marzo del 2005:

Code Listing 2: Ajustando la fecha/hora

```
# date 032916212005
```

... : Localizando el archivo Stage3

Si ha configurado su red porque necesita descargar un archivo stage3 correspondiente a su arquitectura, continúe con [Alternativa: utilizando un stage3 procedente de Internet](#). De lo contrario lea [Por defecto: utilizar un stage3 del CD de instalación](#).

2: Por defecto: utilizar un stage3 del LiveCD

Los stages del CD se encuentran en el directorio `/mnt/cdrom/stages`. Para ver una lista completa de los stages, utilice `ls`:

Code Listing 3: Comprobando los stages disponibles

```
# ls /mnt/cdrom/stages
```

Si el sistema le devuelve un error, quizá necesite montar el CD-ROM:

Code Listing 4: Montando the CD-ROM

```
# ls /mnt/cdrom/stages
ls: /mnt/cdrom/stages: No such file or directory
# mount /dev/cdroms/cdrom0 /mnt/cdrom
# ls /mnt/cdrom/stages
```

Ahora situese en el punto de montaje de Gentoo (normalmente `/mnt/gentoo`):

Code Listing 5: Cambiando de directorio a /mnt/gentoo

```
# cd /mnt/gentoo
```

Ahora vamos a descomprimir el stage que haya elegido. Esto lo haremos con la herramienta GNU `tar`. ¡Asegúrese de usar las mismas opciones `-xvjpf`!. La `x` se usa para *Desempaquetar*, la `v` para ver que pasa durante el proceso de extracción (Ok, sí, es opcional), la `j` para *Descomprimir con bzip2*, la `p` para *Preservar los permisos* y el `f` para decir que extraemos un archivo, no la entrada estándar. En el siguiente ejemplo, descomprimimos el stage `stage3-<subarch>-2005.1.tar.bz2`. Asegúrese de sustituir el nombre del tarball por su propio stage.

Code Listing 6: Descomprimiendo el stage

```
# tar -xvjpf /mnt/cdrom/stages/stage3-<subarch>-2005.1.tar.bz2
```

Ahora que el stage está instalado, continúe con [Instalando Portage](#).

3: Alternativa: utilizando utilizando un stage procedente

de Internet

Vaya al punto de montaje de Gentoo en donde haya montado los sistemas de archivo (probablemente /mnt/gentoo):

Code Listing 7: Ir al punto de montaje de Gentoo

```
# cd /mnt/gentoo
```

Dependiendo del medio de instalación, tendremos un par de herramientas disponibles para descargar el stage. Si disponemos de [links2](#), podremos navegar por [la lista de servidores réplica de Gentoo](#) [<http://www.gentoo.org/main/en/mirrors.xml>] y escoger el más cercano a nosotros.

Si no tiene [links2](#), debería disponer de [lynx](#). Si necesita pasar a través de un proxy, exporte las variables [http_proxy](#) y [ftp_proxy](#):

Code Listing 8: Configurando la información del proxy para lynx

```
# export http_proxy="http://proxy.server.com:port"  
# export ftp_proxy="http://proxy.server.com:port"
```

A partir de ahora asumiremos que tiene [links2](#) a su disposición.

Luego entre en el directorio `releases/` seguido de tu arquitectura (por ejemplo `x86/` y la versión de Gentoo (2005.1/) para finalizar con el directorio de los stages `stages/`. Allí deberías ver todos los archivos de las stages disponibles para tu arquitectura. Selecciona uno y presiona **D** para descargarlo. Cuando se haya descargado, presiona **Q** para cerrar el navegador.

Code Listing 9: Navegando por la lista de servidores réplica con links2

```
# links2 http://www.gentoo.org/main/en/mirrors.xml  
  
(Si necesita soporte para proxy con links2:)  
# links2 -http-proxy proxy.server.com:8080  
http://www.gentoo.org/main/en/mirrors.xml
```

Si quiere comprobar la integridad del tarball de stage que ha descargado, utilice [md5sum](#) y compare la salida con la suma de comprobación MD5 que proporciona el servidor réplica. Por ejemplo, para comprobar la validez del tarball de stage de

x86:

Code Listing 10: Ejemplo de comprobación de integridad del tarball de stage

```
# md5sum -c stage1-x86-2005.1.tar.bz2.md5
stage1-x86-2005.1.tar.bz2: OK
```

... : Extraer el Stage comprimido

Ahora desempaquetamos el stage descargado en sistema. Usaremos el [tar](#) de GNU para dicha labor y este es el método más fácil.

Code Listing 11: Descomprimiendo el Stage

```
# tar -xvjpf stage3-*.tar.bz2
```

Asegúrese de usar las mismas opciones [-xvjpf](#)). La [x](#) se usa para *Desempaquetar*, la [v](#) para ver que pasa durante el proceso de extracción (Ok, sí, es opcional), el [j](#) para *Descomprimir con bzip2*, el [p](#) para *Preservar los permisos* y el [f](#) para decir que extraemos un archivo, no la entrada estándar.

Ahora que el stage está instalado sigamos con [Instalando Portage](#).

4: Instalando Portage

Ahora tiene que instalar una imagen de Portage, es un conjunto de archivos que informan a Portage sobre los programas que puede instalar, que perfiles están disponibles, etc.

... : Descomprimiendo la imagen del CD de instalación

Para instalar la imagen, eche un vistazo a `/mnt/cdrom/snapshots/` para comprobar que imagen está disponible:

Code Listing 12: Comprobando el contenido de /mnt/cdrom/snapshots

```
# ls /mnt/cdrom/snapshots
```

Ahora extraeremos la imagen siguiendo el siguiente método. De nuevo, asegúrese de usar las opciones correctas para [tar](#). También, la [-C](#) que es mayúscula [C](#), no [c](#).

En el siguiente ejemplo usaremos `portage-<date>.tar.bz2` como nombre de la imagen. Asegúrese de sustituirlo por el nombre de la imagen que tiene en su CD de instalación.

Code Listing 13: Extrayendo la imagen de Portage

```
# tar -xvjf /mnt/cdrom/snapshots/portage-<date>.tar.bz2 -C /mnt/gentoo/usr
```

... : Copiando los archivos de código fuente

También necesita copiar todo el código fuente desde el CD Universal de instalación.

Code Listing 14: Copiando el código fuente

```
# mkdir /mnt/gentoo/usr/portage/distfiles
# cp /mnt/cdrom/distfiles/* /mnt/gentoo/usr/portage/distfiles/
```

5: Configurando las opciones de compilación

Para optimizar Gentoo, tendrás que ajustar un par de variables que afectarán el comportamiento de Portage. Todas estas variables se pueden fijar como variables de entorno (usando `export`) pero eso no es permanente. Para mantener tu configuración, Portage dispone de `/etc/make.conf`, un fichero de configuración para Portage. Este es el fichero que editaremos ahora.

Note: Una lista comentada de todas las posibles variables puede encontrarse en `/mnt/gentoo/etc/make.conf.example`. Para una instalación de Gentoo correcta lo único que necesita es configurar las variables que se mencionan en las siguientes líneas.

Usa tu editor favorito (en esta guía nosotros usaremos `nano`. Así que empezamos con la modificación de las variables.

Code Listing 15: Abriendo /etc/make.conf

```
# nano -w /mnt/gentoo/etc/make.conf
```

Como probablemente te darás cuenta, el fichero `make.conf` está estructurado de una manera genérica: Las líneas comentadas empiezan con "#", otras líneas

definen variables usando la sintaxis `VARIABLE="contenido"`. Discutiremos muchas de esas variables más adelante.

Warning: No haga ninguna modificación en la variable `USE` si va a ejecutar un instalación `stage3` con `GRP`. Puede modificar la variable `USE` después de tener instalados los paquetes que desee. ¡Los Gremlins serán avisados para asaltar su sistema si ignora este aviso!

...: CHOST

Warning: Aunque puede ser interesante para los usuarios que no iniciaron la instalación con el Stage 1, no deberían cambiar el parámetro `CHOST` en el `make.conf` ya que pueden volver el sistema inutilizable. De nuevo, sólo cambie esta variable si usa una instalación desde el `stage1`.

La instalación sin conexión a la red solamente soporta instalaciones desde `stage3`. ¡No toque la variable `CHOST`!

...: CFLAGS y CXXFLAGS

Las variables `CFLAGS` y `CXXFLAGS`, definen los parámetros de optimización para el compilador de C y C++ de `gcc` respectivamente. Aunque generalmente se definen aquí, tendrás el máximo rendimiento si optimizas estos parámetros para cada programa por separado. La razón es que cada programa es diferente.

En el fichero `make.conf` deberás definir los parámetros de optimización que pienses que vayan a hacer tu sistema el mejor *en todas las situaciones*. No coloques parámetros experimentales en esta variable; un nivel demasiado alto de optimización puede hacer que los programas se comporten mal (cuelgues, o incluso peor, funcionamientos erróneos).

No explicaremos todas las opciones posibles para la optimización. Pero si quieres conocerlas todas, léete [El manual en línea de GNU](#) [<http://www.gnu.org/software/gcc/onlinedocs/>] o la página información de `gcc` (`info gcc -- Solo en un sistema Linux funcional`). El fichero `make.conf` también contiene una gran cantidad de ejemplos e información; no olvides leerlo también.

La primera variable es el parámetro `-march=`, que especifica el nombre de la arquitectura seleccionada. Las posibles opciones están descritas en el fichero `make.conf` (como comentarios). Por ejemplo, para la arquitectura x86 Athlon XP:

Code Listing 16: El parámetro march de GCC

```
# Los usuarios de AMD64 que quieran utilizar un sistema nativo de 64 bits deben
utilizar -march=k8
# Los usuarios de EM64T deberían utilizar -march=nocona
-march=athlon-xp
```

Seguida de esta, está el parámetro `-O`, que especifica la clase optimización de `gcc`. Las clases posibles son `s` (para tamaño optimizado), `0` (para no optimizar), `1`, `2` or `3` para la optimización de velocidad (cada clase tiene los mismos parámetros que la primera, más algunos extras). Por ejemplo para una optimización de clase 2:

Code Listing 17: El parámetro O de GCC

```
-O2
```

Otros parámetros de optimización bastante populares son los `-pipe` (usando tuberías en lugar de ficheros temporales para la comunicación entre las diferentes etapas de compilación).

Cuidado con utilizar `-fomit-frame-pointer` (el cual no mantiene el puntero de macro en un registro para las funciones que no lo necesiten) pues podría tener graves repercusiones en la depuración de errores en aplicaciones.

Cuando definimos las variables `CFLAGS` y `CXXFLAGS`, deberías combinar algunos parámetros de optimización, como en el ejemplo siguiente:

Code Listing 18: Definiendo las variables CFLAGS y CXXFLAGS

```
CFLAGS="-march=athlon-xp -pipe -O2" # Los usuarios de AMD64 utilizan march=k8
                                     # Los usuarios de EM64T deberían utilizar
-march=nocona
CXXFLAGS="${CFLAGS}"                # Usa las mismas opciones para las dos
variables
```

..:: MAKEOPTS

Con la variable `MAKEOPTS` definimos cuantas compilaciones paralelas pueden hacerse al mismo tiempo cuando instalamos un paquete. El número sugerido es la cantidad de CPUs de tu sistema, más uno

Code Listing 19: MAKEOPTS para un sistema normal de 1-CPU

```
MAKEOPTS="-j2"
```

. . . : ¡Preparados, listos, ya!

Actualiza tu `/mnt/gentoo/etc/make.conf` con tus propios parámetros y guarda los cambios. Ahora estamos listos para continuar con [Instalando el sistema base de Gentoo](#).

Instalando el sistema base de Gentoo

“Independientemente del archivo stage que escoja, el resultado es que se dispone de un sistema base Gentoo. Este capítulo describe cómo llegar a esta etapa.”

1: Chrooting

Monte el sistema de ficheros `/proc` en `/mnt/gentoo/proc` para permitir a la instalación utilizar la información proporcionada por el kernel incluso dentro del entorno `chroot`.

Code Listing 1: Montando /proc

```
# mount -t proc none /mnt/gentoo/proc
```

. . . : Opcional: Copiar la información DNS

Si ha configurado su red para descargar posteriormente el stage correspondiente desde Internet, necesita copiar la información DNS guardada en `/etc/resolv.conf` a `/mnt/gentoo/etc/resolv.conf`. Este archivo contiene los nombres de los servidores que su sistema utiliza para resolver los nombres y las direcciones IP.

Code Listing 2: Copiando los DNS

```
# cp -L /etc/resolv.conf /mnt/gentoo/etc/resolv.conf
```

. . . : Entrando al nuevo entorno

Ahora que todas las particiones están inicializadas y el sistema base instalado, es hora de entrar en nuestro nuevo entorno de instalación *chrooting*. Esto significa pasar desde el actual entorno de instalación (LiveCD o otro medio de instalación) hacia tu entorno de instalación (o sea, las particiones inicializadas).

El cambio de raíz se hace en tres pasos. Primero cambiamos la raíz desde `/` (en el medio de instalación) a `/mnt/gentoo` (en tus particiones) usando `chroot`.

Después crearemos un nuevo entorno usando `env-update`, el cual, en esencia crea las variables de entorno. Finalmente, cargamos esas variables en memoria tecleando `source`.

Code Listing 3: Entrando al nuevo entorno

```
# chroot /mnt/gentoo /bin/bash
# env-update
  * Caching service dependencies...
# source /etc/profile
```

¡Enhorabuena! Estás dentro de tu nuevo entorno Gentoo Linux. Por supuesto aún no hemos terminado, todavía quedan unas cuantas secciones ;)

2: Configurando la variable USE

`USE` es una de las variables más importantes que Gentoo proporciona a sus usuarios. Muchos programas pueden ser compilados con o sin soporte opcional para ciertas cosas. Por ejemplo, algunos programas pueden ser compilados con soporte `gtk`, o con soporte `qt`. Otros programas pueden ser compilados con o sin soporte `SSL`. Algunos programas pueden ser compilados con soporte `framebuffer` en lugar de soporte `X11` (servidor X).

Muchas distribuciones compilan sus paquetes con el mayor soporte posible, aumentando el tamaño de los programas y su tiempo de carga, sin mencionar una cantidad enorme de dependencias. Con Gentoo puedes definir con que opciones debe ser compilado un paquete. Ahí es donde actúa la variable `USE`.

En la variable `USE` definimos palabras clave que son mapeadas a opciones de compilación. Por ejemplo `ssl` compilará los programas que lo requieran con soporte `ssl`. `-X` quitará el soporte para el servidor X (nótese el signo menos delante). `gnome gtk -kde -qt` compilará tus programas con soporte para `gnome` y `gtk`, pero sin soporte para `kde` (y `qt`), haciendo tu sistema completamente compatible con `GNOME`.

. . . : Modificando la variable USE

Warning: No haga ninguna modificación en la variable USE si va a ejecutar un instalación stage3 con GRP. Puede modificar la variable USE después de tener instalados los paquetes que desee. ¡Los Gremlins serán avisados para asaltar su sistema si ignora este aviso!.

Los valores por defecto de la variable USE se encuentran en `make.defaults`, archivos de su perfil. Encontrará los archivos `make.defaults` en el directorio al cual apunte `/etc/make.profile` y todos sus directorios padres. El valor predeterminado de configuración de la variable USE es la suma de todas las configuraciones de USE en todos los archivos `make.defaults`. Lo que modifique en `/etc/make.conf` se calcula contra estos valores. Si pone algún valor en su USE, es añadido a la lista por defecto. Si elimina algo en su variable USE, poniéndole un signo menos delante, es eliminado de la lista por defecto (si estaba en ella claro). *Nunca* cambie nada en `/etc/make.profile` ya que se sobrescribirá cuando actualice Portage!

Puede encontrar una descripción más amplia sobre la variable USE en la segunda parte del Manual de Gentoo [Capítulo 1: Variables USE](#). Encontrará una descripción más extensa sobre las opciones de la variable USE en su sistema, en `/usr/portage/profiles/use.desc`.

Code Listing 4: Viendo las opciones disponibles

```
# less /usr/portage/profiles/use.desc
(Puede desplazarse arriba y abajo utilizando sus teclas de flechas y salir
pulsando 'q')
```

Como ejemplo, te mostramos unas opciones USE para un sistema basado en KDE con DVD, ALSA y soporte para grabar CD's.

Code Listing 5: Abriendo /etc/make.conf

```
# nano -w /etc/make.conf
```

Code Listing 6: Configurando la variable USE

```
USE="-gtk -gnome qt kde dvd alsa cdr"
```

Configurando el kernel

“El kernel Linux kernel es el núcleo de cada distribución de Linux. Este capítulo explica cómo configurarlo.”

1: Zona Horaria

Primero necesita seleccionar su zona horaria para que su sistema sepa dónde está localizado. Busque su zona horaria en `/usr/share/zoneinfo`, luego haga un vínculo simbólico a `/etc/localtime` usando `ln`:

Code Listing 1: Configurar la Zona Horaria

```
# ls /usr/share/zoneinfo  
(Suponga que quiere usar el GMT)  
# ln -sf /usr/share/zoneinfo/GMT /etc/localtime
```

2: Instalar las Fuentes

El punto alrededor del cual se construyen todas las distribuciones es el Kernel de Linux. Es la capa entre los programas de usuario y el hardware del sistema. Gentoo proporciona a sus usuarios varias fuentes de kernel. Una lista completa está disponible en la [Guía Gentoo del Kernel](http://www.gentoo.org/doc/es/gentoo-kernel.xml) [http://www.gentoo.org/doc/es/gentoo-kernel.xml].

Para los sistemas basados en x86 tenemos, el kernel que recibe mayor soporte es conocido como [gentoo-sources](#). Este kernel está basado en las fuentes oficiales pero tiene parches aplicados que mejoran su seguridad, estabilidad, compatibilidad y arreglan errores. De manera alternativa, las fuentes de Linux, tal cual, sin parchear se suministran en un paquete bajo el nombre de [vanilla-sources](#).

Ambas fuentes del kernel están basadas en las fuentes oficiales del kernel 2.6. Si quiere utilizar un kernel basado en 2.4, necesitará instalar Gentoo con una conexión a Internet activa ya que no suministramos estas fuentes en nuestro CD de instalación

Elija sus fuentes de kernel e instálelas usando [emerge](#).

Code Listing 2: Instalar unas fuentes del kernel

```
# emerge gentoo-sources
```

Cuando vea en `/usr/src` verá un enlace simbólico llamado `linux` apuntando a las fuentes de su kernel. Se asume que las fuentes instaladas son las [gentoo-sources-2.6.11-r3](#):

Code Listing 3: Ver el enlace a las fuentes del kernel

```
# ls -l /usr/src/linux
lrwxrwxrwx  1 root  root           12 Oct 13 11:04 /usr/src/linux ->
linux-2.6.11-gentoo-r3
```

Si el enlace simbólico no apunta a las fuentes del kernel que ha elegido (tenga en cuenta que [linux-2.6.11-gentoo-r3](#) es un mero ejemplo) debería cambiarlo para que lo haga a las del kernel correcto.

Code Listing 4: Cambiar el en enlace a las fuentes del kernel

```
# rm /usr/src/linux
# cd /usr/src
# ln -s linux-2.6.11-gentoo-r3 linux
```

Ahora vamos a configurar y compilar las fuentes del kernel. Puede usar para esto [genkernel](#), que construirá un kernel genérico como el usado por el CD de instalación. Aquí explicaremos la configuración "manual", ya que es la mejor manera de optimizar su entorno.

Si quiere configurar manualmente su kernel, continúe con [Predeterminado: Configuración Manual](#). En cambio, si quiere usar [genkernel](#) debe leer [Alternativa: Usar genkernel](#).

3: Predeterminado: Configuración Manual

Configurar manualmente un kernel frecuentemente se ve como el procedimiento más difícil que tiene que realizar un usuario de Linux. Nada mas lejos de la verdad -- después de configurar un par de kernels no recordará si fue difícil ;)

Sin embargo, una cosa es cierta: debe conocer su sistema cuando empiece a

configurar su kernel manualmente. Mucha información se puede recolectar instalando `pciutils` (`emerge pciutils`) el cual contiene `lspci`. Ahora será capaz de utilizar `lspci` en el entorno `chroot`. Puede ignorar tranquilamente los avisos sobre `pcilib` (como `pcilib: cannot open /sys/bus/pci/devices`) que `lspci` le muestre. Alternativamente, puede ejecutar `lspci` desde un entorno `no-chroot`. Los resultados son los mismos. `lspci` si está disponible). También puede ejecutar `lsmod` para ver que módulos del kernel usa el CD de instalación (puede proporcionarle buenos consejos sobre qué habilitar).

Ahora vaya al directorio de sus fuentes y ejecute `make menuconfig`. Esto lanzará un menú de configuración basado en `ncurses`.

Code Listing 5: Invocar a menuconfig

```
# cd /usr/src/linux
# make menuconfig
```

Le darán la bienvenida varias secciones de configuración. Listaremos primero algunas opciones que debe activar (de otro modo Gentoo no funcionará, o no funcionará adecuadamente sin configuración adicional).

. . . : Activar Opciones Necesarias

Asegúrese de que todos los controladores vitales para el arranque del sistema (como pueden ser los de SCSI, ...) están compilados *dentro* del kernel y no como módulos o, de lo contrario, su sistema no será capaz de arrancar correctamente.

Seleccione la familia correcta de procesadores:

Code Listing 6: Soporte General y familia de procesador

```
General setup --->
                [*] Support for hot-pluggable devices
Processor type and features --->
  (Selecciónelo de acuerdo a su sistema)
  (Athlon/Duron/K7) Processor family
```

Vaya ahora a [File Systems](#) y seleccione los soportes para los sistemas de archivos que use. No los compile como módulos, de otro modo su sistema Gentoo no será capaz de montar sus particiones. También seleccione [/proc file system](#), [Virtual memory](#). No seleccione [/dev file system](#):

Code Listing 7: Seleccionar los sistemas de archivos necesarios

```
File systems --->
Pseudo Filesystems --->
  <*> /proc file system support
  < > /dev file system support (OBSOLETE)
  <*> Virtual memory file system support (former shm fs)

(Seleccione una o mas de las siguientes opciones según las necesite su sistema)
  <*> Reiserfs support
  <*> Ext3 journalist file system support
  <*> JFS filesystem support
  <*> Second extended fs support
  <*> XFS filesystem support
```

No olvide activar el DMA para sus unidades:

Code Listing 8: Activating DMA

```
Device Drivers --->
  ATA/ATAPI/MFM/RLL support --->
    [*] Generic PCI bus-master DMA support
    [*] Use PCI DMA by default when available
```

Si está usando PPPoE para conectarse a Internet o está usando un módem dial-up, necesitará las siguientes opciones en el kernel:

Code Listing 9: Seleccionar los controladores PPPoE necesarios

```
Device Drivers --->
  Networking support --->
    <*> PPP (point-to-point protocol) support
    <*> PPP support for async serial ports
    <*> PPP support for sync tty ports
```

Las dos opciones de compresión no lo afectan pero no son necesarias, ni para la opción [PPP over Ethernet](#), tal vez solo sea usada por [rp-pppoe](#) cuando configure un kernel en modo PPPoE.

Si lo requiere, no olvide incluir el soporte en el kernel para su tarjeta ethernet:

Si tiene un CPU Intel que soporte HyperThreading (tm), o tiene un sistema multi-CPU, debe activar "Symmetric multi-processing support":

Code Listing 10: Activar el soporte SMP

```
Processor type and features --->
  <*> Symmetric multi-processing support
```

Si usa dispositivos de entrada USB (como un ratón o teclado) no olvide activarlos

también:

Code Listing 11: Activar Soporte para Dispositivos de Entrada USB

```
Device Drivers --->
USB Support --->
  <*> USB Human Interface Device (full HID) support
  [*] HID input layer support
```

Los usuarios de portátiles que quieran soporte PCMCIA *no deben* usar los controladores PCMCIA si eligen usar un kernel 2.4. Controladores más recientes están disponibles a través del paquete [pcmcia-cs](#) que será instalado después. Los usuarios de kernel 2.6 sin embargo, si deben usar los controladores PCMCIA desde el kernel.

A la par que compila el soporte PCMCIA en el kernel 2.6, no olvide activar el soporte para el puente de tarjetas PCMCIA que se encuentre en su sistema:

Code Listing 12: Activando el soporte PCMCIA para kenels 2.6

```
Bus options (PCI, PCMCIA, EISA, MCA, ISA) --->
PCCARD (PCMCIA/CardBus) support --->
  <*> PCCard (PCMCIA/CardBus) support
  (Seleccione 16 bit si necesita soporte para las antiguas tarjetas PCMCIA. La
  mayoría de la gente querrá utilizar esto.)
  <*> 16-bit PCMCIA support
  [*] 32-bit CardBus support
  (Seleccione los puentes adecuados a continuación)
  --- PC-card bridges
  <*> CardBus yenta-compatible bridge support (NEW)
  <*> Cirrus PD6729 compatible bridge support (NEW)
  <*> i82092 compatible bridge support (NEW)
  <*> i82365 compatible bridge support (NEW)
  <*> Databook TCIC host bridge support (NEW)
```

...: : Compilar e Instalar

Ahora que ya está configurado su kernel, es hora de compilarlo e instalarlo. Salga de la configuración y comience con el proceso de compilación:

Code Listing 13: Compilar el kernel

```
# make && make modules_install
```

Cuando el kernel ha terminado de compilar, copie la imagen a /boot. De aquí en adelante asumimos que el kernel que está instalando es la versión 2.6.11-r3 de las [gentoo-sources](#). Use el nombre que le parezca apropiado y recuerde que lo

necesitará después cuando configure el gestor de arranque.

Code Listing 14: Instalar el kernel

```
# cp arch/i386/boot/bzImage /boot/kernel-2.6.11-gentoo-r3
```

Ahora continúe con [Configurar los módulos del kernel](#).

4: Alternativa: Usar genkernel

Si está leyendo esta sección, eligió usar nuestro guión [genkernel](#) para configurar el kernel.

Ahora que el árbol de las fuentes del kernel está instalado, es hora de compilarlo usando nuestro guión [genkernel](#) que automáticamente construirá un kernel. [genkernel](#) trabaja configurando un kernel prácticamente idéntico al kernel de nuestro CD de instalación. Esto significa que cuando use [genkernel](#) para construir su kernel, su sistema generalmente detectará todo su hardware en el arranque, tal como lo hace el CD de instalación. Debido a que [genkernel](#) no requiere ninguna configuración manual del kernel, es una solución ideal para esos usuarios que no se sienten cómodos compilando sus propios kernels.

Ahora, veamos como usar [genkernel](#). Primero, la instalación del ebuild de [genkernel](#):

Code Listing 15: Instalar genkernel

```
# emerge genkernel
```

Ahora, compile las fuentes del kernel ejecutando [genkernel all](#). Recuerde que como [genkernel](#) compila un kernel que soporta casi todo el hardware, esta compilación ¡tardará un rato en terminar!

Observe que si su partición de arranque no usa ext2 o ext3 como sistema de archivos, necesitará configurar manualmente su kernel usando [genkernel --menuconfig all](#) y agregar soporte para su sistema de archivos *en* el kernel (*no* como módulo). Los usuarios de EVMS2 o LVM2 probablemente querrán añadir también los argumentos [--evms2](#) or [--lvm2](#).

Code Listing 16: Ejecutar genkernel

```
# genkernel all
```

Una vez que **genkernel** haya terminado, un kernel, un conjunto completo de módulos y un *disco raíz de inicio* (initrd) serán creados. Usaremos el kernel e initrd para configurar un gestor de arranque mas tarde en este documento. Escriba los nombres del kernel y de initrd ya que los necesitará para el archivo de configuración del gestor de arranque. El initrd iniciará inmediatamente después del arranque para realizar la autodetección de hardware (tal como en el CD de instalación) antes que inicie su sistema "real".

Code Listing 17: Verificar los nombres del kernel e initrd creados

```
# ls /boot/kernel* /boot/initrd*
```

Si se quiere que el sistema se parezca al CD de instalación, se debería, cuando termine la instalación, instalar **coldplug**. Mientras initrd autodetecta el hardware necesario para arrancar su sistema, **coldplug** autodetecta todo lo demás.

Code Listing 18: Instalar y habilitar coldplug

```
# emerge coldplug  
# rc-update add coldplug default
```

5: Configurar los módulos del kernel

Debe listar los módulos que quiera cargar automáticamente en `/etc/modules.autoload.d/kernel-2.6` (o en `kernel-2.4`). Puede agregar opciones extras a los módulos si lo desea.

Para ver todos los módulos disponibles, ejecute el comando **find**. No olvide sustituir "`<versión del kernel>`" con la versión del kernel que acaba de compilar:

Code Listing 19: Ver todos los módulos disponibles

```
# find /lib/modules/<versión del kernel>/ -type f -iname '*.o' -or -iname '*.ko'
```

Por ejemplo, para cargar automáticamente el módulo **3c59x.o**, edite el archivo `kernel-2.6` (o `kernel-2.4`) y escriba el nombre del módulo ahí.

Code Listing 20: Editar /etc/modules.autoload.d/kernel-2.6

```
(Ejemplo para un kernel 2.6)  
# nano -w /etc/modules.autoload.d/kernel-2.6
```

Code Listing 21: /etc/modules.autoload.d/kernel-2.6

```
3c59x
```

Continúe la instalación con [Configurar su Sistema](#).

Configurando su sistema

“Hará falta modificar unos archivos de configuración importantes. En este capítulo verá cuáles son estos archivos y cómo proceder.”

1: Información del Sistema de Ficheros

En Linux, todas las particiones usadas por el sistema deben estar reflejadas en `/etc/fstab`. Este fichero contiene los puntos de montaje de esas particiones (donde se encuentran en la estructura del sistema de ficheros), cómo deben ser montadas y con que opciones especiales (automáticamente o no, si los usuarios pueden montarlas o no, etc.).

... : Creando el `/etc/fstab`

`/etc/fstab` usa una sintaxis especial. Cada línea está formada por seis campos, separados por espacios en blanco (espacio(s), tabuladores o una combinación). Cada campo tiene su propio significado:

- El primer campo muestra la **partición** descrita (la ruta al fichero de dispositivo)
- El segundo campo muestra el **punto de montaje** donde la partición debe montarse
- El tercer campo muestra el **sistema de ficheros** usado por la partición
- El cuarto campo muestra las **opciones de montaje** usadas por `mount` cuando trata de montar la partición. Como cada sistema de ficheros tiene sus propias opciones de montaje, le animamos a leer la página man de mount (`man mount`) para un listado completo. Cuando existen múltiples opciones se separan por comas.
- El quinto campo es usado por `dump` para determinar si la partición requiere ser volcada o no. En general puede dejar esto como `0` (cero).
- El sexto campo es usado por `fsck` para determinar el orden en que los sistemas de ficheros deben ser **comprobados** si el sistema no se apagó correctamente. La partición raíz debe tener un `1` mientras que el resto puede tener `2` (o `0` en el caso en que la comprobación del sistema de ficheros no sea necesaria).

El archivo `/etc/fstab` que proporciona Gentoo de manera predeterminada *no es un archivo `fstab` válido*, así que ejecute `nano` (o su editor favorito) para crear su propio `/etc/fstab`:

Code Listing 1: Abriendo `/etc/fstab`

```
# nano -w /etc/fstab
```

Vamos a ver como anotaremos las opciones para la partición `/boot`. Esto es solo un ejemplo, así que si su arquitectura no requiere una partición `/boot` (como por ejemplo **PPC**), no lo copie al pie de la letra.

En nuestro ejemplo de particionamiento estándar para x86, `/boot` es la partición `/dev/hda1`, con un sistema de ficheros `ext2`. Esta necesita ser comprobada durante el arranque. Entonces escribiríamos:

Code Listing 2: Un ejemplo de línea `/boot` para `/etc/fstab`

```
/dev/hda1 /boot ext2 defaults 1 2
```

Algunos usuarios no quieren que su partición `/boot` sea montada automáticamente para mejorar la seguridad de su sistema. Estos usuarios deberían sustituir `defaults` por `noauto`. Esto implica la necesidad de montar manualmente la partición cada vez que se quiera usarla.

Para aumentar el rendimiento, la mayoría de usuarios podrían querer agregar la opción `noatime` como opción de montaje, que desemboca en un sistema más rápido, puesto que los tiempos de acceso no son registrados (de todas formas, no necesitará esto en general):

Code Listing 3: Una línea `/boot` mejorada para `/etc/fstab`

```
/dev/hda1 /boot ext2 defaults,noatime 1 2
```

Si seguimos estos pasos, acabaríamos con las siguientes tres líneas (para `/boot`, `/` y la partición de swap):

Code Listing 4: Tres líneas del `/etc/fstab`

```
/dev/hda1 /boot ext2 defaults,noatime 1 2
/dev/hda2 none swap sw 0 0
/dev/hda3 / ext3 noatime 0 1
```

Para finalizar, debería agregar una línea para `/proc`, `tmpfs` (requerido) y sus dispositivos CD-ROM (y por supuesto, si tiene otras particiones o dispositivos,

para ellos también):

Code Listing 5: Un ejemplo de /etc/fstab completo

```
/dev/hda1 /boot ext2 defaults,noatime 1 2
/dev/hda2 none swap sw 0 0
/dev/hda3 / ext3 noatime 0 1

none /proc proc defaults 0 0
none /dev/shm tmpfs nodev,nosuid,noexec 0 0

/dev/cdroms/cdrom0 /mnt/cdrom auto noauto,user 0 0
```

auto provoca que **mount** intente adivinar el sistema de archivos (se recomienda para los dispositivos extraíbles ya que pueden ser creados con distintos sistemas de ficheros) y **user** hace posible a los usuarios que no pertenezcan a root monten el CD.

Ahora haga uso del ejemplo anterior, para crear su `/etc/fstab`. Si es un usuario de SPARC, además debería añadir la siguiente línea a su `/etc/fstab`:

Code Listing 6: Añadiendo el sistema de ficheros openprom al /etc/fstab

```
none /proc/openprom openpromfs defaults 0 0
```

Repase su `/etc/fstab`, guarde los cambios y salga para continuar.

2: Información de red

Una de las elecciones que un usuario ha de hacer es el nombre de su PC. Esto parece muy fácil, pero *muchos* usuarios tienen dificultades eligiendo el nombre apropiado para su PC-Linux. Para acelerar las cosas, sepa que el nombre que elija puede cambiarlo mas tarde. Para el caso que nos preocupa, usted puede llamar su sistema simplemente **tux** y su dominio **redcasera**.

Usaremos esos nombres en los siguientes ejemplos. Primero ajustaremos el nombre de host:

Code Listing 7: Ajustando el nombre de Host

```
# nano -w /etc/conf.d/hostname  
  
(Configure la variable HOSTNAME con el nombre del host)  
HOSTNAME="tux"
```

En segundo lugar ajustaremos el nombre del dominio:

Code Listing 8: Ajustando el nombre de Host

```
# nano -w /etc/conf.d/domainname  
  
(Configure la variable DNSDOMAIN con su nombre de dominio)  
DNSDOMAIN="homenetwork"
```

Si dispone de un dominio NIS (si no sabe lo que es, entonces no lo tiene), necesita definirlo también:

Code Listing 9: Ajustando el nombre del dominio NIS

```
# nano -w /etc/conf.d/domainname  
  
(Configura la variable NISDOMAIN con su nombre de dominio NIS)  
NISDOMAIN="my-nisdomain"
```

Ahora añade el script `domainname` al nivel de arranque por defecto:

Code Listing 10: Agregando domainname al nivel de arranque por defecto

```
# rc-update add domainname default
```

... : Configurando su Red

Antes de llegar a experimentar esa sensación "Hey, ya lo tengo todo", debes recordar que la red que configuraste en el inicio de la instalación de Gentoo fue tan solo para la instalación. A partir de ahora vamos a configurar la red permanentemente para su sistema Gentoo.

Note: Información más detallada sobre redes, incluyendo temas más avanzados como bonding, bridging, 802.11q VLANs o redes inalámbricas está disponible en la sección [Configuración de redes en Gentoo](#).

Toda la información de red esta reunida en `/etc/conf.d/net`. Este fichero usa una directa, aunque no intuitiva sintaxis si no se sabe como configurar una red

manualmente. Pero no se asuste, se lo explicaremos todo. Un ejemplo completamente comentado que cubre los distintos tipos de configuraciones, puede encontrarse en `/etc/conf.d/net.example`.

DHCP se emplea por defecto y no requiere ningún tipo de configuración.

Si se necesita configurar la conexión de red porque es necesario incluir opciones de DHCP específicas o porque no utiliza DHCP, abra `/etc/conf.d/net` con su editor favorito ([nano](#) se emplea en este ejemplo):

Code Listing 11: Abriendo `/etc/conf.d/net` para su edición

```
# nano -w /etc/conf.d/net
```

Verá el siguiente fichero:

Code Listing 12: Archivo `/etc/conf.d/net` por defecto

```
# This blank configuration will automatically use DHCP for any net.*
# scripts in /etc/init.d. To create a more complete configuration,
# please review /etc/conf.d/net.example and save your configuration
# in /etc/conf.d/net (this file :)!).
```

Para introducir su propia dirección IP, máscara de red y pasarela, necesita configurar tanto [config_eth0](#) como [routes_eth0](#):

Code Listing 13: Configurar manualmente la información IP para `eth0`

```
config_eth0=( "192.168.0.2 netmask 255.255.255.0" )
routes_eth0=( "default gw 192.168.0.1" )
```

Para utilizar DHCP y añadir opciones específicas, defina, [config_eth0](#) y [dhcp_eth0](#):

Code Listing 14: Obtener una dirección IP automáticamente para `eth0`

```
config_eth0=( "dhcp" )
dhcp_eth0="nodns nonntp nonis"
```

Por favor, consulte `/etc/conf.d/net.example` para conocer una lista con todas las opciones disponibles.

Si tiene varias interfaces de red, repita los pasos anteriores utilizando [config_eth1](#), [config_eth2](#), etc.

Ahora guarde la configuración y salga para continuar.

... : Inicio automático de red en el arranque

Para disponer de su interfaz de red activada en el arranque, necesita agregarla al nivel de ejecución por defecto (default). Si dispone de interfaces PCMCIA debe saltarse este paso puesto que las interfaces PCMCIA son iniciadas por el script de inicialización PCMCIA

Code Listing 15: Agregando net.eth0 al nivel de ejecución default

```
# rc-update add net.eth0 default
```

Si tiene distintas interfaces de red, necesitará crear los apropiados `net.eth1`, `net.eth2` etc. scripts de inicio. Puede usar `ln` para hacer esto:

Code Listing 16: Creando scripts de inicio adicionales

```
# cd /etc/init.d
# ln -s net.eth0 net.eth1
# rc-update add net.eth1 default
```

... : Anotando la Información de la Red

Necesita informar a Linux sobre su red. Esto se define en `/etc/hosts` y ayuda a transformar los nombres de host a direcciones IP para aquellas máquinas que no se resuelven a través de su servidor de nombres (DNS). Por ejemplo, si su red interna consiste en tres PCs llamados `jenny` (192.168.0.5), `benny` (192.168.0.6) y `tux` (192.168.0.7 - este sistema) abra `/etc/hosts` y complete los valores:

Code Listing 17: Abriendo /etc/hosts

```
# nano -w /etc/hosts
```

Code Listing 18: Completando la información de red

```
127.0.0.1    localhost
192.168.0.5  jenny.redlocal jenny
192.168.0.6  benny.redlocal benny
192.168.0.7  tux.redlocal  tux
```

Si su sistema es el único presente (o los servidores de nombres se encargan de todas las resoluciones) una única línea es suficiente. Por ejemplo, si quiere llamar a su sistema `tux`:

Code Listing 19: /etc/hosts para un solo PC o varios plenamente integrados

```
127.0.0.1    localhost tux
```

Guarde y salga del editor para continuar.

Si no tiene PCMCIA, puede continuar con [Información del Sistema](#). Los usuarios de PCMCIA deberían leer el siguiente apartado sobre PCMCIA.

. . . : Opcional: Hacer funcionar el PCMCIA

Note: `pcmcia-cs` está solamente disponible para plataformas x86, amd64 y ppc.

Los usuarios de PCMCIA deben instalar primero el paquete `pcmcia-cs`. Esto también incluye a los usuarios que estén trabajando con un kernel 2.6 (incluso aunque no quieran utilizar los controladores PCMCIA de este paquete). La `USE="-X"` es necesaria para evitar la instalación de `xorg-x11` en este momento.

Code Listing 20: Instalando `pcmcia-cs`

```
# USE="-X" emerge pcmcia-cs
```

Cuando `pcmcia-cs` esté instalado, agregue `pcmcia` al nivel de arranque *default*.

Code Listing 21: Agregando `pcmcia` al nivel de arranque por defecto

```
# rc-update add pcmcia default
```

3: Información del sistema

Primero fijamos la contraseña de administrador escribiendo:

Code Listing 22: Configurando la contraseña de administrador

```
# passwd
```

Si desea que el administrador sea capaz de entrar en el sistema a través de un consola de serie (`tts`), añada `tts/0` al fichero `/etc/securetty`:

Code Listing 23: Añadiendo `tts/0` a `/etc/securetty`

```
# echo "tts/0" >> /etc/securetty
```

... : Información del sistema

Gentoo usa el fichero `/etc/rc.conf` para una configuración general del sistema. Abra el `/etc/rc.conf` y disfrute de todos los comentarios que hay en él :)

Code Listing 24: Abriendo el fichero `/etc/rc.conf`

```
# nano -w /etc/rc.conf
```

Cuando haya terminado de configurar `/etc/rc.conf`, guarde los cambios y salga.

Como puede ver, este fichero está bien comentado para ayudarle a ajustar las variables de configuración necesarias. Puede configurar el sistema para que utilice unicode y definir su editor predeterminado y su gestor de inicio (cómo `gdm` o `kdm`).

Gentoo utiliza `/etc/conf.d/keymaps` para gestionar la configuración del teclado. Edítelo y configure su teclado.

Code Listing 25: Abrir `/etc/conf.d/keymaps`

```
# nano -w /etc/conf.d/keymaps
```

Tenga especial cuidado con **KEYMAP**: si selecciona un **KEYMAP** incorrecto, conseguirá extraños resultados escribiendo en su teclado.

Note: Los usuarios de sistemas SPARC basados en USB y los clones de SPARC podrían necesitar seleccionar un mapeo del teclado `i386` (como "su") en vez de utilizar "sunkeymap". **PPC** utiliza mapeos de teclado `x86` en la mayoría de sus sistemas. Los usuarios que quieren disponer de mapeos ADB en el arranque, necesitan activar "ADB keycode sendings" en su kernel y tener configurado un mapeo de teclado `mac/ppc` en `/etc/conf.d/keymaps`.

Cuando termine de configurar `/etc/conf.d/keymaps`, guarde los cambios y salga.

Gentoo utiliza `/etc/conf.d/clock` para fijar las opciones del reloj. Edítelo de acuerdo a sus necesidades.

Code Listing 26: Abriendo /etc/conf.d/clock

```
# nano -w /etc/conf.d/clock
```

Si su reloj hardware no está utilizando UTC, necesita añadir `CLOCK="local"` al archivo. De lo contrario notará como su reloj no funciona correctamente.

Cuando haya acabado de configurar `/etc/conf.d/clock`, guárdelo y salga.

Si no está instalando Gentoo en un IBM IBM POWER5 o un sistema JS20 system, continúe con [Instalando las herramientas de sistema necesarias](#) .

.. : : Configurar la consola

Note: La siguiente sección se aplica a las plataformas IBM POWER5 y JS20.

Si está ejecutando Gentoo en un LPAR o en un JS20 blade, debería descomentar la línea `hvc` en `/etc/inittab` para que la consola virtual muestre un símbolo de entrada al sistema (prompt).

Code Listing 27: Activando el soporte `hvc` o `hvs` en `/etc/inittab`

```
hvc0:12345:respawn:/sbin/agetty -L 9600 hvc0
hvs0:12345:respawn:/sbin/agetty -L 19200 hvs0
```

Debería tomar el tiempo necesario para comprobar la consola apropiada que se lista en `/etc/securetty`.

Puede continuar con [Instalando las herramientas de sistema necesarias](#)

.

Instalando herramientas necesarias para el sistema

“Como hemos mencionado antes, Gentoo representa la posibilidad de escoger. En este capítulo ayudaremos a escoger e instalar algunas herramientas importantes.”

1: Bitácora del Sistema

Cuando mencionamos que era la *stage3*, dijimos que contenía todas las herramientas del sistema necesarias para las cuales no podíamos proporcionar una opción para nuestro usuarios. También dijimos que instalaríamos otras herramientas mas tarde. Bueno, aquí están :)

La primera herramienta por la que tiene que decidirse es la que proporciona el registro y las bitácoras para su sistema. Unix y Linux tienen una excelente historia en sus capacidades de registros -- si lo quisiera podría registrar todo lo que pasa en su sistema en bitácoras. Esto sucede con el *registro del sistema*.

Gentoo ofrece varios sistemas de registro para elegir. Están [sysklogd](#), que es el conjunto tradicional de demonios de bitácoras, [sysklogd](#), un sistema de bitácora avanzado, y [metalog](#) que es una bitácora de sistemas altamente configurable. También puede haber otros en el Portage - el número de paquetes disponibles crece día a día.

Si está pensando utilizar [sysklogd](#) o [syslog-ng](#) quizá quiera instalar posteriormente [logrotate](#) ya que esos logeadores no proporcionan ningún mecanismo de rotación para los archivos de log.

Para instalar la bitácora del sistema de su elección, use [emerge](#) y agrégelo al nivel de arranque predeterminado usando [rc-update](#). El siguiente ejemplo instala [syslog-ng](#). Desde luego, sustitúyalo por el sistema de bitácora de su elección:

Code Listing 1: Instalar un sistema de bitácoras

```
# emerge syslog-ng
# rc-update add syslog-ng default
```

2: Opcional: Demonio Cron

El siguiente es el demonio cron. Aunque es opcional y no lo requiere su sistema es recomendable instalar uno. ¿Pero que es un demonio cron? Un demonio cron ejecuta comandos en horarios planificados. Es muy cómodo si necesita ejecutar comandos regularmente (por ejemplo diario, cada semana o mensualmente).

Gentoo ofrece tres posibles demonios cron: [dcron](#), [fcron](#) y [vixie-cron](#). Instalar cualquiera es similar a instalar un sistema de bitácoras. Sin embargo [dcron](#) y [fcron](#) requieren un comando extra de configuración, que es [crontab /etc/crontab](#). Si no sabe cuál escoger, use [vixie-cron](#).

Solo proporcionamos [vixie-cron](#) para instalaciones sin red. Si quiere otro demonio cron puede esperar e instalarlo más tarde.

Code Listing 2: Instalar un demonio cron

```
# emerge vixie-cron
# rc-update add vixie-cron default
(Sólo si optó por dcron o fcron) # crontab /etc/crontab
```

3: Opcional: Indexar Archivos

Si quiere crear un índice de su sistema de archivos para habilitar su rápida localización usando la herramienta [locate](#), necesita instalar [sys-apps/slocate](#).

Code Listing 3: Instalar slocate

```
# emerge slocate
```


4: Herramientas del Sistema de Archivos

Dependiendo en que sistema de archivos este usando, necesita instalar las utilerías necesarias (para verificar la integridad del sistema de archivos, crear adicionales, etc.).

La siguiente tabla lista las herramientas que necesita instalar según un sistema de archivos determinado. No todos los sistemas de ficheros están disponibles para cada una de las arquitecturas.

<i>Sistema de Archivos</i>	<i>Herramienta</i>	<i>Comando para Instalar</i>
XFS	xfsprogs	<code>emerge xfsprogs</code>
ReiserFS	reiserfsprogs	<code>emerge reiserfsprogs</code>
JFS	jfsutils	<code>emerge jfsutils</code>

Si es usuario de EVMS, necesita instalar también [evms](#):

Code Listing 4: Instalar herramientas EVMS

```
# emerge evms
```

Si no necesita ninguna herramienta adicional relacionada con la red (como `rp-pppoe` o un cliente `dhcp`) continúe con [Configuración del Cargador de Arranque](#).

5: Herramientas de Red

Si necesita que Gentoo obtenga automáticamente una dirección IP para sus interfaces de red, necesita instalar en su sistema [dhcpcd](#) (o cualquier otro cliente DHCP) . Si no lo hace, ¡tal vez no sea capaz de conectarse a Internet después de la instalación!

Code Listing 5: Instalar dhcpcd

```
# emerge dhcpcd
```

. . . : Opcional: Instalar un Cliente PPPoE

Si necesita [rp-pppoe](#) para conectarse a la red, necesita instalarlo.

Code Listing 6: Instalar rp-pppoe

```
# USE="-X" emerge rp-pppoe
```

El `USE="-X"` le prohíbe a `xorg-x11` instalarse como una dependencia (`rp-pppoe` tiene herramientas gráficas; si quiere habilitarlas, puede recompilar `rp-pppoe` más tarde o cuando `xorg-x11` ya esté instalado -- porque toma mucho tiempo en compilarse).

Ahora continúe con [Configurar el Cargado de Arranque](#).

Configurando el gestor de arranque

“Existen varios gestores de arranque para la arquitectura x86. Cada uno tiene su propia configuración. Procederemos paso a paso al configurar el gestor de arranque según sus necesidades.”

1: Realizando su elección

Ahora que su kernel está configurado y compilado y los archivos de configuración necesarios han sido llenados correctamente, ha llegado la hora de instalar el programa que iniciará el kernel cuando arranque el sistema. Este programa es conocido como *gestor de arranque*. Para x86, Gentoo dispone de [GRUB](#) y [LILO](#). Pero antes de instalar uno de los dos gestores de arranque, le informaremos sobre como configurar la memoria de imagen (framebuffer) , siempre que usted quiera, claro. Con la memoria de imagen puede ejecutar la línea de comandos de Linux con algunas características gráficas (limitadas) como, por ejemplo, el empleo de un bonita imagen de Gentoo durante el inicio (bootsplash)

... : Opcional: Memoria de imagen (Framebuffer)

Si ha configurado su kernel con soporte framebuffer (o ha utilizado la configuración predeterminada de [genkernel](#)) puede activarlo añadiendo el parámetro [vga](#) y/o [video](#) al archivo de configuración del gestor de inicio.

Lo primero que necesita es conocer que tipo de dispositivo de framebuffer está utilizando. Si está utilizando un kernel parcheado por Gentoo (como puede ser [gentoo-sources](#)) tiene la posibilidad de seleccionar [vesafb-tng](#) como tipo de controlador para vesa (*VESA driver type*), éste viene predeterminado en las fuentes del kernel. Si este es su caso, está utilizando [vesafb-tng](#) y no necesita el parámetro [vga](#). Si no es así, está utilizando el controlador [vesafb](#) y necesita el parámetro [vga](#).

El parámetro [vga](#) controla la resolución y la profundidad de color de su framebuffer utilizando [vesafb](#). Como se indica en `/usr/src/linux/Documentation/fb/vesafb.txt` (el cual se instala cuando instalas un paquete con las fuentes del kernel) necesita pasar el número VESA

correspondiente con la resolución y profundidad de color que quiera.

La siguiente lista muestra las resoluciones y profundidad de color disponibles y su correspondiente valor para el parámetro `vga`.

	640x480	800x600	1024x768	1280x1024
256	0x301	0x303	0x305	0x307
32k	0x310	0x313	0x316	0x319
64k	0x311	0x314	0x317	0x31A
16M	0x312	0x315	0x318	0x31B

El parámetro `video` controla las opciones de visualización. Necesita que se le indique el controlador de framebuffer (`vesafb` para kernels 2.6, o `vesa` para kernels 2.4) seguido de los parámetro de control que quiera activar. Todas las variables se pueden encontrar en `/usr/src/linux/Documentation/fb/vesafb.txt`, pero aquí le informamos sobre las más utilizadas:

Control	Descripción
<code>ywrap</code>	Asume que su tarjeta gráfica puede volver sobre su memoria. (e.j. continuar al principio cuando se aproxima al final)
<code>mtrr</code>	Configura los registros MTRR
<code>mode</code>	(solamente <code>vesafb-tng</code>) Configura la resolución, la profundidad de color y la tasa de refresco. Por ejemplo, <code>1024x768-32@85</code> para una resolución de 1024x768, profundidad de color 32 bit y una tasa de refresco de 85 Hz.

El resultado de estas dos configuraciones podría ser algo como: `vga=0x318 video=vesafb:mtrr,ywrap` o `video=vesafb:mtrr,ywrap,1024x768-32@85`. Recuerde (o anote) está configuración. La necesitará dentro de poco.

Ahora continúe con la instalación de GRUB o LILO.

2: Por defecto: Usando GRUB

La parte más crítica para la comprensión de GRUB, es el habituarse a la manera en que GRUB se refiere a los discos duros y las particiones. Su partición Linux `/dev/hda1` es denominada `(hd0,0)` por GRUB. Note los paréntesis alrededor de `hd0,0`, que son obligatorios.

La numeración de los discos duros comienza con un cero, en lugar de utilizar una "a" y las particiones empiezan con cero en lugar de con un uno. Advertimos también que entre los dispositivos `hd`, sólo se cuentan los discos duros y no los

dispositivos atapi-ide como cdroms y grabadoras. Además, la especificación es igual para dispositivos scsi. (Usualmente son asignados números mayores, excepto cuando el bios está configurado para arrancar desde los dispositivos scsi).

Asumiendo que tiene un disco duro en `/dev/hda`, un lector de cdrom en `/dev/hdb`, una grabadora en `/dev/hdc`, un segundo disco duro en `/dev/hdd` y ningún disco duro scsi, la partición `/dev/hdd7` que descrita como `(hd1, 6)`. A lo mejor puede sonar raro y bien podría serlo, pero como veremos, GRUB ofrece un mecanismo de terminación por tabulador que es de gran ayuda para los que tienen una gran cantidad de discos duros y que andan algo perdidos con este esquema de numeración.

Habiéndole tomado la medida, es hora de instalar GRUB.

. . . : Instalando GRUB

Para instalar GRUB, primero lo instalamos con emerge:

Code Listing 1: Instalando GRUB

```
# emerge grub
```

Aunque GRUB esté instalado, todavía necesitamos crear un archivo de configuración para él e instalar GRUB en nuestro MBR para que pueda arrancar automáticamente nuestro kernel recién creado. Cree el archivo `/boot/grub/grub.conf` con `nano`, o cualquier otro editor:

Code Listing 2: Creando el archivo `/boot/grub/grub.conf`

```
# nano -w /boot/grub/grub.conf
```

Ahora vamos a escribir el `grub.conf`. A continuación encontrará dos posibles archivos `grub.conf` para el ejemplo de particiones que manejamos en esta guía. Sólo hemos comentado exhaustivamente el primer `grub.conf`. Asegúrese de utilizar el nombre de `su` archivo de imagen del kernel y, si es necesario, el nombre de `su` imagen `initrd`.

- El primer `grub.conf` es para quienes no hayan usado `genkernel` para construir su kernel
- El segundo `grub.conf` es para quienes hayan utilizado `genkernel` para construir su kernel.

Note: Si su sistema de fichero para root es JFS, *deberá* añadir " ro" a la línea del [kernel](#) ya que JFS necesita leer su log antes de permitir montajes de lectura-escritura.

Code Listing 3: grub.conf para quienes no hayan usado genkernel

```
# Cual título arrancar por defecto. 0 es el primero, 1 el segundo, etc.
default 0
# Cuantos segundos esperar antes de arrancar el título por defecto.
timeout 30
# Una bella, hermosa imagen para ensalzar las cosas un poco :)
# Coméntela si no tiene una tarjeta gráfica instalada
splashimage=(hd0,0)/grub/splash.xpm.gz

title=Gentoo Linux 2.6.11-r11
# Partición donde se encuentra la imagen del kernel (o sistema operativo)
root (hd0,0)
kernel /kernel-2.6.11-gentoo-r11 root=/dev/hda3

# Las siguientes cuatro líneas sólo se usan en caso de arranque dual con un
# sistema Windows.
# En este caso, Windows reside en la partición /dev/hda6.
title=Windows XP
rootnoverify (hd0,5)
makeactive
chainloader +1
```

Code Listing 4: grub.conf para usuarios del genkernel

```
default 0
timeout 30
splashimage=(hd0,0)/grub/splash.xpm.gz

title=Gentoo Linux 2.6.11-r11
root (hd0,0)
kernel /kernel-2.6.11-gentoo-r11 root=/dev/ram0 init=/linuxrc ramdisk=8192
real_root=/dev/hda3 udev
initrd /initrd-2.6.11-gentoo-r11

# Solo en caso de arranque dual
title=Windows XP
root (hd0,5)
makeactive
chainloader +1
```

Note: Si usa un esquema de particiones y/o imagen de kernel distinta, haga los ajustes respectivos. Sin embargo, asegúrese que cualquier cosa que siga un dispositivo GRUB (tal como (hd0,0)) sea relativa al punto de montaje y no de la raíz. En otras palabras, (hd0,0)/grub/splash.xpm.gz es en realidad /boot/grub/splash.xpm.gz ya que (hd0,0) es /boot.

Si necesita pasar algún parámetro adicional al kernel, sencillamente agréguelo al final de la línea de comando del kernel. Ya estamos pasando una opción ([root=/dev/hda3](#) o [real_root=/dev/hda3](#)), pero se pueden pasar otras también. Como ejemplo usamos el enunciado [vga](#) para el framebuffer discutido previamente:

Code Listing 5: Agregando el enunciado vga como parámetro del kernel

```
title=Gentoo Linux 2.4.26-r9
root (hd0,0)
kernel /kernel-2.4.26-gentoo-r9 root=/dev/hda3 vga=788
```

Si está utilizando un kernel 2.6.7 o superior y ha puentado su disco duro porque la BIOS no puede manejar discos duros grandes, necesitará añadir [hdx=stroke](#).

Los usuarios de [genkernel](#) deben saber que sus kernels usan las mismas opciones de arranque que el CD de instalación. Por ejemplo, si tiene dispositivos scsi, debería agregar el parámetro [doscsi](#) al kernel.

Ahora grabe el archivo `grub.conf` y salga. Aún necesita instalar GRUB en el MBR (Master Boot Record) para que GRUB se ejecute automáticamente cuando arranque su sistema.

Los desarrolladores de GRUB recomiendan utilizar [grub-install](#). Sin embargo, si por alguna razón [grub-install](#) no funciona correctamente todavía tiene la opción de instalar GRUB manualmente.

Continué con [Por defecto: Configurando GRUB utilizando grub-install](#) o [Alternativa: Configurando GRUB a mano, utilizando instrucciones](#).

. . . : Por defecto: Configurando GRUB utilizando grub-install

Para instalar GRUB necesita ejecutar el comando [grub-install](#). Sin embargo [grub-install](#) no funcionará tal cual se instala, ya que estamos dentro de un entorno chroot. Antes de seguir, necesitará actualizar `/etc/mtab` (el archivo que contiene la información relativa a todos los sistemas de archivos montados): afortunadamente hay una manera sencilla de realizar esto, simplemente copie `/proc/mounts` a `/etc/mtab`:

Code Listing 6: Actualizando /etc/mtab

```
# cp /proc/mounts /etc/mtab
```

Ahora podemos instalar GRUB utilizando [grub-install](#):

Code Listing 7: Ejecutando grub-install

```
# grub-install /dev/hda
```

Si tiene alguna pregunta más acerca de GRUB, por favor consulte el [FAQ de GRUB \(en inglés\)](http://www.gnu.org/software/grub/grub-faq.html) [http://www.gnu.org/software/grub/grub-faq.html] o el [Manual de GRUB \(en inglés\)](http://www.gnu.org/software/grub/manual/) [http://www.gnu.org/software/grub/manual/].

Note: Cuando reinstale un kernel, no necesita sobrescribir los archivos. Simplemente, ejecute `make install` después de compilar el kernel; automáticamente copiará los archivos necesarios y ajustará la configuración de GRUB.

Continué con [Reiniciando el sistema](#).

... : Alternativa: Configurando GRUB a mano, utilizando instrucciones

Para comenzar a configurar GRUB, debe ejecutar los comandos dentro de `grub`. Conocerá al interprete de comandos propio de grub. Ahora necesita ejecutar los comandos necesarios para instalar el registro de arranque de GRUB en su disco duro.

Code Listing 8: Iniciando el intérprete de comandos de GRUB

```
# grub
```

Note: Si su sistema no tiene ningún dispositivo de disquetes, añada la opción `--no-floppy` al comando anterior para evitar que grub pruebe los dispositivos (no existentes) de disquetes.

En el ejemplo de configuración queremos instalar GRUB para que lea la información de la partición de arranque `/dev/hda1`, e instala el registro de arranque de GRUB en el MBR (Master boot Record) de su disco duro para que lo primero que veamos aparecer al encender el ordenador sea GRUB. Por supuesto, si no ha seguido el ejemplo de configuración durante la instalación, cambie los comandos de acuerdo a su modelo:

El mecanismo de completar comandos por tabulación de GRUB puede utilizarse dentro de GRUB. Por ejemplo, si escribe `"root ("` seguido de una tabulación, notará que se le presenta una lista de dispositivos (como pueda ser `hd0`). Si tecleamos `"root (hd0,"` seguido de una tabulación recibiremos una lista de particiones disponibles para elegir (como pueda ser `hd0 , 0`).

Utilizando este mecanismo de completar por tabulación, configurar GRUB no debería resultar tan duro. Ahora vamos a por ello, configuremos GRUB!.

Code Listing 9: Instalando GRUB en el MBR

```
grub> root (hd0,0)           (Especifique donde tiene su partición /boot)
grub> setup (hd0)           (Instalamos GRUB en el MBR)
grub> quit                  (Salimos del intérprete de comandos de GRUB)
```

Note: Si quiere instalar GRUB en una partición concreta en lugar del MBR, debe modificar el comando `setup` para que apunte a la partición correcta. Por ejemplo, si quiere que GRUB se instale en `/dev/hda3`, el comando adecuado sería `setup (hd0,2)`. Sin embargo, pocos usuarios querrán hacer esto.

Si tiene alguna pregunta más acerca de GRUB, por favor consulte el [FAQ de GRUB](http://www.gnu.org/software/grub/grub-faq.html) [http://www.gnu.org/software/grub/grub-faq.html] o el [Manual de GRUB](http://www.gnu.org/software/grub/manual/) [http://www.gnu.org/software/grub/manual/].

Continué con [Reiniciando el sistema](#).

3: Alternativa: Utilizando LILO

LILO, representa LinuxLOader, y es el caballito de batalla probado y comprobado de los gestores de inicio de Linux. Sin embargo, carece de algunas características de GRUB (razón por la cual GRUB actualmente está ganando popularidad). La razón por la cual LILO sigue en uso es que en algunos sistemas, GRUB no funciona mientras que LILO sí. Por supuesto también se usa porque hay muchos que lo conocen y prefieren seguir con este gestor. De cualquier manera, Gentoo soporta ambos gestores y por lo visto, ha elegido usar LILO.

Instalar LILO es fácil, sencillamente use `emerge`.

Code Listing 10: Instalando LILO

```
# emerge lilo
```

. . . : Configurando LILO

Para configurar LILO, debe crear el archivo `/etc/lilo.conf`. Use su editor de textos preferido (en el manual usamos `nano` para ser consistentes) y creemos este archivo.

Code Listing 11: Creando `/etc/lilo.conf`

```
# nano -w /etc/lilo.conf
```

Algunas secciones atrás, le pedimos que se acordara del nombre de la imagen del kernel creado. En el siguiente ejemplo de `lilo.conf` haremos uso del esquema ejemplo de particionamiento. Hay dos partes separadas:

- Para los que no hayan usado `genkernel` para construir su kernel
- Para los que hayan usado un `genkernel` para construir su kernel

Asegúrese de utilizar el nombre de *su* archivo de imagen del kernel y, si es necesario, el nombre de *su* imagen `initrd`.

Note: Si su sistema de ficheros para root es JFS, *deberá* añadir la línea `append="ro"` después de cada elemento de arranque, ya que JFS necesita leer su log antes de permitir montajes de lectura-escritura.

Code Listing 12: Ejemplo de /etc/lilo.conf

```
boot=/dev/hda           # Instalar LILO en el MBR
prompt                 # Darle al usuario una oportunidad de seleccionar otra
opción
timeout=50            # Esperar 5 (cinco) segundos antes de arrancar la opción
por defecto
default=gentoo        # Al transcurrir el plazo de espera, arrancar la opción
"gentoo"

# Para los que no usaron genkernel
image=/boot/kernel-2.6.11-gentoo-r11
label=gentoo          # El nombre de la sección
read-only             # Comience con la raíz solo lectura. ¡No modifique!
root=/dev/hda3        # Ubicación del sistema raíz de archivos

# Para usuarios de genkernel
image=/boot/kernel-2.6.11-gentoo-r11
label=gentoo
read-only
root=/dev/ram0
append="init=/linuxrc ramdisk=8192 real_root=/dev/hda3 udev"
initrd=/boot/initrd-2.6.11i-gentoo-r11

# La siguientes dos línea solo corresponden si hace arranque
dual con un sistema Windows.
# En este caso, Windows se encuentra en /dev/hda6.
other=/dev/hda6
label=windows
```

Note: Si usa un esquema de particionamiento o imagen de kernel distinta, haga los ajustes correspondientes.

Si hace falta pasar alguna opción adicional al kernel, incluya un enunciado `append` a la sección. A modo de ejemplo, agregamos un enunciado `vga=788` para activar framebuffer:

Code Listing 13: Usando append para agregar opciones al kernel

```
image=/boot/kernel-2.6.11-gentoo-r11
label=gentoo
read-only
root=/dev/hda3
append="vga=788"
```

Si está utilizando un kernel 2.6.7 o superior y ha puentado su disco duro porque la BIOS no puede manejar discos duros grandes, necesitará añadir `hdx=stroke`.

Usuarios de `genkernel` deben saber que sus kernel usan las mismas opciones de arranque que el CD de instalación. Por ejemplo, si tiene dispositivos SCSI, debe agregar `doscsi` como opción del kernel.

Ahora, salve el archivo y salga del editor. Para terminar, debe ejecutar el comando `/sbin/lilo` para poder aplicar `/etc/lilo.conf` a su sistema (que se instale en el disco). Acuérdesse de que debe volver a ejecutar `/sbin/lilo` cada vez que instale un nuevo kernel o haga cambios en el menú.

Code Listing 14: Terminando la instalación de LILO

```
# /sbin/lilo
```

Note: Cuando reinstale un kernel, no necesita sobrescribir los archivos. Simplemente, ejecute `make install` después de compilar el kernel; automáticamente copiará los archivos necesarios y ajustará la configuración de LILO.

Ahora puede continuar con [Reiniciando el Sistema](#).

4: Reiniciando el Sistema

Salga del entorno chroot y desmonte todas las particiones que continúen montadas. Después podemos ejecutar el mágico comando que hemos estado esperando: `reboot`.

Code Listing 15: Desmontando todas las particiones y reiniciando

```
# exit
cdimage ~# cd
cdimage ~# umount /mnt/gentoo/boot /mnt/gentoo/proc /mnt/gentoo
cdimage ~# reboot
```

Por supuesto, no olvide quitar el CD arrancable, o el CD será arrancado de nuevo en lugar de su nuevo sistema Gentoo.

Una vez que haya reiniciado su instalación de Gentoo, termínela con [Finalizando su instalación de Gentoo](#).

Finalizando su instalación Gentoo

“Estamos casi listos. Crearemos uno (o más) usuarios para el sistema.”

1: Administración del Usuario

Trabajar como root en un sistema Unix/Linux es *peligroso* y su uso debería evitarse tanto como sea posible. Es por ello que se recomienda *encarecidamente* añadir un usuario para el uso cotidiano del sistema.

Los grupos a los que pertenece el usuario definen que actividades puede realizar. La siguiente tabla muestra una lista de los grupos más importantes que podría querer utilizar.

Grupo	Descripción
audio	para ser capaz de acceder a los dispositivos de audio
cdrom	para poder acceder directamente al cdrom
floppy	para poder acceder directamente a los dispositivos de disquete
games	para poder utilizar los juegos
usb	para poder acceder a los dispositivos USB
video	para acceder al hardware capturador de video y a la aceleración por hardware
wheel	para poder utilizar <code>su</code>

Por ejemplo, para crear un usuario llamado `juan` que pertenezca a los grupos `wheel`, `users` y `audio`, entre en el sistema como root (sólo root puede crear usuarios) y ejecute `useradd`:

Code Listing 1: Añadiendo un usuario de uso cotidiano

```
Login: root
Password: (Su contraseña de root)

# useradd -m -G users,wheel,audio -s /bin/bash juan
# passwd juan
Password: (Introduzca la contraseña para juan)
Re-enter password: (Vuelva a introducir la contraseña para verificar)
```

Si alguna vez este usuario necesita realizar alguna tarea como root, puede utilizar `su` - para obtener temporalmente privilegios de root. Otra forma es utilizar el paquete `sudo` el cual, correctamente configurado, es muy seguro.

2: Opcional: Instalar Paquetes PRG

Important: Esta parte es solo para los usuarios de PRG. El resto deberían saltarse esta parte y continuar con ¿A dónde ir desde aquí?.

Ahora que su sistema está iniciado, entre con el usuario que creó anteriormente (por ejemplo, `juan`) y utilice `su -` para obtener privilegios de root:

Code Listing 2: Obteniendo privilegios de root

```
$ su -  
Password: (Introduzca su contraseña de root)
```

Ahora necesitamos cambiar la configuración de Portage para buscar los binarios pre-compilados del segundo CD (CD de Paquetes de Gentoo). Primero monte éste CD:

Code Listing 3: Monte el CD de Paquetes

```
(Ponga el CD de Paquetes de Gentoo en la bandeja del CD-ROM)  
# mount /mnt/cdrom
```

Ahora configure Portage para usar `/mnt/cdrom` para sus paquetes pre-compilados:

Code Listing 4: Configurando Portage para usar /mnt/cdrom

```
# ls /mnt/cdrom  
  
(Si hay un directorio /mnt/cdrom/packages:)  
# export PKGDIR="/mnt/cdrom/packages"  
  
(Si no:)  
# export PKGDIR="/mnt/cdrom"
```

Ahora instale los paquetes que quiera. El CD de Paquetes contiene varios binarios pre-compilados, como por ejemplo KDE:

Code Listing 5: Instalando KDE

```
# emerge --usepkg kde
```

Asegúrese de instalar los binarios ahora. Cuando haga un `emerge --sync` para

actualizar Portage (como aprenderá más adelante), los binarios pre-compilados quizá no correspondan con los ebuilds de su Portage actualizado. Puede tratar de solventar este problema utilizando `emerge --usepkgonly` en lugar de `emerge --usepkg`.

Enhorabuena, ¡Su sistema está ahora completamente equipado! Continué con [¿A dónde ir desde aquí?](#) para aprender algo más sobre Gentoo.

¿Y ahora qué?

“Ahora que tenemos un sistema Gentoo, ¿qué hacemos después?”

1: Documentación

¡Enhorabuena! Ya tiene funcionando un sistema Gentoo. Pero ¿A donde ir desde aquí? ¿Cuáles son ahora sus opciones? ¿Qué explorar primero? Gentoo ofrece a sus usuarios muchas posibilidades y, por lo tanto, muchas características documentadas (y menos documentadas).

Definitivamente usted debería ojear la siguiente parte del Manual de Gentoo titulada [Trabajando con Gentoo](#) la cual explica cómo mantener su software al día, cómo instalar más software, qué parámetros USE hay, cómo funciona el sistema de inicialización de Gentoo (Gentoo Init system), etc.

Si está interesado en la optimización de su sistema para uso de escritorio, o quiere aprender cómo configurarlo para que sea un completo sistema de escritorio, consulte nuestra extensa [Guía de Configuración del Escritorio](#) [<http://www.gentoo.org/doc/es/index.xml?catid=desktop>].

También disponemos de un [Manual de seguridad en Gentoo](#) [<http://www.gentoo.org/doc/en/gentoo-security.xml>] cuya lectura puede ser muy valiosa.

Para obtener un completo listado de toda nuestra documentación disponible, revise nuestra página de [Recursos de Documentación](#) [<http://www.gentoo.org/doc/es/index.xml>].

2: Gentoo en línea

Por supuesto, usted es siempre bienvenido a nuestros [Foros de Gentoo](#) [<http://forums.gentoo.org>] o a alguno de nuestros [canales de IRC](#) [<http://www.gentoo.org/main/en/irc.xml>].

También tenemos varias [listas de correo](http://www.gentoo.org/main/es/lists.xml) [http://www.gentoo.org/main/es/lists.xml] abiertas para todos nuestros usuarios. La información de cómo entrar en las mismas está disponible en esa página.

Ahora nos callaremos y le dejaremos que disfrute de su instalación :)

3: Cambios en Gentoo desde 2005.1

Gentoo está en continuo movimiento. Las siguientes secciones describen importantes cambios que afectan a la instalación de Gentoo. Solamente se recogen aquellos comunes a la instalación, no los cambios de paquetes que no ocurren durante la instalación.

No se han producido cambios significativos.

Part 2

Trabajando con Gentoo

Aprenda cómo trabajar con Gentoo: instalación de software, configuración de variables, cambiando el comportamiento del sistema Portage, etc.

Contents:

2.1.	Introducción al sistema Portage	92
2.1.1.	Bienvenido a Portage	92
2.1.2.	El Árbol Portage	92
2.1.3.	Mantenimiento de Software	93
2.1.4.	Cuando Portage se queja...	98
2.2.	Los parámetros USE	103
2.2.1.	¿Qué son los parámetros USE?	103
2.2.2.	Usando los parámetros USE	104
2.2.3.	Parámetros USE específicos de un paquete	108
2.3.	Características de Portage	110
2.3.1.	Características del Portage	110
2.3.2.	Compilación Distribuida	111
2.3.3.	Compilación utilizando caché	112
2.3.4.	Soporte para Paquetes Binarios	113
2.4.	Guiones de inicio	115
2.4.1.	Niveles de ejecución	115
2.4.2.	Trabajando con rc-update	119
2.4.3.	Configuración de servicios	120
2.4.4.	Escribiendo scripts de inicio	121
2.4.5.	Cambiando el comportamiento del nivel de ejecución	125
2.5.	Variables de entorno	127
2.5.1.	¿Variables de Entorno?	127
2.5.2.	Definiendo variables globalmente	128
2.5.3.	Definiendo variables locales	131

Introducción al sistema Portage

“Este capítulo explica los pasos “sencillos” que un usuario necesita saber definitivamente para mantener el software en su sistema.”

1: Bienvenido a Portage

Portage es probablemente la más importante innovación de Gentoo en la gestión de software. Con su gran flexibilidad y una gran cantidad de características es frecuentemente apreciado como la mejor herramienta de gestión de software disponible para Linux.

Portage está completamente escrito en [Python](http://www.python.org) y [Bash](http://www.gnu.org/software/bash) y, por tanto, completamente a la vista de los usuarios al ser ambos lenguajes de script.

La mayoría de usuarios trabajarán con Portage a través de la herramienta [emerge](#). Este capítulo no pretende duplicar la información disponible en la página de man sobre emerge. Para una completa información sobre las opciones de emerge, por favor, consulte la página del manual:

Code Listing 1: Leyendo la página del manual sobre emerge

```
$ man emerge
```

2: El Árbol Portage

Cuando hablamos sobre paquetes, nos referimos normalmente a programas software disponible para los usuarios de Gentoo a través del árbol Portage. El árbol Portage es una colección de *ebuilds*, archivos que contienen toda la información que Portage necesita para mantener el software (instalar, buscar, ...). Estos ebuilds residen por defecto en `/usr/portage`.

Cuando le pida a Portage que ejecute alguna acción relacionada con los

programas software, éste utilizará los ebuilds de su sistema como base. Por tanto, es importante que actualice los ebuilds de su sistema para que Portage conozca el nuevo software, actualizaciones de seguridad, etc.

... : Actualizando el árbol Portage

El árbol Portage se actualiza normalmente con [rsync](http://rsync.samba.org) [http://rsync.samba.org], una utilidad rápida de transferencia de archivos incremental. La actualización es muy sencilla, ya que el comando [emerge](#) proporciona una interfaz para rsync:

Code Listing 2: Actualizando el árbol Portage

```
# emerge --sync
```

Si no es capaz de realizar rsync debido a restricciones de cortafuegos puede actualizar su árbol Portage a través de nuestras tres imágenes de Portage generadas diariamente. La herramienta [emerge-webrsync](#) automáticamente comprueba e instala la última en su sistema.

Code Listing 3: Ejecutando emerge-webrsync

```
# emerge-webrsync
```

3: Mantenimiento de Software

Para buscar software utilizando el árbol de Portage , puede utilizar las características de búsquedas propias de [emerge](#). Por defecto, [emerge --search](#) devuelve el nombre de los paquetes cuyo nombre coincide (tanto total como parcialmente) con el término de búsqueda introducido.

Por ejemplo, para buscar todos los paquetes que tengan "pdf" en su nombre:

Code Listing 4: Buscando paquetes cuyo nombre tenga pdf

```
$ emerge --search pdf
```

Si quiere buscar también en las descripciones puede utilizar el parámetro [--searchdesc](#) (o [-S](#)).

Code Listing 5: Buscando paquetes relacionados con pdf

```
$ emerge --searchdesc pdf
```

Cuando eche un vistazo al resultado, notará que le proporciona mucha información. Los campos son etiquetados claramente con lo cual no entraremos en explicar sus significados.

Code Listing 6: Ejemplo de salida de emerge --search

```
* net-print/cups-pdf
  Latest version available: 1.5.2
  Latest version installed: [ Not Installed ]
  Size of downloaded files: 15 kB
  Homepage:      http://cip.physik.uni-wuerzburg.de/~vrbehr/cups-pdf/
  Description:   Provides a virtual printer for CUPS to produce PDF files.
  License:      GPL-2
```

... : Instalando Software

Una vez que haya encontrado el nombre del software que necesite, puede fácilmente instalarlo con **emerge**: simplemente añada el nombre del paquete. Por ejemplo, para instalar **gnumeric**:

Code Listing 7: Instalando gnumeric

```
# emerge gnumeric
```

Muchas aplicaciones depende unas de otras, esto implica que cualquier intento de instalar un cierto paquete de software podría derivar en la instalación de varias dependencias. No se preocupe. Portage maneja también las dependencias. Si quiere conocer que *instalará* Portage cuando le pida que instale un cierto paquete, añada el parámetro **--pretend**. Por ejemplo:

Code Listing 8: Simulando instalar gnumeric

```
# emerge --pretend gnumeric
```

Cuando le pida a Portage que instale un paquete, descargará las fuentes necesarias desde Internet (si fuera necesario) y las guardará por defecto en `/usr/portage/distfiles`. Después de esto, el paquete será descomprimido, compilado e instalado. Si quiere que portage únicamente descargue las fuentes sin instalarlas, añada la opción **--fetchonly** al comando **emerge**:

Code Listing 9: Descargar el código fuente de gnumeric

```
# emerge --fetchonly gnumeric
```

... : Encontrar la documentación de un paquete instalado

Muchos paquetes vienen con su propia documentación. Algunas veces, el parámetro USE `doc` determina si la documentación debe instalarse o no. Puede comprobar la existencia del parámetro USE `doc` con el comando `emerge -vp<nombre paquete>`.

Code Listing 10: Comprobar la existencia de del parámetro USE doc

```
(alsa-lib es tan sólo un ejemplo)
# emerge -vp alsa-lib
[ebuild N    ] media-libs/alsa-lib-1.0.9_rc3  +doc -jack 674 kB
```

Puede activar o desactivar el parámetro USE `doc` a nivel global en el fichero `/etc/make.conf` o para un paquete en concreto en `/etc/portage/package.use`. El capítulo [Parámetros USE](#) cubre todos estos aspectos en detalle.

Una vez que el paquete está instalado, su documentación se encuentra normalmente en una subdirectorio llamado igual que el paquete debajo del directorio `/usr/share/doc`. También puede obtener un listado de todos los archivos instalados con la herramienta `equery` la cual es parte del [paquete](#) [<http://www.gentoo.org/doc/es/gentoolkit.xml>] `app-portage/gentoolkit`.

Code Listing 11: Localizar la documentación de un paquete

```
# ls -l /usr/share/doc/alsa-lib-1.0.9_rc3
total 28
-rw-r--r--  1 root root  669 May 17 21:54 ChangeLog.gz
-rw-r--r--  1 root root 9373 May 17 21:54 COPYING.gz
drwxr-xr-x  2 root root 8560 May 17 21:54 html
-rw-r--r--  1 root root  196 May 17 21:54 TODO.gz
```

(Como alternativa, utilizamos `equery` para localizar los archivos que nos interesan:)

```
# equery files alsa-lib | less
media-libs/alsa-lib-1.0.9_rc3
* Contents of media-libs/alsa-lib-1.0.9_rc3:
/usr
/usr/bin
/usr/bin/alsalisp
(salida truncada)
```

.. : : Desinstalando Software

Cuando quiera desinstalar un paquete software de su sistema, utilice `emerge --unmerge`. Esto le indicará a Portage que desinstale todos los archivos instalados por el paquete en su sistema *excepto* los archivos de configuración de esa aplicación si la había modificado después de la instalación. Esto le permite continuar trabajando con los mismo archivos de configuración si alguna vez decide volver a instalar la aplicación.

Sin embargo, hemos de tener algo muy en cuenta: Portage *no* comprueba si el paquete que está intentando desinstalar es necesario para algún otro. A pesar de esto, le avisará cuando quiera eliminar un paquete importante que pueda romper su sistema si lo desinstala.

Code Listing 12: Desinstalando gnumeric del sistema

```
# emerge --unmerge gnumeric
```

Cuando desinstala un paquete de su sistema, las dependencias de ese paquete que se instalaron automáticamente cuando instaló el software, permanecerán. Para hacer que Portage localice todas las dependencias que puede ser eliminadas actualmente, utilice la utilidad de `emerge depclean`. Hablaremos de esto un poco más adelante.

.. : : Actualizando su Sistema

Para mantener su sistema en perfecto estado (sin mencionar la instalación de las últimas actualizaciones de seguridad) necesita actualizarlo frecuentemente. Partiendo de que Portage solamente comprueba los ebuilds en su árbol Portage, lo primero sería actualizar el propio árbol. Cuando tenga el árbol Portage actualizado, puede actualizar su sistema con `emerge --update world`:

Code Listing 13: Actualizando su sistema

```
# emerge --update world
```

Portage buscará entonces las nuevas versiones de las aplicaciones que tenga instaladas. Sin embargo, solamente comprobará las versiones de las aplicaciones que tenga instaladas, no de las dependencias. Si quiere actualizar cada paquete existente en su sistema, añada la opción `--deep`:

Code Listing 14: Actualizando completamente su sistema

```
# emerge --update --deep world
```

Ya que las actualizaciones de seguridad también afectan a paquetes que no han sido explícitamente instalados en el sistema (pero que son dependencias de otros programas), es recomendable ejecutar este comando de vez en cuando.

Si ha cambiado últimamente alguno de sus parámetros USE quizá quiera añadir también `--newuse`. Portage comprobará si los cambios requieren la instalación de nuevos paquetes o la recompilación de los existentes:

Code Listing 15: Llevando a cabo una actualización completa

```
# emerge --update --deep --newuse world
```

... : Meta-paquetes

Algunos paquetes del árbol Portage no tienen contenido real pero son utilizados para instalar un conjunto de paquetes. Por ejemplo, el paquete `kde` dejará un completo entorno KDE en su sistema a través de instalar varios paquetes relacionados con KDE y sus dependencias.

Si quiere desinstalar dicho paquete de su sistema, ejecutando `emerge --unmerge` sobre el paquete no tendrá efecto total ya que las dependencias permanecerán en su sistema.

Portage tiene la funcionalidad de eliminar las dependencias huérfanas, pero la disponibilidad de software necesita que primero actualice completamente su sistema, incluyendo los nuevos cambios que ha aplicado si actualizó los parámetros USE. Después de esto, puede ejecutar `emerge --depclean` para eliminar las dependencias huérfanas. Cuando haya terminado, necesitará reconstruir las aplicaciones que estuvieran enlazadas dinámicamente a las que acaban de ser eliminadas pero no son necesarias.

Todo esto se lleva a cabo a través de tres comandos:

Code Listing 16: Desinstalando dependencias huérfanas

```
# emerge --update --deep --newuse world
# emerge --depclean
# revdep-rebuild
```

`revdep-rebuild` es parte del paquete `gentoolkit`; no olvide instalarlo primero:

Code Listing 17: Instalando el paquete gentoolkit

```
# emerge gentoolkit
```

4: Cuando Portage se queja...

Como mencionamos anteriormente, Portage es muy potente y soporta muchas características que de las que carecen otras herramientas de gestión de software. Para comprender esto, explicaremos unos cuantos aspectos de Portage sin profundizar demasiado en los detalles.

Con Portage diferentes versiones de un mismo paquete pueden coexistir en un sistema. Mientras otras distribuciones tienden a renombrar el paquete con sus versiones (por ejemplo [freetype](#) and [freetype2](#)). Portage usa una tecnología llamada *SLOTS* (ranuras). Un ebuild declara un cierto SLOT para su versión. Ebuilds con diferentes SLOTS pueden coexistir en el mismo sistema. Por ejemplo, el paquete [freetype](#) tiene ebuilds con `SLOT="1"` y `SLOT="2"`.

También existen paquetes que proporcionan la misma funcionalidad pero están implementados de maneras distintas. Por ejemplo, [metalogd](#), [sysklogd](#) y [syslog-ng](#) son todos logeadores del sistema. Aplicaciones que necesitan la disponibilidad de un "logeador del sistema" no pueden depender, por ejemplo, de [metalogd](#), ya que el resto de logeadores del sistema son igualmente válidos. Portage permite *virtuals*: cada logeador del sistema proporciona [virtual/syslog](#) de tal manera que las aplicaciones puede depender de [virtual/syslog](#).

Los programas en el árbol Portage puede residir en diferentes ramas. Por defecto, su sistema solamente acepta paquetes que Gentoo considera estables. La mayoría de los paquetes nuevos, cuando son aceptados, ingresan en la rama inestable. Esto implica que necesitan hacerse más pruebas antes de marcarlo como estable. Aunque puede ver los ebuilds de ese software en su árbol de Portage, Portage no los actualizará hasta que sean marcados como estables.

Algunos programas sólo están disponibles para unas pocas arquitecturas. O los programas no funcionan en otras arquitecturas, o necesitan más pruebas, o el desarrollador que añade el programa a Portage no es capaz de verificar si el paquete funciona en diferentes arquitecturas.

Cada instalación de Gentoo adhiere un cierto [perfil](#) el cual contiene, entre otra

información, la lista de paquetes necesarios para que el sistema funcione normalmente.

... : Paquetes Bloqueados

Code Listing 18: Portage avisa sobre paquete bloqueados (con --pretend)

```
[blocks B      ] mail-mta/ssmtp (is blocking mail-mta/postfix-2.2.2-r1)
```

Code Listing 19: Portage avisa sobre paquete bloqueados (sin --pretend)

```
!!! Error: the mail-mta/postfix package conflicts with another package.
!!!       both can't be installed on the same system together.
!!!       Please use 'emerge --pretend' to determine blockers.
```

Los Ebuilds contienen campos específicos que informan a Portage sobre sus dependencias. Hay dos posibles dependencias: dependencias de compilación, declaradas en **DEPEND** y dependencias en tiempo de ejecución, declaradas en **RDEPEND**. Cuando una de estas dependencias marca explícitamente un paquete o paquete virtual como *no compatible*, se dispara un bloqueo.

Para solucionar un bloqueo, puede elegir no instalar el paquete o desinstalar primero el paquete conflictivo. En el ejemplo anterior, puedes optar por no instalar **postfix** o eliminar primero **ssmtp**.

También es posible que dos paquetes que aún no se han instalado se estén bloqueando mutuamente. En este caso (poco frecuente), se debería investigar por que necesitamos instalar ambos. En la mayoría de los casos se puede realizar con uno sólo de los paquetes. Si no, por favor envía un informe de error al [sistema de seguimiento de errores de Gentoo](http://bugs.gentoo.org) [http://bugs.gentoo.org].

... : Paquetes enmascarados (masked)

Code Listing 20: Portage avisa sobre paquetes enmascarados

```
!!! all ebuilds that could satisfy "bootsplash" have been masked.
```

Code Listing 21: Portage avisa sobre paquetes enmascarados - razón

```
!!! possible candidates are:  
  
- gnome-base/gnome-2.8.0_pre1 (masked by: ~x86 keyword)  
- lm-sensors/lm-sensors-2.8.7 (masked by: -sparc keyword)  
- sys-libs/glibc-2.3.4.20040808 (masked by: -* keyword)  
- dev-util/cvsd-1.0.2 (masked by: missing keyword)  
- media-video/ati-gatos-4.3.0 (masked by: package.mask)  
- sys-libs/glibc-2.3.2-r11 (masked by: profile)
```

Cuando quiera instalar un paquete que no está disponible para su sistema, recibirá un error de enmascaramiento. Debería probar a instalar una aplicación distinta que este disponible para su sistema o esperar hasta que el paquete este disponible. Siempre hay una razón para que un paquete esté enmascarado.

- **~arch keyword** implica que la aplicación no esta probada lo suficiente para ser parte de la rama estable. Espere unos cuantos días o semanas y vuelva a intentarlo.
- **-arch keyword** o **-* keyword** implica que la aplicación no funciona en su arquitectura. Si cree que el paquete funcionará, cree un bug en nuestro sitio web [bugzilla](http://bugs.gentoo.org) [http://bugs.gentoo.org].
- **missing keyword** implica que la aplicación aún no ha sido probada en su arquitectura. Pida al equipo encargado de portar arquitecturas que pruebe el paquete o pruébelo por ellos y informe de su experiencia en nuestro sitio web [bugzilla](http://bugs.gentoo.org) [http://bugs.gentoo.org].
- **package.mask** implica que el paquete se ha encontrado corrupto, inestable o peor y ha sido marcada deliberadamente para que no se use.
- **profile** implica que el paquete no está disponible para su perfil. La aplicación podría romper su sistema si la instala o no es compatible con el perfil que está usando.

... : Dependencias perdidas**Code Listing 22: Portage avisa sobre dependencias perdidas**

```
emerge: there are no ebuilds to satisfy ">=sys-devel/gcc-3-4.2-r4".  
  
!!! Problem with ebuild sys-devel/gcc-3.4.2-r2  
!!! Possibly a DEPEND/*DEPEND problem.
```

La aplicación que está tratando instalar depende de otro paquete que no esta disponible para su sistema. Por favor, compruebe [bugzilla](http://bugs.gentoo.org) [http://bugs.gentoo.org] para ver si el problema se conoce o no, en este caso informe de ello. A menos que este mezclando ramas esto no debería ocurrir y lo consideraremos un error.

... : Nombre ambiguo del Ebuild

Code Listing 23: Portage avisa sobre nombre ambiguos en ebuild

```
!!! The short ebuild name "aterm" is ambiguous. Please specify
!!! one of the following fully-qualified ebuild names instead:
```

```
dev-libs/aterm
x11-terms/aterm
```

La aplicación que quiere instalar tiene un nombre que corresponde con más de un paquete. Necesita aportar también el nombre de la categoría. Portage le informará de los posibles casos entre los que puede elegir.

... : Dependencias Circulares

Code Listing 24: Portage avisa sobre dependencias circulares

```
!!! Error: circular dependencies:

ebuild / net-print/cups-1.1.15-r2 depends on ebuild /
app-text/ghostscript-7.05.3-r1
ebuild / app-text/ghostscript-7.05.3-r1 depends on ebuild /
net-print/cups-1.1.15-r2
```

Dos (o más) paquetes que quiere instalar dependen uno de otro y, por tanto, no pueden instalarse. Esto casi siempre se considera un error en el árbol Portage. Por favor, vuelva a sincronizar después de un tiempo e inténtelo de nuevo. También puede comprobar [bugzilla](http://bugs.gentoo.org) [http://bugs.gentoo.org] para saber si se tiene conocimiento sobre el tema o si no, en cuyo caso informe sobre ello.

... : Fallo en la descarga

Code Listing 25: Portage avisa sobre un fallo en la descarga

```
!!! Fetch failed for sys-libs/ncurses-5.4-r5, continuing...
(...)
!!! Some fetch errors were encountered. Please see above for details.
```

Portage no es capaz de descargar las fuentes para una aplicación específica y tratará de continuar instalando el resto de aplicaciones (si es posible). Este fallo puede deberse a que un servidor réplica no esta bien sincronizado o a que el ebuild apunta a una localización incorrecta. El servidor donde reside las fuentes podría estar caído por alguna razón.

Pruebe después de una hora y vea si el problema persiste.

. . . : Protección del Perfil de Sistema

Code Listing 26: Portage avisa sobre un paquete protegido por perfil

```
!!! Trying to unmerge package(s) in system profile. 'sys-apps/portage'  
!!! This could be damaging to your system.
```

Está intentando eliminar un paquete que es parte del fundamental de su sistema. Éste se haya en su perfil y es necesario y, por tanto, no debería ser eliminado del sistema.

Los parámetros USE

“Los parámetros USE son una parte importante de Gentoo. En este capítulo aprenderá a trabajar con ellos y comprender cómo estos parámetros USE interactúan con su sistema.”

1: ¿Qué son los parámetros USE?

Mientras esté instalando Gentoo (o cualquier otra distribución, incluso otro sistema operativo), tomará varias decisiones dependiendo del entorno en el que esté trabajando. Una instalación para un servidor es distinta a una para una estación de trabajo. También una estación de trabajo dedicada a juegos es diferente a una estación de trabajo que se use para renderizados en 3D.

Estas diferencias no solo dependen de los paquetes instalados, si no también de las características para las que ciertos paquetes tienen soporte. Si no necesita OpenGL, ¿para qué molestarse en instalar OpenGL y construir la mayoría de sus aplicaciones con soporte OpenGL? Si no quiere usar KDE, ¿para qué molestarte en compilar paquetes con soporte para KDE si podrían funcionar perfectamente sin él?

Para ayudar a los usuarios a decidir qué instalar/activar o no, necesitamos que el usuario especifique su entorno de una manera sencilla. Esto obliga al usuario a decidir que es lo que realmente quiere; además de facilitar a Portage, nuestro sistema de gestión de paquetes, la tarea de tomar decisiones útiles.

. . . : Definición de un parámetro USE

Comencemos por definir qué son los parámetros USE. Un parámetro USE es una palabra clave que incorpora información de soporte y dependencias para un concepto en concreto. Si define un determinado parámetro USE, Portage sabrá que el usuario desea soporte para la palabra clave escogida. Por supuesto, también altera las dependencias de un paquete.

Veamos un ejemplo específico: la palabra clave `kde`. Si no la tiene en su variable `USE`, todos los paquetes que tengan soporte *opcional* para KDE se construirán *sin* él. Los que tengan una dependencia *opcional* con KDE se instalarán *sin* instalar

las librerías de KDE (como dependencia). Si ha definido la palabra clave [kde](#), entonces dichos paquetes *sí* se construirán con soporte para KDE, y las librería de KDE serán instaladas

Definiendo correctamente las palabras clave, conseguirá un sistema confeccionado específicamente para sus necesidades.

... : ¿Qué parámetros USE existen?

Hay dos tipos de parámetros USE: *globales* y *locales*.

- Un parámetro USE *global* lo usan varios paquetes, en todo el sistema. Es lo que la mayoría de la gente entiende como parámetros USE.
- Un parámetro USE *local* lo utiliza un sólo paquete para tomar decisiones específicas para dicho paquete.

Puede encontrar una lista de los parámetros USE globales [en línea](http://www.gentoo.org/dyn/use-index.xml) [\[http://www.gentoo.org/dyn/use-index.xml\]](http://www.gentoo.org/dyn/use-index.xml) o localmente en `/usr/portage/profiles/use.desc`. Un pequeño (*realmente* pequeño) extracto:

Code Listing 1: Un pequeño extracto de los parámetros USE disponibles

```
gtk      - Adds support for x11-libs/gtk+ (The GIMP Toolkit)
gtk2     - Use gtk+-2.0.0 over gtk+-1.2 in cases where a program supports both.
gtkhtml  - Adds support for gnome-extra/gtkhtml
guile    - Adds support for dev-util/guile (interpreter for Scheme)
icc      - Use the Intel C++ Compiler if the package supports it
icc-pgo  - Enable PGO data generation or use when use icc.
imap     - Adds support for IMAP
```

Se puede encontrar una lista de los parámetros USE locales en `/usr/portage/profiles/use.local.desc`.

2: Usando los parámetros USE

Esperamos que se haya convencido de la importancia de los parámetros USE. Ahora pasaremos a explicar como se declaran estos parámetros.

Como ya se ha dicho anteriormente, todos los parámetros del [USE](#) se declaran dentro de la variable [USE](#). Para simplificar al usuario la tarea de buscar y escoger

parámetros USE, ya proporcionamos una configuración **predeterminada**. Esta configuración es un compendio de parámetros que creemos se utilizan frecuentemente por los usuarios de Gentoo. Dicha configuración predeterminada se declara en los ficheros `make.defaults` que forman parte de su perfil.

El perfil al que atiende su sistema lo indica el enlace simbólico `/etc/make.profile`. Cada perfil funciona sobre otro, más extenso, y el resultado final es una suma de todos ellos. El perfil más alto es el perfil base (`/usr/portage/profiles/base`).

Echemos un vistazo a la configuración predeterminada para el perfil 2004.3:

Code Listing 2: Variable USE acumulada de `make.defaults` para el perfil 2004.3

(Esto es un ejemplo de la suma de configuraciones en: `base`,
This example is the sum of the settings in `base`, `default-linux`,
`default-linux/x86` y `default-linux/x86/2004.3`)

```
USE="x86 oss apm arts avi berkdb bitmap-fonts crypt cups encode fortran f77
foomaticdb gdbm gif gpm gtk gtk2 imlib jpeg kde gnome libg++ libwww mad
mikmod motif mpeg ncurses nls oggvorbis opengl pam pdflib png python qt
quicktime readline sdl spell ssl svga tcpd truetype X xml2 xmms xv zlib"
```

Como puede ver, esta variable contiene bastantes palabras clave. **No modifique** el fichero `make.defaults` para ajustar la variable **USE** a sus necesidades: ¡los cambios se perderán al actualizar el árbol del Portage!

Para modificar esta configuración predeterminada, necesita añadir o eliminar palabras clave a la variable **USE**. Para llevarlo a cabo, se define la variable **USE** en `/etc/make.conf`. En esta variable añada los parámetros USE que necesite o elimine los que no quiera. Para eliminarlos coloque el símbolo menos ("-") delante.

Por ejemplo, para eliminar el soporte para KDE y QT además de añadir soporte para ldap, puede definirse el siguiente parámetro **USE** en `/etc/make.conf`:

Code Listing 3: Un ejemplo de confirmación USE en `/etc/make.conf`

```
USE="-kde -qt ldap"
```

.. : Declarar parámetros USE para paquetes específicos

A veces le interesará establecer un cierto parámetro USE tan sólo para una o dos aplicaciones, pero no para todo el sistema. Para solventar esto, necesitará crear el directorio `/etc/portage` (si no existiera) y editar `/etc/portage/package.use`.

Por ejemplo, si no le interesa soporte global para `berkdb` pero lo quiere para `mysql`, necesita añadir:

Code Listing 4: Ejemplo de `/etc/portage/package.use`

```
dev-db/mysql berkdb
```

Por supuesto también puede *desactivar* el empleo específico de un parámetro USE para una aplicación en concreto. Por ejemplo si no quiere soporte para `java` en PHP:

Code Listing 5: Segundo ejemplo de `/etc/portage/package.use`

```
dev-php/php -java
```

...: Declarar parámetros USE temporales

A veces necesitará utilizar una cierta configuración de USE tan sólo una vez. En lugar de editar `/etc/make.conf` dos veces (una para hacer y otra para deshacer los cambios) puede declarar la variable USE como una variable de entorno. Recuerde que, si utiliza este método, cuando vuelva a emerger o actualice esta aplicación (tanto si es particular como si forma parte de una actualización del sistema) perderá los cambios.

Como ejemplo, vamos a eliminar temporalmente el parámetro `java` USE durante la instalación de mozilla.

Code Listing 6: Utilizando USE como una variable de entorno

```
# USE="-java" emerge mozilla
```

...: USE flags automáticas

Después de instalar ciertos paquetes, nuevos parámetros USE adicionales se activarán automáticamente si no especifica explícitamente lo contrario. Para comprobar la lista de paquetes que activan automáticamente algunos parámetros USE, compruebe los archivos `/etc/make.profile/use.defaults` y `use.defaults` de los perfiles padres.

Code Listing 7: Un extracto de `/etc/make.profile/use.defaults`

```
gnome          gnome-base/gnome
gtk            x11-libs/gtk+
qt            x11-libs/qt
kde           kde-base/kdebase
motif         x11-libs/openmotif
```

... : Precedencia

Por supuesto, hay una determinada precedencia respecto a qué configuración tiene prioridad sobre la configuración del USE. No querrá declarar `USE="-java"` y comprobar posteriormente que `java` continua utilizándose debido a una configuración que tiene mayor prioridad. La precedencia para la configuración del USE es (el primero tiene la mínima prioridad):

1. Configuración predeterminada del USE declarada en los archivos `make.defaults` de su perfil de su perfil.
2. Configuración del USE heredada si un paquete de `/etc/make.profile/use.defaults` está instalado
3. Configuración definida por el usuario en `/etc/make.conf`
4. Configuración definida por el usuario en `/etc/portage/package.use`
5. Configuración definida por el usuario como variable de entorno

Para observar el valor final del `USE` tal y como lo verá Portage, ejecute `emerge --info`. Se listarán una serie de variables importantes (incluyendo la variable `USE`) con sus valores correspondientes.

Code Listing 8: Ejecutando `emerge info`

```
# emerge --info
```

... : Adaptando su Sistema Completamente a los Nuevos Parámetros USE

Si ha cambiado sus parámetros USE y desea actualizar todo su sistema para que utilice el nuevo parámetro, utilice la opción de `emerge` llamada `--newuse`:

Code Listing 9: Recompilando todo su sistema

```
# emerge --update --deep --newuse world
```

A continuación, ejecute una limpieza completa de Portage para eliminar las

dependencias que habían sido instaladas en su "antiguo" sistema pero que han quedado obsoletas por los nuevos parámetros de USE.

Warning: Ejecutar `emerge depclean` es una operación peligrosa y debería tratarse con cuidado. Revise en profundidad la lista de paquetes "obsoletos" y asegúrese de que no elimina ningún paquete que necesite. En el siguiente ejemplo hemos añadido `-p` para mostrar la lista de paquetes que serían eliminados pero sin eliminarlos físicamente.

Code Listing 10: Desinstalando los paquetes obsoletos

```
# emerge -p depclean
```

Cuando haya finalizado la limpieza, ejecute `revdep-rebuild` para recompilar las aplicaciones que están enlazadas dinámicamente con los objetos que proporcionaban los paquetes eliminados. `revdep-rebuild` forma parte del paquete `gentoolkit`; no olvide emergerlo primero.

Code Listing 11: Ejecutando revdep-rebuild

```
# revdep-rebuild
```

Cuando todo esto haya terminado, su sistema estará utilizando la nueva configuración de los parámetros USE.

3: Parámetros USE específicos de un paquete

Veamos el ejemplo de `mozilla`: ¿Qué parámetros USE influyen sobre él? Para averiguarlo, usamos `emerge` con las opciones `--pretend` (simula llevar a cabo la acción) y `--verbose` (obtener una salida más detallada):

Code Listing 12: Viendo los parámetros USE usados

```
# emerge --pretend --verbose mozilla
These are the packages that I would merge, in order:

Calculating dependencies ...done!
[ebuild N    ] net-www/mozilla-1.5-r1 +java +crypt -ipv6 -gtk2 +ssl +ldap
+gnome -debug +mozcalendar -mozaccess -mozxmlterm -moznoirc -moznomail
-moznocompose -moznoxft
```

`emerge` no es la única herramienta disponible para esta labor. De hecho, tenemos una herramienta llamada `equery` dedicada a obtener información sobre los paquetes; la cual se encuentra en el paquete `gentoolkit`. En primer lugar, instale `gentoolkit`:

Code Listing 13: Instalando gentoolkit

```
# emerge gentoolkit
```

Ahora ejecute `equery` con el argumento `uses` para ver los parámetros del USE de un paquete en concreto. Por ejemplo, en el caso del paquete `gnumeric`:

Code Listing 14: Utilizando equery para ver los parámetros USE usados

```
# equery uses gnumeric
[ Colour Code : set unset ]
[ Legend      : (U) Col 1 - Current USE flags      ]
[             : (I) Col 2 - Installed With USE flags ]

U I [ Found these USE variables in : app-office/gnumeric-1.2.0 ]
- - libgda : Adds GNU Data Access (CORBA wrapper) support for gnumeric
- - gnomedb : unknown
+ + python : Adds support/bindings for the Python language
+ + bonobo  : Adds support for gnome-base/bonobo (Gnome CORBA interfaces)
```

Características de Portage

“Descubra las características de Portage, como el soporte para la compilación distribuida, ccache y aún más.”

1: Características del Portage

Portage tiene varias características adicionales que hacen de su experiencia con Gentoo algo mucho mejor. Muchas de estas características residen en ciertas herramientas software que mejoran el rendimiento, la estabilidad, la seguridad, ...

Para activar o desactivar ciertas características de Portage necesita editar la variable **FEATURES** del archivo `/etc/make.conf`. Esta variable contiene una lista con las palabras clave de cada característica separadas por un espacio en blanco. En algunos casos necesita además instalar la herramienta que implementa la característica.

No todas las características que soporta Portage están aquí reflejadas. Para una consulta completa por favor revise la página de la ayuda referente a `make.conf`

Code Listing 1: Consultando la página de ayuda sobre `make.conf`

```
$ man make.conf
```

Para conocer que características están siendo utilizadas por defecto, ejecute `emerge --info` y busque la variable **FEATURES** o utilice `grep`:

Code Listing 2: Conociendo que características están configuradas

```
$ emerge --info | grep FEATURES
```

2: Compilación Distribuida

`distcc` es un programa para distribuir un trabajo de compilación a través de muchas, no necesariamente idénticas, máquinas en una red. Los clientes de `distcc` envían toda la información necesaria a los servidores DistCC disponibles (corriendo `distccd`) así pueden compilar trozos de código fuente para el cliente. El resultado final, es un tiempo de compilación más rápido.

Puede encontrar información mas detallada sobre `distcc` (e información de como tenerlo funcionando sobre Gentoo) en nuestra [Documentación Gentoo de Distcc](http://www.gentoo.org/doc/es/distcc.xml) [<http://www.gentoo.org/doc/es/distcc.xml>].

. . . : Instalando distcc

Distcc se distribuye con un monitor gráfico para monitorizar las tareas que su computador está enviando para compilar. Si usa Gnome entonces ponga 'gnome' en su configuración [USE](#). De todas formas, si no usa Gnome pero sigue deseando disponer de un monitor, entonces debería poner 'gtk' en su configuración [USE](#).

Code Listing 3: Instalando distcc

```
# emerge distcc
```

. . . : Activando el soporte en el Portage

Añada `distcc` a la variable `FEATURES` dentro de `/etc/make.conf`. Hecho esto, edite la variable `MAKEOPTS` a sus necesidades. Una pauta conocida para configurarla es poner `-jX` con `X` representando el número de CPUs que ejecutan `distccd` (incluyendo el host local) más uno, pero quizá obtenga mejores resultados con otros números.

Ahora ejecute `distcc-config` y cree una lista de los servidores distcc disponibles. Para un ejemplo simple, supondremos que los servidores DistCC son 192.168.1.102 (el host local), 192.168.1.103 y [192.168.1.104](#) (los dos hosts "remotos"):

Code Listing 4: Configurando distcc para usar los tres servidores DistCC

```
disponibles
```

```
# distcc-config --set-hosts "192.168.1.102 192.168.1.103 192.168.1.104"
```

Por supuesto, no se olvide de ejecutar también el demonio `distccd`:

Code Listing 5: Arrancando el demonio `distcc`

```
# rc-update add distccd default
# /etc/init.d/distccd start
```

3: Compilación utilizando caché

`ccache` es un caché de compilación rápida. Cuando compila un programa, puede cachear resultados intermedios, de forma que, si usted recompilara el mismo programa, el tiempo de compilación se reduciría ampliamente. En las aplicaciones comunes, esto puede significar un aumento de velocidad entre 5 y 10 veces.

Si está interesado en los pros y contras de `ccache`, por favor visite la [página web de ccache](http://ccache.samba.org) [http://ccache.samba.org].

...: Instalando `ccache`

Para instalar `ccache`, ejecute `emerge ccache`:

Code Listing 6: Instalando `ccache`

```
# emerge ccache
```

...: Activando el Soporte en el Portage

Primero, edite el `/etc/make.conf` y añada a la variable `FEATURES` la palabra clave `ccache`. A continuación, añada una nueva variable llamada `CCACHE_SIZE` y dele el valor de "2G":

Code Listing 7: Editando `CCACHE_SIZE` en `/etc/make.conf`

```
CCACHE_SIZE="2G"
```

Para comprobar si `ccache` funciona, pídale a `ccache` que te muestre las

estadísticas:

Code Listing 8: Viendo las estadísticas de ccache

```
# ccache -s
```

... : Utilizando ccache para compilaciones de C sin relación con Portage

Si quiere utilizar ccache para compilaciones que no tengan que ver con Portage, añade `/usr/lib/ccache/bin` al principio de su variable `PATH` (antes de `/usr/bin`). Esto puede llevarse a cabo editando `/etc/env.d/00basic`, ya que es este el primer archivo de entorno que define la variable `PATH`:

Code Listing 9: Editando /etc/env.d/00basic

```
PATH="/usr/lib/ccache/bin:/opt/bin"
```

4: Soporte para Paquetes Binarios

Portage soporta la instalación de paquetes precompilados. A pesar de que Gentoo no proporciona paquetes precompilados por sí mismo (excepto para las imágenes GRP) Portage puede estar funcionando perfectamente con paquetes precompilados.

Para crear un paquete precompilado puede utilizar `quickpkg` si el paquete está instalado en su sistema, o `emerge` con las opciones `--buildpkg` o `--buildpkgonly`.

Si quiere que Portage cree paquetes precompilados de cada paquete individual que instale, añade `buildpkg` a la variable `FEATURES`.

Puede encontrar mayor soporte para la creación de conjuntos de paquetes precompilados con `catalyst`. Para más información sobre `catalyst`, por favor lea [Manual de Referencia de Catalyst](http://www.gentoo.org/proj/en/releng/catalyst/reference.xml) [http://www.gentoo.org/proj/en/releng/catalyst/reference.xml] y la [Guía de Catalyst](http://www.gentoo.org/proj/en/releng/catalyst/catalyst-howto.xml) [http://www.gentoo.org/proj/en/releng/catalyst/catalyst-howto.xml] (en inglés ambos documentos).

. . : : Instalando Paquetes Precompilados

A pesar de que Gentoo no proporciona uno, puede crear un repositorio central donde almacene paquetes precompilados. Si quiere utilizar este repositorio, necesita que Portage lo conozca a través de la variable `PORTAGE_BINHOST` que debe apuntar al repositorio. Por ejemplo, si los paquetes precompilados están en `ftp://buildhost/gentoo`:

Code Listing 10: Configurando `PORTAGE_BINHOST` en `/etc/make.conf`

```
PORTAGE_BINHOST="ftp://buildhost/gentoo"
```

Cuando quiera instalar un paquete precompilado, añada la opción `--getbinpkg` al comando `emerge` junto a la opción `--usepkg`. La primera le indica a `emerge` que descargue el paquete precompilado del servidor definido previamente, mientras que el segundo indica a `emerge` que intente instalar el paquete precompilado antes de buscar el código fuente y compilarlo.

Por ejemplo, para instalar `gnumeric` a través de paquetes precompilados:

Code Listing 11: Instalando el paquete precompilado `gnumeric`

```
# emerge --usepkg --getbinpkg gnumeric
```

Más información sobre las opciones para utilizar paquetes precompilados con `emerge` puede consultarse en la página de la ayuda:

Code Listing 12: Leyendo la página de ayuda sobre `emerge`

```
$ man emerge
```

Guiones de inicio

“Gentoo usa un formato especial para los guiones de inicio, que permite, entre otras cosas, decisiones derivadas de dependencias y guiones virtuales. Este capítulo explica estos aspectos y cómo operar con estos guiones..”

1: Niveles de ejecución

Al iniciar, notará que pasará al frente suyo una gran cantidad de texto. Si pone atención, notará que estos textos son iguales cada vez que reinicie su sistema. La secuencia de todas estas acciones se llama la *secuencia de inicio* y es (más o menos) definido estáticamente.

Primero, su gestor de arranque cargará a la memoria la imagen del kernel que definido en la configuración del gestor de arranque, después de lo cual, se indica a la CPU que debe ejecutar el kernel. Al ser cargado y luego ejecutado inicializa todas las estructuras y tareas específicas del kernel e inicia el proceso [init](#).

Este proceso asegura que todos los sistemas de archivo (definidos en `/etc/fstab`) estén montados y listos para usar. Luego ejecuta varios scripts en `/etc/init.d`, correspondientes a los servicios requeridos para tener un sistema correctamente iniciado.

Finalmente, al concluir la ejecución de los scripts, [init](#) activa los terminales (generalmente solo las consolas virtuales accesibles con [Alt-F1](#), [Alt-F2](#), etc.) fijándoles un proceso especial denominado [agetty](#). Este proceso hará posible que pueda ingresar al sistema a través de uno de estos terminales ejecutando [login](#).

. . . : Scripts de inicio (init scripts)

Ahora bien, [init](#) no solamente ejecuta los scripts contenidos en `/etc/init.d` de manera aleatoria. Aún más, no ejecuta todos los scripts del `/etc/init.d`, solamente los que han sido seleccionados para ejecutar. Los scripts seleccionados para ejecutar se encuentran dentro del directorio `/etc/runlevels`.

Primero, [init](#) ejecuta todos los scripts de `/etc/init.d` cuyos vínculos simbólicos

se encuentran dentro de `/etc/runlevels/boot`. Usualmente los iniciará en orden alfabético, pero algunos scripts tienen información relativa a dependencias, para lo cual otros scripts deben ser iniciados anteriormente.

Cuando todos los scripts referenciados en `/etc/runlevels/boot` sean ejecutados, `init` continua su trabajo con los scripts en `/etc/runlevels/default`. Una vez más, usará el orden alfabético, salvo cuando hay dependencias, en cuyo caso es alterado el orden de inicio para realizar una secuencia válida de arranque.

... :: ¿Cómo funciona Init?

Por supuesto que `init` no decide todo eso por su cuenta. Requiere un archivo de configuración que especifica las acciones a tomar. Este archivo es `/etc/inittab`.

Si recuerda al secuencia de inicio recién explicada, recordará que la primera acción de `init` es montar todos los sistemas de archivo. Esto está definido en la siguiente línea de `/etc/inittab`:

Code Listing 1: La línea de inicialización del sistema en `/etc/inittab`

```
si::sysinit:/sbin/rc sysinit
```

Es línea dice a `init` que debe ejecutar `/sbin/rc sysinit` al iniciar el sistema. Los scripts `/sbin/rc` se encargan de la inicialización, con lo que podríamos decir que `init` no hace mucho, delega la tarea de inicialización del sistema a otro proceso.

En segundo lugar, `init` ejecutó los scripts con vínculos simbólicos en `/etc/runlevels/boot`. Esto se define en la siguiente línea:

Code Listing 2: Inicialización del sistema, continuada

```
rc::bootwait:/sbin/rc boot
```

Una vez más, el script `rc` lleva a cabo las tareas necesarias. Note que la opción de `rc` (*boot*) corresponde al subdirectorio usado bajo `/etc/runlevels`.

Ahora `init` revisa su archivo de configuración para ver que *nivel de ejecución* debe ejecutar. Para decidirlo, lee la siguiente línea de `/etc/inittab`:

Code Listing 3: La línea init por defecto (default)

```
id:3:initdefault:
```

En este caso (para la mayoría de usuarios Gentoo), el identificador del *nivel de ejecución* será el 3. Con esta información `init` revisa qué debe ejecutar para iniciar el *nivel de ejecución 3*:

Code Listing 4: Definiciones de niveles de ejecución

```
l0:0:wait:/sbin/rc shutdown
l1:51:wait:/sbin/rc single
l2:2:wait:/sbin/rc nonetwork
l3:3:wait:/sbin/rc default
l4:4:wait:/sbin/rc default
l5:5:wait:/sbin/rc default
l6:6:wait:/sbin/rc reboot
```

La línea que define el nivel 3, de nuevo usa el script `rc` para iniciar los servicios (ahora con el parámetro por defecto *default*). Note una vez más que el parámetro pasado al script `rc` corresponde al subdirectorio de `/etc/runlevels`.

Al terminar `rc`, `init` decide cuáles consolas virtuales debe activar y qué comandos deben ser ejecutados para cada una:

Code Listing 5: Definición de las consolas virtuales

```
c1:12345:respawn:/sbin/agetty 38400 tty1 linux
c2:12345:respawn:/sbin/agetty 38400 tty2 linux
c3:12345:respawn:/sbin/agetty 38400 tty3 linux
c4:12345:respawn:/sbin/agetty 38400 tty4 linux
c5:12345:respawn:/sbin/agetty 38400 tty5 linux
c6:12345:respawn:/sbin/agetty 38400 tty6 linux
```

... :: ¿Qué es un nivel de ejecución?

Ha visto que `init` utiliza un esquema de numeración para decidir cual *nivel de ejecución* debe activar. Un *nivel de ejecución* es un estado en el cual su sistema está corriendo y contiene scripts (del nivel de ejecución o *initscripts*) que serán ejecutados al ingresar o salir del nivel de ejecución.

En Gentoo, hay siete niveles de ejecución definidos: tres internos y cuatro definidos por el usuario. Los internos se llaman *sysinit*, *shutdown* y *reboot* y hacen exactamente lo que implican sus nombres, inicialización, apagado y reinicio del sistema.

Los niveles de ejecución definidos por el usuario están acompañados de un subdirectorío bajo `/etc/runlevels`: `boot`, `default`, `nonetwork` y `single`. El nivel de ejecución `boot` inicia los servicios necesarios que requieren los demás niveles de ejecución. Los tres niveles de ejecución restantes difieren respecto a los servicios que inician: `default` es para uso diario, `nonetwork` en caso de no requerirse la red y `single` es utilizado en caso de necesitar arreglar el sistema.

. . . : Trabajando con los scripts de inicio

Los scripts iniciados por el proceso `rc` son llamados *scripts de inicio* o *init scripts*. Cada script en `/etc/init.d` puede ser ejecutado con los parámetros `start`, `stop`, `restart`, `pause`, `zap`, `status`, `ineed`, `iuse`, `needsme`, `usesme` o `broken`.

Para iniciar, parar o reiniciar un servicio (y sus respectivas dependencias), deben usarse `start`, `stop` y `restart`:

Code Listing 6: Iniciando postfix

```
# /etc/init.d/postfix start
```

Note: Sólo los servicios que necesiten (*need*) del servicio nombrado serán parados o reiniciados. Los demás servicios, aquellos que usen (*use*) el servicio nombrado, pero que no lo necesiten (*need*) continuarán sin ser tocados.

Si desea parar un servicio, pero no los que dependan de él, puede usar el parámetro para pausarlo `pause`:

Code Listing 7: Parando postfix, manteniendo ejecución de los demás servicios

```
# /etc/init.d/postfix pause
```

Si desea ver el estado de un servicio (iniciado, parado, pausado, ...) puede usar el parámetro `status`:

Code Listing 8: Estado del servicio postfix

```
# /etc/init.d/postfix status
```

Si la respuesta a `status` indica que el servicio está corriendo, pero realmente no es así, puede reajustarlo manualmente con el parámetro `zap`:

Code Listing 9: Reajustando la información del estado del servicio postfix

```
# /etc/init.d/postfix zap
```

Para preguntar por las dependencias que tiene un servicio, puede usar `iuse` o `ineed`. Con `ineed` puede ver cuales servicios son realmente necesarios para el correcto funcionamiento del servicio nombrado. Por otra parte, el parámetro `iuse` muestra los servicios que pueden ser usados por el servicio nombrado, pero que no son requeridos para su correcto funcionamiento.

Code Listing 10: Solicitando una lista de servicios de los cuales depende postfix

```
# /etc/init.d/postfix ineed
```

De igual manera, puede indagar que servicios requieren el servicio nombrado (`needsme`) o cuáles pueden usarlo (`usesme`):

Code Listing 11: Solicitando una lista de todos los servicios que requieren postfix

```
# /etc/init.d/postfix needsme
```

Finalmente, puede indagar cuales dependencias son requeridas y están faltando:

Code Listing 12: Solicitando una lista de dependencias faltantes para postfix

```
# /etc/init.d/postfix broken
```

2: Trabajando con rc-update

El sistema de inicio (init) de Gentoo usa un árbol de dependencias para decidir qué servicios deben iniciarse primero. Como ésta es una tarea tediosa, que no deseamos que nuestros usuarios tengan que hacer manualmente, hemos creado unas herramientas para facilitar la administración de los niveles de ejecución y los scripts de inicio.

Con `rc-update` puede añadir o quitar scripts de inicio a un nivel de ejecución. La herramienta `rc-update` automáticamente usará el script `depscan.sh` para reconstruir el árbol de dependencias.

. . . : Añadiendo y removiendo servicios

Ya hemos agregado scripts de inicio al nivel de ejecución por defecto durante la instalación de Gentoo. En ese instante tal vez no haya tenido una idea clara acerca del uso de un nivel de ejecución "por defecto", aunque ahora sí. El script `rc-update` requiere un segundo parámetro que define la acción a llevar a cabo: `add`, `del` o `show` para agregar, borrar o mostrar.

Para añadir o quitar un script de inicio, use `rc-update` con el parámetro `add` o `del`, seguido por el nombre del script de inicio y el nivel de ejecución, por ejemplo:

Code Listing 13: Quitar postfix del nivel de ejecución por defecto

```
# rc-update del postfix default
```

El comando `rc-update show` mostrará todos los scripts de inicio con los niveles de ejecución donde ejecutarán:

Code Listing 14: Recibiendo información de los scripts de inicio

```
# rc-update show
```

3: Configuración de servicios

Los scripts de inicio pueden ser bastante complejos, por lo cual no es interesante que los usuarios modifiquen directamente el script de inicio, ya que esto puede ser propenso a errores. Sin embargo es importante poder configurar estos servicios, en caso que se quieren dar más opciones al servicio.

Una segunda razón para mantener esta información fuera del script de inicio es para poder actualizar estos scripts sin que los cambios de configuración sean perdidos.

... : El directorio /etc/conf.d

Gentoo provee una manera fácil de configurar estos servicios: cada script de inicio configurable tiene un archivo dispuesto en /etc/conf.d. Por ejemplo, el script de inicio apache2 (llamado /etc/init.d/apache2) tiene un archivo de configuración de nombre /etc/conf.d/apache2, el cual contiene las opciones a pasar al servidor web Apache 2 en el momento de inicio:

Code Listing 15: Variables definidas en /etc/conf.d/apache2

```
APACHE2_OPTS="-D PHP4"
```

Este tipo de archivo de configuración contiene solamente variables (como /etc/make.conf), lo que facilita la configuración de servicios. También nos permite suministrar información adicional acerca de las variables (en forma de comentarios).

4: Escribiendo scripts de inicio

Realmente, no. Escribir un script de inicio usualmente no hace falta, ya que Gentoo provee scripts listos para usar para todos los servicios suministrados. Sin embargo, puede haber instalado un servicio sin usar Portage, en cuyo caso probablemente tenga que crear un script de inicio.

No use el script de inicio suministrado por el servicio si no está explícitamente escrito para Gentoo: los scripts de inicio de Gentoo ¡no son compatibles con los de las demás distribuciones!

... : Disposición

La disposición básica de un script de inicio se muestra a continuación.

Code Listing 16: Disposición básica de un script de inicio

```
#!/sbin/runscript

depend() {
    (Información acerca de las dependencias)
}

start() {
    (Comandos requeridos para iniciar el servicio)
}

stop() {
    (Comandos requeridos para parar el servicio)
}

restart() {
    (Comandos requeridos para reiniciar el servicio)
}
```

Cualquier script de inicio *requiere* la definición de la función `start()`. Todas las demás son opcionales.

... : Dependencias

Hay dos dependencias que puede definir: `use` y `need`. Tal como hemos mencionado anteriormente, la dependencia `need` es más estricta que la dependencia `use`. Siguiendo este esquema, se declaran los servicios que dependen de éste o la dependencia *virtual*.

Una dependencia *virtual* es una suministrada por un servicio, pero no solo por ese servicio. Su script de inicio puede depender de un gestor de registro de sistema, habiendo disponibilidad de varios (metalogd, syslog-ng, sysklogd, ...). Como no se `necesitan` todos (ningún sistema normal tiene todos estos gestores de registro instalados y corriendo) nos aseguramos que todos estos servicios `provean` una dependencia virtual.

Examinemos la información de dependencia del servicio postfix.

Code Listing 17: Información de dependencias de postfix

```
depend() {
    need net
    use logger dns
    provide mta
}
```

Como podemos ver, el servicio postfix:

- requiere la dependencia (virtual) `net` (suministrada por, en este caso, `/etc/init.d/net.eth0`)
- usa la dependencia (virtual) `logger` (suministrada por, en este caso, `/etc/init.d/syslog-ng`)
- usa la dependencia virtual (virtual) `dns` (suministrada por, en este caso, `/etc/init.d/named`)
- provee la dependencia (virtual) `mta` (común a todos los servidores de correo electrónico)

... : Controlando el orden

En algunos casos, tal vez no requiera un servicio determinado, pero desea que inicie *antes* (o *después*) de otro servicio *si* está disponible en el sistema (note la condicionalidad, esto ya no es una dependencia) y en el mismo nivel de ejecución (note la condicionalidad, solo involucra servicios del mismo nivel de ejecución). Puede suministrar esta información usando los parámetros `before` o `after`.

Como ejemplo, podemos ver la disposición del servicio `portmap`:

Code Listing 18: La función `depend()` en el servicio `portmap`

```
depend() {
    need net
    before inetd
    before xinetd
}
```

También puede usar el carácter `c` que engloba "*" para todos los servicios, aunque no es aconsejable.

Code Listing 19: Ejecutando un script de inicio como el primer script del nivel de ejecución

```
depend() {
    before *
}
```

... : Funciones estándar

Junto con la función `depend()`, hará falta definir la función `start()`, que contiene los comandos necesarios para inicializar su servicio. Es aconsejable usar las funciones `ebegin` y `eend` para informarle al usuario acerca de lo que está ocurriendo:

Code Listing 20: Ejemplo de función start()

```
start() {
  ebegin "Starting my_service"
  start-stop-daemon --start --quiet --exec /path/to/my_service
  eend $?
}
```

Si requiere más ejemplos de funciones [start\(\)](#), favor leer directamente las fuentes de los scripts de inicio en su directorio `/etc/init.d`. En lo que respecta al [start-stop-daemon](#), hay un excelente página man disponible en caso de requerir mayor información:

Code Listing 21: Buscando página man para el start-stop-daemon

```
# man start-stop-daemon
```

La sintáxis de los scripts de inicio de Gentoo se basa en Bourne Again Shell (bash), lo cual permite utilizar libremente construcciones compatibles con bash dentro de sus scripts de inicio.

Otras funciones que puede definir son: [stop\(\)](#) y [restart\(\)](#). ¡No es obligatorio definir las! Nuestro sistema de inicio es suficientemente inteligente para llevar a cabo estas funciones solo, si usa el [start-stop-daemon](#).

.. : : Añadiendo opciones personalizadas

Si desea que su script de inicio soporte un mayor número de opciones de las que hemos encontrado hasta ahora, debe agregar la opción a la variable `opts` y crear una función con el mismo nombre de la opción. Por ejemplo, para soportar una opción de nombre [restartdelay](#):

Code Listing 22: Soporte para la opción restartdelay

```
opts="${opts} restartdelay"

restartdelay() {
  stop
  sleep 3    # Espere 3 segundo antes de reiniciar
  start
}
```

... : Variables para la configuración de servicios

No hay que hacer nada para soportar un archivo de configuración en `/etc/conf.d`: si su script de inicio se ejecuta, los siguientes archivos serán automáticamente leídos (`sourced`) y las variables estarán disponibles para usar.

- `/etc/conf.d/<your init script>`
- `/etc/conf.d/basic`
- `/etc/rc.conf`

También, si su script de inicio provee una dependencia virtual (como `net`), el archivo asociado a esa dependencia (el `/etc/conf.d/net`) será leído también.

5: Cambiando el comportamiento del nivel de ejecución

Muchos usuarios de equipos portátiles conocen la situación: en casa necesita iniciar `net.eth0` mientras que puede no querer iniciar `net.eth0` mientras está de viaje (cuando no hay una red disponible). Con Gentoo puede modificar el comportamiento del nivel de ejecución para sus propios propósitos.

Por ejemplo puede crear un segundo nivel de ejecución "default" con el cual puede arrancar y que utiliza otros scripts de inicio que le han sido asignados. Puede seleccionar al arrancar que nivel de ejecución quiere utilizar.

... : Utilizando softlevel

Antes de nada, cree el directorio para su segundo nivel de ejecución "default". Como ejemplo vamos a crear el nivel de ejecución `offline`:

Code Listing 23: Creando el directorio para el nivel de ejecución

```
# mkdir /etc/runlevels/offline
```

Añada los scripts de inicio necesarios para el nuevo nivel de ejecución. Por ejemplo, si quiere una copia exacta de su actual "default" pero sin `net.eth0`:

Code Listing 24: Añadiendo los scripts de inicio necesarios

```
(Copiar todos los servicios desde el nivel de ejecución default al nivel offline)
# cd /etc/runlevels/default
# for service in *; do rc-update add $service offline; done
(Eliminar servicios no deseados en el nivel offline)
# rc-update del net.eth0 offline
(Mostrar los servicios activos en el nivel offline)
# rc-update show offline
(Salida incompleta de ejemplo)
    acpid | offline
    domainname | offline
    local | offline
    net.eth0 |
```

Ahora edite la configuración de su gestor de arranque y añada una nueva entrada para el nivel de ejecución **offline**. Por ejemplo, en `/boot/grub/grub.conf`:

Code Listing 25: Añadiendo una entrada para el nivel de ejecución offline

```
title Gentoo Linux Offline Usage
    root (hd0,0)
    kernel (hd0,0)/kernel-2.4.25 root=/dev/hda3 softlevel=offline
```

Listo, ha terminado de configurarlo. Si arranca su sistema y selecciona la nueva entrada al inicio, el nivel de ejecución **offline** será el utilizado en lugar del **default**.

... : Utilizando bootlevel

Utilizar **bootlevel** es completamente análogo a **softlevel**. La única diferencia es que se define un segundo nivel de ejecución "boot" en lugar de un segundo "default".

Variables de entorno

“Con Gentoo se pueden manejar fácilmente las variables de entorno del sistema. Este capítulo explica cómo hacerlo y describe las variables más comunes.”

1: ¿Variables de Entorno?

Una variable de entorno es un objeto designado para contener información usada por una o más aplicaciones. Algunos usuarios (especialmente aquellos nuevos en Linux) encuentran esto un poco extraño o inmanejable. Sin embargo esto no es cierto: usando variables de entorno hace que cualquiera pueda cambiar una opción de configuración para una o más aplicaciones fácilmente.

. . . : Ejemplos Importantes

La siguiente tabla muestra un listado de variables de entorno usado por un sistema Linux y describe su uso. Los valores de ejemplo se encuentran después de la tabla.

<i>Variable</i>	<i>Descripción</i>
PATH	Esta variable contiene una lista de directorios separados por ":" en la cual el sistema buscará los archivos ejecutables. Al introducir el nombre de un ejecutable (como ls , rc-update o emerge) que no se encuentre en un de los directorios listados, el sistema no lo encontrará, (a menos que se introduzca la ruta completa, por ejemplo: /bin/ls).
ROOTPATH	Esta variable tiene la misma función que PATH , pero únicamente contiene los directorios que el sistema debe revisar cuando el usuario root introduce un comando.
LDPATH	Esta variable contiene una lista de directorios separados por ":" en la cual el enlazador dinámico busca para encontrar una librería.
MANPATH	Esta variable contiene una lista de directorios separados por ":" en la cual el comando man busca las páginas de manual.
INFODIR	Esta variable contiene una lista de directorios separados por ":" en la cual el comando info busca las páginas info.
PAGER	Esta variable contiene la ruta hacia el programa utilizado para mostrar el contenido de los ficheros (como less o more).
EDITOR	Esta variable contiene la ruta hacia el programa utilizado para modificar el contenido de los archivos (como nano o vi).
KDEDIRS	Esta variable contiene una lista de directorios separados por ":" los cuales

	contienen material específico de KDE.
CLASSPATH	Esta variable contiene una lista de directorios separados por ":" los cuales contienen las clases de Java.
CONFIG_PROTECT	Esta variable una lista de directorios separados por <i>espacio</i> los cuales deben ser protegidos por Portage durante las actualizaciones..
CONFIG_PROTECT_MASK	Esta variable una lista de directorios separados por <i>espacio</i> los cuales no deben ser protegidos por Portage durante las actualizaciones.

A continuación puedes encontrar ejemplos de definiciones para todas estas variables:

Code Listing 1: Definiciones de ejemplo

```
PATH="/bin:/usr/bin:/usr/local/bin:/opt/bin:/usr/games/bin"
ROOTPATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin"
LDPATH="/lib:/usr/lib:/usr/local/lib:/usr/lib/gcc-lib/i686-pc-linux-gnu/3.2.3"
MANPATH="/usr/share/man:/usr/local/share/man"
INFODIR="/usr/share/info:/usr/local/share/info"
PAGER="/usr/bin/less"
EDITOR="/usr/bin/vim"
KDEDIRS="/usr"
CLASSPATH="/opt/blackdown-jre-1.4.1/lib/rt.jar:."
CONFIG_PROTECT="/usr/X11R6/lib/X11/xkb /opt/tomcat/conf \
                /usr/kde/3.1/share/config /usr/share/texmf/tex/generic/config/ \
                /usr/share/texmf/tex/latex/config/ /usr/share/config"
CONFIG_PROTECT_MASK="/etc/gconf"
```

2: Definiendo variables globalmente

Para centralizar la definición de estas variables, Gentoo introduce el directorio `/etc/env.d`. Dentro de este directorio se encuentran varios ficheros como por ejemplo `00basic`, `05gcc`, etc. los cuales contienen las variables necesarias para la aplicación de la cual llevan el nombre.

Por ejemplo, al instalar `gcc`, un fichero llamado `05gcc` que contiene la definición de las siguientes variables, fue creado por el ebuild:

Code Listing 2: /etc/env.d/05gcc

```
PATH="/usr/i686-pc-linux-gnu/gcc-bin/3.2"
ROOTPATH="/usr/i686-pc-linux-gnu/gcc-bin/3.2"
MANPATH="/usr/share/gcc-data/i686-pc-linux-gnu/3.2/man"
INFOPATH="/usr/share/gcc-data/i686-pc-linux-gnu/3.2/info"
CC="gcc"
CXX="g++"
LDPATH="/usr/lib/gcc-lib/i686-pc-linux-gnu/3.2.3"
```

Otras distribuciones le piden modificar o añadir definiciones de variables de entorno semejantes en `/etc/profile` o en otros sitios. Por otro lado, Gentoo nos hace (y a Portage) más fácil mantener y manejar las variables de entorno sin tener que prestar atención a los numerosos ficheros que pueden contenerlas.

Por ejemplo, cuando `gcc` es actualizado, también es actualizado el fichero `/etc/env.d/05gcc` sin ser necesaria ninguna interacción por parte del usuario.

Esto no solo beneficia a Portage, sino también al usuario. En ocasiones se podrá pedir establecer cierta variable de entorno para todo el sistema. Como ejemplo, tomamos la variable `http_proxy`. En lugar de perder el tiempo con `/etc/profile`, puedes crear el fichero (`/etc/env.d/99local`) y introducir la(s) definición(es) en él:

Code Listing 3: /etc/env.d/99local

```
http_proxy="proxy.server.com:8080"
```

Usando el mismo fichero para todas las variables, se obtiene una visión rápida de las variables que definidas por uno mismo.

... : El script `env-update`

Varios archivos de `/etc/env.d` definen la variable `PATH`. esto no es un error: cuando ejecute `env-update`, este concatenará las múltiples definiciones antes de actualizar las variables de entorno, haciendo más fácil a los paquetes (o usuarios) añadir sus propias opciones en las variables de entorno sin interferir con los valores ya existentes.

El script `env-update` concatenará los valores alfabéticamente ordenados por el nombre de los ficheros de `/etc/env.d`. Esto es así porque muchos de los ficheros de `/etc/env.d` empiezan por un número.

Code Listing 4: Update order used by env-update

```
    00basic          99kde-env          99local
+-----+-----+-----+
PATH="/bin:/usr/bin:/usr/kde/3.2/bin:/usr/local/bin"
```

La concatenación de variables no siempre funciona, sólo con las siguientes variables: [KDEDIRS](#), [PATH](#), [CLASSPATH](#), [LDPATH](#), [MANPATH](#), [INFODIR](#), [INFOPATH](#), [ROOTPATH](#), [CONFIG_PROTECT](#), [CONFIG_PROTECT_MASK](#), [PRELINK_PATH](#) y [PRELINK_PATH_MASK](#). Para el resto de variables (archivos en orden alfabético en `/etc/env.d`) se utilizará el último valor definido

Cuando ejecute [env-update](#), el script creará todas las variables de entorno y las colocará en `/etc/profile.env` (el cual es usado por `/etc/profile`). Además, también extraerá la información de la variable [LDPATH](#) y la usará para crear `/etc/ld.so.conf`. Después de esto, ejecutará [ldconfig](#) para recrear el archivo usado por el enlazador dinámico: `/etc/ld.so.cache`.

Si quiere observar el efecto de [env-update](#) inmediatamente después de ejecutarlo, ejecute el siguiente comando para actualizar su entorno. Posiblemente, los usuarios que instalaron Gentoo ellos mismos, recordarán estas instrucciones de la instalación:

Code Listing 5: Actualizando el entorno

```
# env-update && source /etc/profile
```

Note: El comando anterior, solamente actualiza las variables en la terminal actual y las *nuevas* consolas. Conociendo esto, si se está trabajando en X11, necesitará ejecutar [source /etc/profile](#) en cada nueva terminal que abra o reiniciar las X para que todas las nuevas terminales cogan las nuevas variables. Si está utilizando un gestor de inicio, conviértase en root y ejecute [/etc/init.d/xdm restart](#). Si no, necesitará salir de la sesión y volver a entrar para las X generen hijos con las nuevas variables.

3: Definiendo variables locales

No siempre queremos definir variables de entorno globales. Por ejemplo, podríamos querer añadir `/home/my_user/bin` y el directorio de trabajo actual (en el cual nos encontramos), a la variable `PATH`, pero no queremos que todos los usuarios de nuestro sistema lo tengan en su `PATH`. Si queremos definir una variable localmente, debemos usar `~/.bashrc` o `~/.bash_profile`:

Code Listing 6: Extendiendo el PATH para uso local en: `~/.bashrc`

(Dos puntos sin incluir después un directorio son tratados como el directorio de trabajo actual)

```
PATH="${PATH}:/home/my_user/bin:"
```

Cuando vuelva a iniciar la sesión, su variable `PATH` será actualizada.

.. : Específicas de sesión

En ocasiones, se requieren definiciones aún más estrictas. Puede querer usar binarios de un directorio temporal que ha creado sin tener que usar la trayectoria completa a los binarios o sin editar `~/.bashrc`. Para estos momentos necesitará esto.

En este caso, puede definir la variable `PATH` en su sesión activa usando el comando `export`. Mientras no cierre la sesión, la variable `PATH` usará los valores temporales.

Code Listing 7: Definiendo una variable específica a la sesión

```
# export PATH="${PATH}:/home/my_user/tmp/usr/bin"
```


Part 3

Trabajando con Portage

"Trabajando con Portage" cubre en profundidad la herramienta de manejo de software de Gentoo, el sistema Portage.

Contents:

3.1.	Archivos y directorios	134
3.1.1.	Ficheros de Portage	134
3.1.2.	Guardando ficheros	136
3.1.3.	Compilando aplicaciones	137
3.1.4.	Características de registro de acciones (log)	138
3.2.	Configuración por medio de variables	139
3.2.1.	Configuración del sistema Portage	139
3.2.2.	Opciones al momento de construcción	139
3.2.3.	Protección de los archivos de configuración	140
3.2.4.	Opciones de descarga	141
3.2.5.	Configuración de Gentoo	142
3.2.6.	Comportamiento de Portage	143
3.3.	Mezcla de ramales de software	144
3.3.1.	Utilizando una sola rama	144
3.3.2.	Mezclando estable con pruebas	145
3.3.3.	Empleo de paquetes enmascarados	146
3.4.	Herramientas adicionales de portage	147
3.4.1.	etc-update	147
3.4.2.	dispatch-conf	149
3.4.3.	quickpkg	149
3.5.	Divergiendo del árbol oficial	151
3.5.1.	Utilizando un subconjunto del árbol Portage	151
3.5.2.	Añadiendo Ebuilds no oficiales	151
3.5.3.	Software no mantenido por Portage	152
3.6.	La aplicación de Ebuilds	153
3.6.1.	Emerge y Ebuild	153
3.6.2.	Instalación manual de software	153
3.6.3.	Características adicionales de ebuild	156
3.6.4.	Más información	157

Archivos y directorios

“Ya que desea conocer Portage en profundidad, le interesará saber donde almacena sus archivos y datos.”

1: Ficheros de Portage

Portage viene con una configuración predefinida guardada en `/etc/make.globals`. Cuando le eche un vistazo, comprobará que toda la configuración de Portage se realiza a través de variables. A qué variables atiende Portage y que significan se describe un poco después.

Como muchas directivas de configuración varían de unas arquitecturas a otras, Portage también posee algunos archivos de configuración que son parte de perfil. Su perfil está apuntado por el enlace simbólico `/etc/make.profile`; las configuraciones de Portage se realizan en los archivos `make.defaults` de su perfil y de todos los perfiles padres. Explicaremos algo más sobre perfiles y el directorio `/etc/make.profile` más adelante.

Si está pensando en cambiar una variable de configuración, *no* modifique `/etc/make.globals` o `make.defaults`. En lugar de eso utilice `/etc/make.conf` el cual tiene preferencia sobre los archivos anteriores. También encontrará un archivo `/etc/make.conf.example`. Este archivo es meramente un ejemplo y Portage no lo tendrá en cuenta.

También puede definir una variable de configuración para Portage como una variable de entorno, pero no es recomendable.

... : Información específica del perfil

Ya hemos hablado del directorio `/etc/make.profile`. Bien, exactamente no es un directorio pero es un enlace simbólico a un perfil, por defecto uno perteneciente a `/usr/portage/profiles` también puede crear un perfil en cualquier otro lado y apuntarlo. El perfil al cual apunta el enlace simbólico será el que tenga en cuenta su sistema.

Un perfil contiene información específica para Portage sobre cada arquitectura, tal

como una lista de paquetes que pertenecen al sistema correspondiente con ese perfil, una lista de paquetes que no funcionan (o están enmascarados) para ese perfil, etc.

.. : : Configuración específica para usuarios

Cuando necesite sobrescribir una característica de Portage relativa a la instalación de software, necesitará editar los archivos contenidos en `/etc/portage`. Se *recomienda encarecidamente* que utilice los archivos pertenecientes a `/etc/portage` y está *desaconsejada* la sobrescritura de estas características con variables de entorno.!

Dentro de `/etc/portage` puede crear los siguientes archivos:

- `package.mask` el cual especifica los paquetes que nunca quiere que Portage instale en su sistema.
- `package.unmask` especifica los paquetes que quiere instalar a pesar de haber sido desaconsejados por los desarrolladores.
- `package.keywords` especifica los paquetes que quiere instalar a pesar de no haber sido considerados adecuados para su sistema o arquitectura (todavía).
- `package.use` especifica la lista de variables USE que quiere utilizar para unos determinados paquetes sin tener que configurar el sistema por completo para que use esas variables USE.

Más información sobre el directorio `/etc/portage` y una lista completa de posibles archivos que puede crear se pueden encontrar en la página del man:

Code Listing 1: Leyendo la página del manual sobre Portage

```
$ man portage
```

.. : : Cambiando el fichero de Portage y el lugar del directorio

Los archivos de configuración mencionados anteriormente no pueden ser guardados en ningún otro sitio, Portage siempre los buscará en esos lugares exactos. Sin embargo, Portage utiliza otros muchos lugares para varios propósitos: el directorio de compilación, el lugar donde guardar el código fuente, la localización del árbol de Portage, ...

Todos estos propósitos tienen unas direcciones predeterminadas muy claras pero puede cambiarlas por las que más le gusten indicándolo en `/etc/make.conf`. El

resto de este capítulo explica los lugares destinados a un propósito especial que utiliza Portage y como puede ser modificado su emplazamiento en el sistema de ficheros.

Este documento no pretende ser utilizado como referencia. Si necesita una cobertura 100%, por favor consulte las páginas del man relativas a Portage y `make.conf`:

Code Listing 2: Leyendo las páginas del manual sobre Portage y `make.conf`

```
$ man portage
$ man make.conf
```

2: Guardando ficheros

La ubicación predeterminada del árbol de Portage es `/usr/portage`. Esta definida por la variable `PORTDIR`. Cuando guarde el árbol de Portage en cualquier otro lugar (modificando esta variable), no olvide cambiar el enlace simbólico `/etc/make.profile` de acuerdo con su cambio.

Si modifica la variable `PORTDIR`, seguramente quiera cambiar las siguientes variables ya que no tienen constancia del cambio de `PORTDIR`. Esto es debido a cómo Portage maneja las variables: `PKGDIR`, `DISTDIR`, `RPMDIR`.

.. : : Binarios Pre-compilados

Aunque Portage no utilice binarios pre-compilados por defecto, tiene un buen soporte para ellos. Cuando a Portage se le indica que trabaje con paquetes pre-compilados, los buscará en `/usr/portage/packages`. Esta ubicación está definida por la variable `PKGDIR`.

.. : : Código Fuente

El código fuente de las aplicaciones se guarda por defecto en `/usr/portage/distfiles`. Esta ubicación viene definida por la variable `DISTDIR`.

... : Archivos RPM

Aunque Portage no puede utilizar archivos RPM, es capaz de generarlos utilizando el comando `ebuild` (vea [La aplicación Ebuild](#)). La ubicación predeterminada para estos archivos RPM es `/usr/portage/rpm` y está definida por la variable `RPMDIR`.

... : Base de datos de Portage

Portage guarda el estado del sistema (que paquetes están instalados, what archivos pertenecen a cada paquete, ...) en `/var/db/pkg`. ¡No se deben modificar estos archivos manualmente! Podría romper el conocimiento que tiene Portage sobre el sistema.

... : Caché de Portage

La caché de Portage (con modificaciones temporales, paquetes virtuales, árbol de dependencias, ...) se guarda en `/var/cache/edb`. Esta ubicación es una verdadera caché: se puede limpiar si no se está ejecutando ninguna aplicación que tenga relación con Portage en este momento.

3: Compilando aplicaciones

Los ficheros temporales de portage se guardan por defecto en `/var/tmp`. Esta ubicación se define en la variable `PORTAGE_TMPDIR`.

Si modifica la variable `PORTAGE_TMPDIR`, necesitará cambiar las siguientes variables ya que no tendrán constancia del cambio. Esto es debido a cómo Portage maneja la variable: `BUILD_PREFIX`.

... : Directorio de compilación

Portage crea directorios de compilación específicos para cada paquete que se emerge dentro de `/var/tmp/portage`. Esta ubicación viene definida por la variable `BUILD_PREFIX`.

. . : : Ubicación del sistema de ficheros

Por defecto, Portage instala todos los archivos en el sistema de ficheros activo (/), pero puede cambiarse esta configuración a través de la variable de entorno ROOT. Esto es útil cuando quiera crear nuevas imágenes compiladas.

4: Características de registro de acciones (log)

Portage puede crear un registro por ebuild, pero solamente cuando la variable PORT_LOGDIR esté configurada y apuntando a una dirección con permisos de escritura para Portage (usuario Portage). De manera predeterminada esta variable está desactivada.

Configuración por medio de variables

“Portage es completamente configurable por medio de diversas variables en el archivo de configuración o a través de variables de entorno.”

1: Configuración del sistema Portage

Como hemos acotado previamente, Portage es configurable a través de múltiples variables de entorno que se deben definir en `/etc/make.conf`. Por favor, refiérase a la página man de `make.conf` para más información.

Code Listing 1: Lectura de la página man de `make.conf`

```
$ man make.conf
```

2: Opciones al momento de construcción

Cuando Portage construye las aplicaciones, pasa el contenido de las siguientes variables al guión de compilación y configuración:

- `CFLAGS` & `CXXFLAGS` define los parámetros deseados para la compilación de fuentes en C y C++.
- `CHOST` define la plataforma correspondiente a la máquina en la que se construye para el guión de configuración
- `MAKEOPTS` se pasa al comando `make` para definir el grado de paralelismo al compilar. Para más información acerca de sus opciones, vea la página man de `make`.

El parámetro `USE` también se usa al configurar y compilar, pero éste ha sido explicado ampliamente en capítulos previos.

... : Opciones al integrar

Cundo Portage integra una versión más nueva de algún paquete de software, también eliminará los archivos obsoletos de la versión anterior del sistema. Portage otorga un tiempo de gracia de 5 segundos al usuario antes de llevar esta tarea a cabo. Este tiempo se define por medio de la variable `CLEAN_DELAY`.

3: Protección de los archivos de configuración

Portage sobrescribe los archivos provistos por versiones más nuevas de un paquete si estos no están almacenados en un lugar *protegido*. Estos lugares protegidos se definen con la variable `CONFIG_PROTECT` y generalmente corresponden a rutas de archivos de configuración. Este listado de directorios es delimitado con espacios en blanco.

Los archivos de configuración nuevos que se escriban en rutas protegidas lo serán con un nombre modificado y el usuario será advertido acerca de su presencia.

Puede averiguar qué lugares están protegidos en la variable `CONFIG_PROTECT` con la salida del comando `emerge --info`:

Code Listing 2: Obteniendo información acerca del contenido de `CONFIG_PROTECT`

```
$ emerge --info | grep 'CONFIG_PROTECT='
```

Más información acerca de la protección de archivos de configuración por Portage está disponible a través del comando `emerge`:

Code Listing 3: Más información acerca de la protección de archivos de configuración

```
$ emerge --help config
```

... : Exclusión de directorios

Para 'desproteger' ciertos subdirectorios en directorios protegidos, use la variable `CONFIG_PROTECT_MASK`.

4: Opciones de descarga

Cuando la información o datos no están disponibles en su sistema, Portage los descargará de la Internet. Las ubicaciones de los servidores para los canales de información y datos se definen mediante los siguientes variables:

- `GENTOO_MIRRORS` define una lista de servidores que contienen código fuente (distfiles)
- `PORTAGE_BINHOST` define un servidor en particular que contiene paquetes pre-compilados para su sistema

Un tercer parámetro involucra la ubicación del servidor `rsync` utilizado al actualizar el árbol Portage:

- `SYNC` define un servidor en particular a ser utilizado por Portage para descargar el árbol

Las variables `GENTOO_MIRRORS` y `SYNC` pueden ser configurados automáticamente a través de la aplicación [mirrorselect](#). Debe hacer [emerge mirrorselect](#) primero, antes de usarla. Para más información, vea la ayuda de `mirrorselect` en línea:

Code Listing 4: Más información acerca de mirrorselect

```
# mirrorselect --help
```

Si su entorno requiere el uso de un servidor proxy, configure las variables `HTTP_PROXY`, `FTP_PROXY` y `RSYNC_PROXY` para declararlos.

. . . : Comandos para descargar

Cuando Portage requiera descargar fuentes, utiliza por defecto el comando [wget](#). Puede cambiar esto usando la variable `FETCHCOMMAND`.

Portage puede continuar una descarga hecha en forma parcial. Usa [wget](#) por defecto, pero puede cambiarlo usando la variable `RESUMECOMMAND`.

Asegúrese que sus `FETCHCOMMAND` y `RESUMECOMMAND` guarde las fuentes en la ubicación correcta. Al definir las variables debe usar `\${URI}` y `\${DISTDIR}`

para apuntar a la ubicación de las fuentes y la ubicación del directorio distfiles respectivamente.

Puede definir manejadores específicos por protocolo con `FETCHCOMMAND_HTTP`, `FETCHCOMMAND_FTP`, `RESUMECOMMAND_HTTP`, `RESUMECOMMAND_FTP`, etc.

... : Configuración de rsync

Aunque no se puede alterar el comando rsync usado para actualizar el árbol Portage, podrá configurar algunas de las variables para modificar su comportamiento:

- `RSYNC_EXCLUDEFROM` apunta a un archivo que lista los paquetes y/o categorías que rsync debe ignorar durante la actualización
- `RSYNC_RETRIES` define cuántas veces rsync debe intentar la conexión al servidor espejo apuntado por la variable `SYNC` antes de rendirse. El valor por defecto es 3.
- `RSYNC_TIMEOUT` define el tiempo en segundos que una conexión permanece activa antes de cortar. El valor por defecto es 180, pero usuarios en conexiones por discado tal vez quieran aumentarlo a 300 o más.

5: Configuración de Gentoo

Puede escoger su rama por defecto a través de la variable `ACCEPT_KEYWORDS`. El valor por defecto es la rama estable de su plataforma. Para más información acerca de las ramas de Gentoo, vea el capítulo siguiente.

... : Características de Portage

Puede activar ciertas características de Portage por medio de la variable `FEATURES`. Estas han sido discutidas en capítulos previos, por ejemplo [Características de Portage](#).

6: Comportamiento de Portage

Con la variable `PORTAGE_NICENESS`, puede aumentar o reducir el valor "nice" con el que ejecuta Portage. El valor de la variable `PORTAGE_NICENESS` se *suma* al valor "nice" actual.

Para más información acerca de valores "nice", vea la página `man nice`:

Code Listing 5: Más información acerca de nice

```
$ man nice
```

... : Comportamiento de la salida

El valor de `NOCOLOR`, que por defecto es "falso", define si Portage desactiva el uso de los colores en su salida.

Mezcla de ramales de software

“Gentoo suministra software en varias ramas, dependiendo de la estabilidad y soporte de cada arquitectura. “Mezcla de Ramales de Software” le informa cómo configurar estas ramas y cómo redefinir esta separación de manera individual para cada paquete.”

1: Utilizando una sola rama

La variable `ACCEPT_KEYWORDS` define que rama de programas va a utilizar en su sistema. Como predeterminada figura la rama estable para su arquitectura, por ejemplo `x86`.

Recomendamos que solamente utilice la rama estable. Sin embargo si no le importa demasiado la estabilidad y quiere ayudar a Gentoo a través del envío de informes de error a <http://bugs.gentoo.org>, siga leyendo.

. . . : La rama de pruebas

Si quiere utilizar los programas más recientes, puede considerar utilizar la rama de pruebas. Para que Portage utilice la rama de pruebas, añada un `~` delante de su arquitectura.

La rama de pruebas es exactamente para eso - *pruebas*. Si un paquete se encuentra en pruebas, eso significa que los desarrolladores creen que funciona, pero no ha sido probado concienzudamente. Podría, perfectamente, ser el primero en descubrir un error en el paquete, en cuyo caso puede rellenar un [informe](http://bugs.gentoo.org) [http://bugs.gentoo.org] para ponerlo en conocimiento de los desarrolladores.

Por ejemplo, para seleccionar la rama de pruebas en una arquitectura `x86`, edite `/etc/make.conf` y escriba:

Code Listing 1: Configurando la variables `ACCEPT_KEYWORDS`

```
ACCEPT_KEYWORDS="~x86"
```

Si actualiza su sistema ahora, encontrará que *muchos* paquetes serán

actualizados. Tenga cuidado ya que: cuando haya actualizado su sistema para emplear la rama inestable, normalmente no hay una manera sencilla de volver a la rama estable (excepto mediante el empleo de copias de seguridad, claro).

2: Mezclando estable con pruebas

Puede pedirle a Portage que le permita utilizar la rama de pruebas para algunos paquetes en concreto pero seguir utilizando la rama estable en el resto del sistema. Para realizar esto, añada la categoría del paquete y el nombre si quiere utilizar la rama de pruebas al fichero `/etc/portage/package.keywords`. Por ejemplo, para utilizar la rama de pruebas con [gnumeric](#):

Code Listing 2: Configurando `/etc/portage/package.keywords` para `gnumeric`, línea completa

```
app-office/gnumeric ~x86
```

.. : : Probando versiones concretas

Si quiere utilizar una versión específica de algún paquete desde la rama de pruebas pero no quiere que portage utilice esa rama de pruebas para las siguientes versiones, puede añadir la versión en el fichero `package.keywords`. En este caso se *debe* utilizar el operador `=`. También puede introducir un rango de versiones utilizando los operadores `<=`, `<`, `>` or `>=`.

En cualquier caso, si añade información sobre una versión, *debe* utilizar un operador. Si lo deja sin información sobre la versión, no *puede* emplear un operador.

En el siguiente ejemplo indicamos a Portage que acepte `gnumeric-1.2.13`:

Code Listing 3: Utilizando una versión concreta de `gnumeric`

```
=app-office/gnumeric-1.2.13 ~x86
```

3: Empleo de paquetes enmascarados

Los desarrolladores de Gentoo **no** darán soporte al empleo de estos archivos. Por favor, tenga cuidado cuando haga esto. Las peticiones de soporte relacionadas con [package.unmask](#) y/o [package.mask](#) no serán respondidas. Aquí queda el aviso.

Cuando un paquete ha sido enmascarado por los desarrolladores de Gentoo y aún así desea utilizarlo a pesar de la razón que se menciona en el fichero `package.mask` (situado por defecto en `/usr/portage/profiles`), añada *exactamente* la misma línea en `/etc/portage/package.unmask`.

Por ejemplo, si [=net-mail/hotwayd-0.8](#) está enmascarado, puede desenmascararlo añadiendo exactamente la misma línea en el fichero `package.unmask`:

Code Listing 4: `/etc/portage/package.unmask`

```
=net-mail/hotwayd-0.8
```

. . . : El fichero `package.mask`

Cuando no quiera que Portage instale un paquete en concreto o una versión específica de un paquete en su sistema, puede enmascararlo simplemente añadiendo la línea apropiada a `/etc/portage/package.mask`.

Por ejemplo, si no quiere que Portage instale otras fuentes del kernel que no sean [gentoo-sources-2.6.8.1](#), puede añadir la siguiente línea a `package.mask`:

Code Listing 5: ejemplo de `/etc/portage/package.mask`

```
>sys-kernel/gentoo-sources-2.6.8.1
```

Herramientas adicionales de portage

“El sistema Portage viene con algunas herramientas adicionales que puede hacer que su experiencia con Gentoo sea aún mejor. Siga leyendo para averiguar cómo se usa `dispatch-conf` y otras herramientas.”

1: etc-update

`etc-update` es una herramienta que ayuda a combinar los ficheros `._cfg0000_<name>`. Proporciona una fusión interactiva y además puede auto-combinar cambios triviales. Portage genera los `._cfg0000_<name>` cuando quiere sobrescribir un archivo en un directorio protegido por la variable `CONFIG_PROTECT`.

Ejecutar `etc-update` es bastante simple:

Code Listing 1: Ejecutar etc-update

```
# etc-update
```

Después de combinar los cambios sencillos, se presentará una lista con los ficheros protegidos que tienen una actualización pendiente. Al final se muestran las opciones posibles:

Code Listing 2: etc-update options

```
Por favor, seleccione el fichero a editar introduciendo el número correspondiente.  
    (-1 para salir) (-3 para auto-combinar todos los ficheros restantes)  
    (-5 para auto-combinar SIN usar 'mv -i'):
```

Si se indica `-1`, `etc-update` terminará y no continuará con el resto. Si se introduce `-3` o `-5`, *todos* los ficheros de configuración listados serán sobrescritos con las nuevas versiones. Por tanto es muy importante seleccionar primero los ficheros de configuración que no deben ser automáticamente actualizados. Esto se consigue simplemente indicando el número que aparece a la izquierda del fichero de configuración.

Como ejemplo, seleccionamos el fichero de configuración `/etc/pear.conf`:

Code Listing 3: Actualizando un fichero de configuración concreto

```
Comienzo de diferencias entre /etc/pear.conf y /etc/._cfg0000_pear.conf  
[...]  
Fin de diferencias entre /etc/pear.conf y /etc/._cfg0000_pear.conf  
1) Reemplazar el original con la actualización  
2) Borrar la actualización, manteniendo el original inalterado  
3) Combinar interactivamente el original y la actualización  
4) Mostrar de nuevo las diferencias
```

Ahora puede ver las diferencias entre los dos ficheros. Si cree que el fichero de configuración actualizado puede ser utilizado sin problemas, indique [1](#). Si cree que el fichero de configuración actualizado no es necesario, o no proporciona ninguna información nueva o útil, indique [2](#). Si quiere actualizar su fichero de configuración actual de forma interactiva, introduzca [3](#).

Por ahora, no tiene sentido profundizar más sobre la actualización interactiva. Para completarlo, listaremos los comandos que están disponibles durante la combinación interactiva de ambos ficheros. Son mostradas dos líneas (la original, y la nueva propuesta) y un punto indicativo en el cual puede introducir uno de los comandos siguientes:

Code Listing 4: Comandos disponibles en la combinación interactiva

```
ed:    Editar usando ambas versiones, cada una decorada con una cabecera.  
eb:    Editar usando ambas versiones.  
el:    Editar usando la versión de la izquierda.  
er:    Editar usando la versión de la derecha.  
e:     Editar una nueva versión.  
l:     Usar la versión de la izquierda.  
r:     Usar la versión de la derecha.  
s:     Incluir las líneas comunes sin comentarios.  
v:     Incluir las líneas comunes con comentarios.  
q:     Salir.
```

Cuando haya acabado de actualizar los ficheros de configuración importantes, puede actualizar automáticamente el resto. [etc-update](#) acabará si no encuentra más ficheros de configuración para actualizar.

2: dispatch-conf

Usando [dispatch-conf](#) se pueden combinar las actualizaciones de sus ficheros de configuración al tiempo que se mantiene un registro de todos los cambios. [dispatch-conf](#) almacena las diferencias entre los ficheros de configuración como parches o usando el sistema de revisión RCS.

Al igual que [etc-update](#), se puede mantener los ficheros de configuración tal como están, usar el nuevo fichero de configuración, editar el actual o combinar los cambios interactivamente. Sin embargo, [dispatch-conf](#) tiene también algunas características adicionales interesantes:

- Combina automáticamente ficheros de configuración que sólo contienen actualizaciones de comentarios.
- Combina automáticamente ficheros de configuración que sólo difieren en la cantidad de espacio en blanco

Asegúrese de editar `/etc/dispatch-conf.conf` en primer lugar y de crear el directorio referenciado por la variable `archive-dir`.

Para más información, lea la página de man acerca de [dispatch-conf](#):

Code Listing 5: Leyendo la página man de dispatch-conf

```
$ man dispatch-conf
```

3: quickpkg

Con [quickpkg](#) se pueden crear archivos de paquetes que ya han sido instalados en el sistema. Estos archivos pueden usarse como paquetes precompilados. Ejecutar [quickpkg](#) es sencillo: basta añadir los nombres de los paquetes que se quiere archivar.

Por ejemplo, para archivar [curl](#), [arts](#) y [procps](#):

Code Listing 6: Ejemplo de uso de quickpkg

```
# quickpkg curl arts procps
```

Los paquetes precompilados se almacenarán en `$PKGDIR/All` (por defecto `/usr/portage/packages/All`). Los enlaces simbólicos que apunten a esos paquetes se ubican en `$PKGDIR/<category>`.

Divergiendo del árbol oficial

“Divergiendo del árbol oficial” le dará algunos consejos y trucos para usar su propio árbol Portage, cómo sincronizar solamente las categorías que desee, cómo inyectar paquetes y mucho más.”

1: Utilizando un subconjunto del árbol Portage

Puede realizar una actualización selectiva de ciertas categorías/paquetes e ignorar el resto. Esto se realiza a través indicando a `rsync` que excluya categorías/paquetes durante el proceso `emerge --sync`.

Por defecto, `rsync` comprueba el contenido de `/etc/portage/rsync_excludes` (si existe) el cual contiene las categorías o paquetes que *no* quiere que `rsync` actualice.

Recuerde que esto puede provocar ciertos problemas con las dependencias, ya que paquetes nuevos y aceptados en su sistema pueden depender de otros excluidos.

2: Añadiendo Ebuilds no oficiales

Puede indicarle a Portage que utilice ebuilds que no están disponibles oficialmente a través del árbol de Portage. Cree un nuevo directorio (por ejemplo `/usr/local/portage`) en el cual guardará los ebuilds procedentes de otras fuentes. Utilice la misma estructura de directorios que tenemos en Portage.

Después defina la variable `PORTDIR_OVERLAY` en `/etc/make.conf` y haga que apunte al directorio creado previamente. Cuando ahora utilice Portage, éste tendrá en cuenta aquellos ebuilds para no eliminarlos/sobreescribirlos la próxima vez que ejecute `emerge --sync`.

3: Software no mantenido por Portage

En algunos casos querrá configurar, instalar y mantener programas por sí mismo sin que Portage automatice el proceso, incluso aunque Portage pueda suministrarle esos programas. Conocidos son los casos de las fuentes del kernel y los controladores de nvidia. Puede configurar Portage para que conozca cuando un determinado paquete ha sido instalado manualmente en el sistema. Este proceso recibe el nombre de *inyectar* y está soportado por Portage a través del archivo `/etc/portage/profile/package.provided`.

Por ejemplo, si quiere que Portage le informe sobre [vanilla-sources-2.6.11.6](#) el cual ha sido instalado manualmente, añada la siguiente línea a `/etc/portage/profile/package.provided`:

Code Listing 1: Línea de ejemplo para package.provided

```
sys-kernel/vanilla-sources-2.6.11.6
```


La aplicación de Ebuilds

“En “La aplicación de ebuilds” se informa los pasos que toma el sistema Portage al instalar software y cómo puede hacerlo usted mismo con la aplicación de ebuilds.”

1: Emerge y Ebuild

El programa `ebuild` es un interfaz de bajo nivel para el sistema Portage. Usando esta aplicación se pueden ejecutar acciones en un determinado ebuild. Por ejemplo, uno mismo puede instalar un paquete paso a paso.

El uso de `ebuild` está más bien enfocado a los desarrolladores; por tanto, se puede encontrar más información acerca de `ebuild` en el [Manual del desarrollador](http://www.gentoo.org/proj/en/devrel/handbook/handbook.xml) [\[http://www.gentoo.org/proj/en/devrel/handbook/handbook.xml\]](http://www.gentoo.org/proj/en/devrel/handbook/handbook.xml) (en inglés). Sin embargo, nos será muy útil conocer las etapas por las que pasa Portage durante el proceso de instalación de un determinado paquete, y cómo llamar a los pasos de post-configuración que algunos ebuilds permiten realizar.

2: Instalación manual de software

Cada vez que ejecutamos `ebuild` con un determinado fichero ebuild, se verifica que las sumas de control de todos los ficheros involucrados sean iguales a aquéllas que aparecen en el fichero acompañante Manifest o en `files/digest-<nombre>-<versión>`. Lo que tiene lugar tras la descarga de las fuentes.

Para descargar las fuentes usando `ebuild`, ejecutamos:

Code Listing 1: Descargando las fuentes

```
# ebuild ruta/al/ebuild fetch
```

Si la suma de control md5 (`md5sum`) del ebuild no coincide con la listada en el fichero Manifest, o la de una de las fuentes descargadas no coincide con el

listado en el fichero `files/digest-<paquete>`, aparecerá un error similar a este:

Code Listing 2: Fallo en la suma de control del ebuild

```
!!! File is corrupt or incomplete. (Digests do not match)
>>> our recorded digest: db20421ce35e8e54346e3ef19e60e4ee
>>> your file's digest: f10392b7c0b2bbc463ad09642606a7d6
o
!!! El fichero está corrompido o incompleto (Los resúmenes no coinciden)
>>> nuestro resumen almacenado: db20421ce35e8e54346e3ef19e60e4ee
>>> el resumen de su fichero: f10392b7c0b2bbc463ad09642606a7d6
```

La línea que sigue a este mensaje indicará el fichero problemático.

Si estamos seguros que tanto las fuentes descargadas como el propio ebuild son válidos, se puede regenerar el Manifest y el fichero `digest-<paquete>` usando la funcionalidad 'digest' de [ebuild](#):

Code Listing 3: Regeneración de manifest y digest

```
# ebuild ruta/al/ebuild digest
```

. . . : Descomprimiendo las fuentes

Para descomprimir los programas fuente en `/var/tmp/portage` (o en cualquier otro directorio que se haya especificado en `/etc/make.conf`), se usa la funcionalidad 'unpack' de [ebuild](#):

Code Listing 4: Descomprimiendo las fuentes

```
# ebuild ruta/al/ebuild unpack
```

Lo que ejecutará la función `src_unpack()` del ebuild (si no se ha definido dicha función simplemente se realizará una descompresión de las fuentes). En este paso también se aplican los parches necesarios.

. . . : Compilando las fuentes

El paso siguiente en el proceso de instalación es compilar las fuentes. La funcionalidad de compilar de [ebuild](#) se encarga de esta etapa ejecutando la función `src_compile()` en el ebuild. También incluye los pasos de configuración, si fuera pertinente.

Code Listing 5: Compilando las fuentes

```
# ebuild ruta/al/ebuild compile
```

Si quisiéramos cambiar las instrucciones de compilación deberíamos editar la función `src_compile()` del `ebuild`. Sin embargo, también se puede "engañar" a Portage para que crea que la aplicación `ebuild` ha acabado los pasos de compilación creando un fichero vacío llamado `.compiled` en el directorio de trabajo.

Code Listing 6: Informando a Portage de que se han acabado los trabajos de compilación

```
# touch .compiled
```

... : Instalando los ficheros en una ubicación temporal

En el próximo paso, Portage instalará todos los ficheros necesarios en una ubicación temporal. Este directorio contendrá entonces todos los ficheros que serán instalados en el sistema en producción. Esto se hace mediante la funcionalidad 'install' de `ebuild`, que ejecuta la función `src_install()` del `ebuild`:

Code Listing 7: Instalando los ficheros

```
# ebuild ruta/al/ebuild install
```

... : Instalando los ficheros en el sistema de ficheros real

El paso final es copiar todos los ficheros en el entorno de producción y registrarlos en la base de datos de Portage. El programa `ebuild` llama a este paso "qmerge" que puede descomponerse en los pasos siguientes:

- Ejecutar la función `pkg_preinst()` si se ha definido
- Copiar los ficheros en el sistema
- Registrar los ficheros en la base de datos de Portage
- Ejecutar la función `pkg_postinst()` si se ha definido

A través de la funcionalidad `qmerge` de `ebuild` realizamos estos pasos:

Code Listing 8: Instalando los ficheros en el entorno de producción

```
# ebuild ruta/al/ebuild qmerge
```

... : Limpiando el directorio temporal

Finalmente, se debe limpiar el directorio temporal usando la funcionalidad `clean` de `ebuild`:

Code Listing 9: Limpiando el directorio temporal

```
# ebuild ruta/al/ebuild clean
```

3: Características adicionales de ebuild

Usando la funcionalidad `merge` de `ebuild` se puede ejecutar los comandos `fetch`, `unpack`, `compile`, `install` y `qmerge` de una sola vez:

Code Listing 10: Instalando software

```
# ebuild ruta/al/ebuild merge
```

... : Acciones de configuración de un paquete

Algunas aplicaciones incluyen instrucciones para configurar posteriormente el paquete en nuestro sistema. Estas instrucciones pueden ser interactivas y por lo tanto no son ejecutadas automáticamente. Para ejecutar estos pasos de configuración, listados en la función `config()` (opcional) del `ebuild`, se usa la funcionalidad `config` de `ebuild`:

Code Listing 11: Configurar un paquete

```
# ebuild ruta/al/ebuild config
```

... : Creando un paquete (RPM o no)

Portage puede crear también el paquete binario de un `ebuild` o incluso un fichero RPM. Para crear estos archivos se usan las funcionalidades `package` o `rpm` de `ebuild`. Hay algunas diferencias entre ambas:

- La funcionalidad `package` es muy similar a la funcionalidad `merge`, ya que ejecuta todos los pasos necesarios (descargar, descomprimir, compilar,

- instalar) antes de crear el paquete.
- La funcionalidad rpm crea un paquete RPM a partir de los ficheros creados después de haber ejecutado la funcionalidad install del [ebuild](#).

Code Listing 12: Creación de paquetes

```
(Para un paquete binario compatible con Portage)
# ebuild ruta/al/ebuild package

(Para un paquete RPM)
# ebuild ruta/al/ebuild rpm
```

Pero los ficheros RPM creados no contienen información a cerca de las dependencias del ebuild.

4: Más información

Están disponibles las siguientes páginas de man para más información relativa a Portage, la aplicación ebuild y los ficheros ebuild:

Code Listing 13: Man pages

```
$ man portage (el propio Portage)
$ man emerge (el comando emerge)
$ man ebuild (el comando ebuild)
$ man 5 ebuild (la sintaxis del fichero ebuild)
```

También está disponible más información destinada al desarrollador en [Developers Handbook](http://www.gentoo.org/proj/en/devrel/handbook/handbook.xml) [http://www.gentoo.org/proj/en/devrel/handbook/handbook.xml] (en inglés).

Part 4

Configuración de Redes en Gentoo

Una guía amplia para trabajar con redes en Gentoo.

Contents:

4.1. Iniciándonos	160
4.1.1. Comenzando	160
4.2. Configuración Avanzada	162
4.2.1. Configuración Avanzada	162
4.2.2. Dependencias de red	164
4.2.3. Nombre de variables y valores	165
4.3. Trabajo Modular con Redes	166
4.3.1. Módulos de Red	166
4.3.2. Manejadores de interfaces	166
4.3.3. DHCP	167
4.3.4. Modem ADSL	169
4.3.5. APIPA Direccionamiento Privado Automático (Automatic Private IP Addressing)	169
4.3.6. Bonding	170
4.3.7. Puentes (soporte para 802.1d)	170
4.3.8. Dirección MAC	171
4.3.9. Haciendo túneles	172
4.3.10. VLAN (soporte para 802.1q)	172
4.4. Redes Inalámbricas	174
4.4.1. Introducción	174
4.4.2. WPA Supplicant	174
4.4.3. Wireless Tools	177
4.4.4. Definiendo la configuración de la red por ESSID	181
4.5. Agregando Funcionalidad	182
4.5.1. Puntos de enlace standard en las funciones	182
4.5.2. Puntos de enlace para las herramientas inalámbricas	184
4.6. Manejo de Redes	186
4.6.1. Gestión de la red	186
4.6.2. ifplugd	186

Iniciándonos

“Una guía para poner rápidamente a funcionar el interfaz de red en la mayoría de ambientes comunes.”

1: Comenzando

Note: Este documento asume que se tiene correctamente configurado el kernel, los módulos para su hardware y que conoce el nombre de su interfaz hardware. También asumiremos que se está configurando eth0, pero podría tratarse de eth1, wlan0, etc.

Note: Este documento requiere que este ejecutándose baselayout-1.11.11 o superior.

Para comenzar a configurar su tarjeta de red, necesitamos que el sistema RC (runlevel scripts o initscripts) de Gentoo la reconozca. Esto se hace creando un enlace simbólico desde [net.lo](#) a [net.eth0](#) en `/etc/init.d`

Code Listing 1: Crear el enlace simbólico net.eth0 apuntando a net.lo

```
# cd /etc/init.d
# ln -s net.lo net.eth0
```

El sistema RC de Gentoo ahora conoce la interfaz. También necesita saber cómo configurar la nueva interfaz. Todas las interfaces de red son configuradas en `/etc/conf.d/net`. A continuación se muestra un ejemplo de configuración para DHCP y direcciones estáticas:

Code Listing 2: Ejemplos de /etc/conf.d/net

```
# Para DHCP
config_eth0=( "dhcp" )

# Para IP estática usando notación CIDR
config_eth0=( "192.168.0.7/24" )
routes_eth0=( "default via 192.168.0.1" )

# Para IP estática usando notación de máscara de red
config_eth0=( "192.168.0.7 netmask 255.255.255.0" )
routes_eth0=( "default gw 192.168.0.1" )
```


Note: Si no especifica una configuración para la interfaz entonces se asume DHCP.

Note: CIDR (Classless InterDomain Routing) Originariamente, las direcciones IPv4 fueron clasificadas en A, B, o C. Esta clasificación temprana no tuvo en cuenta la gran popularidad que obtendría Internet, y ahora corre el riesgo de no tener suficientes direcciones únicas. CIDR es un esquema de direccionamiento que permite que una dirección IP designe muchas direcciones IP. Una dirección IP CIDR es igual que una dirección IP normal excepto que termina con una barra invertida seguida de un número; por ejemplo: 192.168.0.0/16. CIDR se describe en [RFC 1519](http://rfc.net/rfc1519.html) [http://rfc.net/rfc1519.html].

Ahora que ya hemos configurado nuestra interfaz, podemos iniciarla o detenerla utilizando los siguientes comandos:

Code Listing 3: Comenzar y detener los scripts de red

```
# /etc/init.d/net.eth0 start
# /etc/init.d/net.eth0 stop
```

Important: Cuando este solucionando problemas con la red, se recomienda configurar `RC_VERBOSE="yes"` en `/etc/conf.d/rc` para obtener mayor información sobre lo que está ocurriendo.

Ahora que ya ha iniciado y detenido la interfaz de red, quizá quiera que comience automáticamente cuando Gentoo arranca. Aquí tiene cómo hacerlo. El último comando "rc" indica a Gentoo que inicie todos los scripts en el nivel de ejecución actual que no hayan sido iniciados ya.

Code Listing 4: Configurando una interfaz de red para que inicie en el arranque

```
# rc-update add net.eth0 default
# rc
```

Configuración Avanzada

“Aquí aprendemos acerca de cómo funciona la configuración - hace falta conocer esta sección para aprender a trabajar con redes modularmente.”

1: Configuración Avanzada

La variable `config_eth0` es el corazón de la configuración de una interfaz. Es una lista de instrucciones de alto nivel para configurar la interfaz (`eth0` en este caso). Cada comando en la lista de instrucciones se ejecuta de manera secuencial. La interfaz será evaluada como OK si, al menos, un comando funciona.

Aquí tiene una lista de instrucciones integradas:

Comando	Descripción
<code>null</code>	No hace nada
<code>noop</code>	Si la interfaz está funcionando y existe una dirección entonces aborta la configuración con éxito.
una dirección IPv4 o IPv6	Añade la dirección a la interfaz
<code>dhcp</code> , <code>adsl</code> o <code>apipa</code> (o un comando propio perteneciente a un módulo de terceras partes)	Ejecuta el módulo que proporciona el comando. Por ejemplo "dhcp" ejecutará un módulo que proporcione dhcp, que pudiera ser uno cualquiera de los siguientes: <code>dhcpcd</code> , <code>udhcpc</code> , <code>dhclient</code> o <code>pump</code> .

Si un comando falla, puede especificar un comando de retorno (fallback). El retorno tiene que coincidir exactamente con la estructura de la configuración.

Puede encadenar estos comandos. Aquí se muestran algunos ejemplos reales:

Code Listing 1: Ejemplos de configuración

```
# Añadir tres direcciones IPv4
config_eth0=(
    "192.168.0.2/24"
    "192.168.0.3/24"
    "192.168.0.4/24"
)

# Añadir una dirección IPv4 y dos IPv6
config_eth0=(
    "192.168.0.2/24"
    "4321:0:1:2:3:4:567:89ab"
    "4321:0:1:2:3:4:567:89ac"
)

# Mantener la dirección asignada por el kernel, a menos que la
interfaz se caiga, entonces asignar otra vía DHCP. Si DHCP falla entonces
añadir una dirección estática determinada mediante APIPA

config_eth0=(
    "noop"
    "dhcp"
)
fallback_eth0=(
    "null"
    "apipa"
)
```

Note: Cuando se utiliza el módulo `ifconfig` y se añade más de una dirección, se crean alias de interfaz para cada dirección extra. De esta manera los dos ejemplos anteriores tendrán interfaces `eth0`, `eth0:1` y `eth0:2`. No se puede hacer nada especial con estas interfaces ya que el kernel y otros programas simplemente tratan `eth0:1` y `eth0:2` como `eth0`.

Important: ¡La orden de retorno es importante! Si no especificamos la opción `"null"`, el comando `"apipa"` solamente sería ejecutado si el comando `"noop"` falla.

Note: APIPA y DHCP serán tratados más adelante.

2: Dependencias de red

Los scripts en `/etc/init.d` pueden depender de una interfaz de red específica o, simplemente, de `net` (red). "net" puede definirse en `/etc/conf.d/rc` ya que puede significar cosas distintas utilizando la variable [RC_NET_STRICT_CHECKING](#).

Valor	Descripción
none	El servicio net se considera siempre funcionando
no	Básicamente significa que al menos uno de los servicios net.* sin contar net.lo debe estar funcionando. Esto podría ser usado por los usuarios de equipos portátiles que tienen una conexión wifi y una estática y solamente quieren tener una activa para que el servicio net aparezca como funcionando.
lo	Es la misma que la opción 'no' pero net.lo también cuenta. Esto puede ser útil para la gente que no le importa que alguna interfaz en concreto se active durante el arranque.
yes	Con esto TODAS las interfaces de red DEBEN estar activas para que el servicio 'net' se considere funcionando.

Pero, ¿y que pasa si `net.br0` depende de `net.eth0` y `net.eth1`? `net.eth1` podría ser un dispositivo wireless o ppp que necesita configurarse antes de añadirse al puente. Esto no puede hacerse en `/etc/init.d/net.br0` ya que es un enlaces simbólico a `net.lo`.

La respuesta es hacer nuestra propia función `depend()` en `/etc/conf.d/net`

Code Listing 2: dependencia de net.br0 en /etc/conf.d/net

```
# Puede utilizar cualquier tipo de dependencia (use, after, before)
# que puede encontrar en los actuales scripts

depend_br0() {
    need net.eth0 net.eth1
}
```

Para una lectura más detallada sobre dependencias, consulte la sección "[Guiones de Inicio](#)"¹ en el manual de Gentoo.

¹: http://www.gentoo.org/doc/es/handbook/handbook-x86.xml?part=2&chap=4#doc_chap4

3: Nombre de variables y valores

Los nombre de variables son dinámicos. Normalmente sigue la estructura `variable_${interface|mac|ssid|apmac}`. Por ejemplo, la variable `dhcpcd_eth0` guarda los valores para las opciones de dhcpcd para eth0 y `dhcpcd_essid` los valores para dhcpcd cuando cualquier interfaz se conecta al ssid "ssid".

Sin embargo, no hay ninguna regla que indique que los nombre de las interfaces sean ethx. De hecho, muchas interfaces wireless tienen nombres como wlanx, rax o ethx. También, algunas interfaces definidas por el usuario como pueden ser puentes puede tener cualquier nombre, como foo. Para hacer la vida un poco más interesante, los puntos de acceso wireless pueden tener nombres con caracteres no alfanuméricos - esto es importante porque puede configurar los parámetros de red por ESSID.

La desventaja de todo esto es que Gentoo usa variables bash para la red - y bash no puede utilizar nada fuera de caracteres alfanuméricos ingleses. Para solucionar esta limitación cambiamos cada carácter que no sea alfanumérico inglés por un carácter "_".

Otra desventaja de bash es el contenido de las variables - algunos caracteres necesitan especificarse de manera especial. Esto se hace utilizando \ delante del carácter. A continuación tenemos una lista de caracteres especiales que necesitamos indicar de esta manera. ", ' y \.

En este ejemplo utilizamos ESSID wireless ya que puede contener un amplio abanico de caracteres. Debemos utilizar ESSID `My "\ NET`:

Code Listing 3: ejemplo de nombre para la variable

```
# Esto funciona, pero el dominio no es válido
dns_domain_My___NET="My \\ \ NET"

# Lo que hay arriba configura el dominio dns a My "\ NET cuando una
# tarjeta wireless se conecta a un AP cuyo ESSID es My "\ NET
```

Trabajo Modular con Redes

“Gentoo permite trabajar con redes de manera flexible - aquí les explicamos cómo escoger entre distintos clientes DHCP, configuración de “bonding”, configuración de puentes, redes virtuales (VLANs) y más.”

1: Módulos de Red

Ahora tenemos soporte para guiones de red, lo cual significa que podemos fácilmente añadir soporte para nuevos tipos de interfaces y módulos de configuración mientras mantenemos la compatibilidad con los actuales.

Los módulos se cargan por defecto si el paquete que los requiere está instalado. Si especifica un modulo aquí que no tiene su paquete instalado, entonces obtendrá un error acerca del paquete que necesita instalar. Lo ideal sería que solamente use la configuración con módulos si tiene dos o más paquetes que proporcionen el mismo servicio y necesita marcar su preferencia de uno sobre los otros.

Code Listing 1: Preferencias en los módulos

```
# Preferir iproute2 sobre ifconfig
modules=( "iproute2" )

# También se puede especificar otros módulos para una interfaz
# En este caso preferimos a udhcpc sobre dhcpcd
modules_eth0=( "udhcpc" )

# También podemos especificar módulos que no queremos utilizar -
# por ejemplo, puede querer utilizar un suplicante o linux-wlang-ng
# para controlar la configuración inalámbrica, pero querer seguir
# configurando la red por ESSID asociado.
modules=( "!iwconfig" )
```

2: Manejadores de interfaces

Proporcionamos dos manejadores de interfaces: ifconfig e iproute2. Hará falta uno de estos para cualquier tipo de configuración de red.

El ifconfig es utilizado por defecto en Gentoo y se incluye en el perfil del sistema. El iproute2 es un paquete más potente y flexible pero no se incluye por defecto.

Code Listing 2: Para instalar iproute2

```
# emerge sys-apps/iproute2

# Para preferir iproute2 sobre ifconfig si ambos están instalados
modules=( "iproute2" )
```

Ya que ifconfig e iproute2 hacen cosas muy parecidas, nos permitimos usar la misma configuración básica para ambos. Los ejemplos a continuación funcionarán sin importar cuál módulo tenga instalado.

Code Listing 3: Ejemplos de ifconfig e iproute2

```
config_eth0=( "192.168.0.2/24" )
config_eth0=( "192.168.0.2 netmask 255.255.255.0" )

# También podemos especificar la dirección de difusión
config_eth0=( "192.168.0.2/24 brd 192.168.0.255" )
config_eth0=( "192.168.0.2 netmask 255.255.255.0 broadcast 192.168.0.255" )
```

3: DHCP

El DHCP es un medio para obtener información de red (dirección IP, servidores DNS, puerta de enlace, etc) de un servidor DHCP. Si existe un servidor DHCP en su red, apenas tendrá que decirle a cada cliente que utilice DHCP y el servidor se encargará de configurar el resto. Por supuesto, tendrá que configurar otras cosas la red inalámbrica, el ppp, etc. u otros si hacen falta antes de poder utilizar DHCP.

DHCP puede ser proporcionado por: dhclient, dhcpcd, pump or udhcpc. Cada módulo DHCP tiene sus propios pros y contras - aquí tiene un pequeño resumen:

Módulo DHCP	Paquete	Pros	Contras
dhclient	net-misc/dhcp	Creado por la ISC, la misma gente que hacen el software dns BIND Muy configurable	Configuración excesivamente complicada Software bastante hinchado No se puede obtener servidores NTP desde DHCP No envía el nombre de host por defecto
dhcpcd	net-misc/dhcpcd	Durante largo tiempo ha sido la opción por defecto de Gentoo No depende de	Ya no es mantenido por los desarrolladores Puede ser lento en ocasiones No pasa a segundo plano si el lapso de arrendamiento es infinito

		herramientas externas	
pump	net-misc/pump	Ligero No depende de herramientas externas	Ya no es mantenido por los desarrolladores No es fiable, especialmente a través de modems No se puede obtener servidores de NIS desde DHCP
udhcp	net-misc/udhcp	Ligero - el cliente más pequeño de dhcp Creado para sistemas embebidos	No muy probado - ninguna distribución lo usa por defecto No puede definir un tiempo de espera (timeout) mayor de 3 segundos

Si tiene más de un cliente DHCP instalado, hará falta especificar cuál utilizar - sino, utilizaremos dhcpcd por defecto, si está disponible.

Para enviar opciones específicas al módulo dhcp, utilizamos `module_eth0="..."` (cambie "module" por el nombre del módulo dhcp que vaya a utilizar - por ejemplo: `dhcpcd_eth0`).

Tratamos que el servicio DHCP sea relativamente agnóstico, de manera que soportamos los siguientes comandos usando la variable `dhcp_eth0`. Por defecto no se configuran ninguno de ellos.

- `release` - suelta la dirección IP para ser re-utilizada
- `nodns` - no sobre-escriba el `/etc/resolv.conf`
- `nontp` - no sobre-escriba el `/etc/ntp.conf`
- `nonis` - no sobre-escriba el `/etc/yp.conf`

Code Listing 4: Ejemplo de configuración DHCP en `/etc/conf.d/net`

```
# Solamente necesita hacerlo si tiene más de un módulo
# instalado
modules=( "dhcpcd" )

config_eth0=( "dhcp" )
dhcpcd_eth0="-t 10" # Espera agotada (timeout) después de 10
# segundos
dhcp_eth0="release nodns nontp nonis" # Solamente obtenga la
# dirección IP
```

Note: `dhcpcd`, `udhcp` y `pump` envían el nombre de máquina al servidor DHCP por defecto así que no necesita especificarlo más.

4: Modem ADSL

Primero necesitamos instalar el software ADSL.

Code Listing 5: Instalar el paquete rp-pppoe

```
# emerge net-dialup/rp-pppoe
```

Warning: baselayout-1.11.x solamente soporta PPPOE. Esperamos soportar PPPOA en futuras versiones.

Ahora necesitamos configurar eth0 para que sea una interfaz ADSL e introducir nuestro nombre de usuario.

Code Listing 6: Configurar eth0 para ADSL

```
config_eth0=( "adsl" )  
user_eth0="nombre-usuario"
```

Finalmente, necesitamos definir nuestro nombre de usuario y contraseña en /etc/ppp/pap-secrets

Code Listing 7: ejemplo /etc/ppp/pap-secrets

```
# Ojo: el * es importante  
"nombre-usuario" * "contraseña"
```

5: APIPA Direccionamiento Privado Automático (Automatic Private IP Addressing)

APIPA intenta encontrar una dirección libre en el rango 169.254.0.0-169.254.255.255 haciendo arping a direcciones aleatorias en ese rango para el interfaz. Si no se obtiene respuesta, se asigna esa dirección al interfaz.

Esto es útil solamente en redes donde no hay servidor DHCP y no hay conexión

directa al Internet y que todos los demás computadores también usen APIPA.

Para soporte APIPA, haga emerge net-misc/iputils o net-analyzer/arping.

Code Listing 8: Configuración de APIPA en /etc/conf.d/net

```
# Intentar DHCP primero - y si falla, entonces usar APIPA
config_eth0=( "dhcp" )
fallback_eth0=( "apipa" )

# Usar únicamente APIPA
config_eth0=( "apipa" )
```

6: Bonding

Para unir enlaces con bonding/trunking haga emerge net-misc/ifenslave.

El "bonding" se utiliza para aumentar el ancho de banda hacia la red. Si tiene dos interfaces que van a usar la misma red, puede unirlos (bond, en inglés) para que las aplicaciones vean solo uno, pero que en realidad usa ambos interfaces.

Code Listing 9: Configuración del bonding en /etc/conf.d/net

```
Para unir dos interfaces
slaves_bond0="eth0 eth1 eth2"

# Tal vez no quiera asignarle dirección IP al interfaz conjunto
config_bond0=( "null" )

# Depend de eth0, eth1 y eth2, ya que podría requerir
configuración adicional
depend_bond0() {
    need net.eth0 net.eth1 net.eth2
}
```

7: Puentes (soporte para 802.1d)

Para soportar puentes, haga emerge net-misc/bridge-utils.

Los puentes se usan para unir redes. Por ejemplo, puede tener un servidor

conectando al internet vía modem ADSL y una tarjeta inalámbrica para que otros computadores se conecten al internet por medio del modem ADSL. Se puede crear un puente para unir ambos interfaces.

Code Listing 10: Configurando un puente en /etc/conf.d/net

```
# Configure el puente - para más detalles: "man brctl"
brctl_br0=( "setfd 0" "sethello 0" "stp off" )

# Para agregar puertos al puente br0
bridge_br0="eth0 eth1"

# Hará falta configurar los puertos a valores nulos para no
iniciar dhcp
config_eth0=( "null" )
config_eth1=( "null" )

# Finalmente, déle una dirección al puente - para esto puede usar DHCP también
config_br0=( "192.168.0.1/24" )

# Depend de eth0 y eth1 ya que pueden requerir configuración adicional
depend_br0() {
    need net.eth0 net.eth1
}
```

Important: Para usar algunas configuraciones de puente, tal vez tenga que consultar la documentación de Configuración por medio de variables.

8: Dirección MAC

No hace falta hacer emerge para poder cambiar la dirección MAC de un interfaz a otra específica. Sin embargo, si necesita cambiarlo a una dirección aleatoria o una dirección aleatoria de un tipo determinado, entonces necesitará hacer emerge net-analyzer/macchanger.

Code Listing 11: Ejemplo de cambio de una dirección MAC

```
# Establecer una dirección MAC de un interfaz
mac_eth0="00:11:22:33:44:55"

# Para cambiar aleatoriamente solo los últimos 3 bytes
mac_eth0="random-ending"

# Para cambiar aleatoriamente entre tipos de conexión
# físicamente iguales (por ejemplo: fibra óptica, cobre, inalámbrica),
# para todos los proveedores
mac_eth0="random-samekind"

# Para cambiar aleatoriamente entre cualquier tipo físico de
# conexión (por ejemplo fibra óptica, cobre, inalámbrica),
# para todos los proveedores
mac_eth0="random-anykind"

# Generación completamente aleatoria -
# ADVERTENCIA: algunas direcciones MAC generadas de esta manera
# tal vez NO se comporten de la forma esperada
mac_eth0="random-full"
```

9: Haciendo túneles

No se requiere hacer emerge de paquete alguno para túneles, ya que el manejador del interfaz se encarga de esto.

Code Listing 12: Configuración de túneles en /etc/conf.d/net

```
# Para túneles GRE
iptunnel_vpn0="mode gre remote 207.170.82.1 key 0xffffffff ttl 255"

# Para túneles IPIP
iptunnel_vpn0="mode ipip remote 207.170.82.2 ttl 255"

# Para configurar el interfaz
config_vpn0=( "192.168.0.2 peer 192.168.1.1" )
```

10: VLAN (soporte para 802.1q)

Para soporte VLAN, haga emerge net-misc/vconfig.

Las redes virtuales son un grupo de dispositivos de red que se comportan como si estuviesen conectados a un solo segmento de red - aunque no lo estén. Miembros de una VLAN solo pueden ver miembros de la misma VLAN, aunque no compartan la misma red física.

Code Listing 13: Configuración de VLANs en /etc/conf.d/net

```
# Especificar los números de las VLAN para el interfaz así
# Por favor asegurar que los ID de las VLANs
# NO tengan ceros a la izquierda
vlans_eth0="1 2"

# También se puede configurar la VLAN
# vea la página man de vconfig para más detalles
vconfig_eth0=( "set_name_type VLAN_PLUS_VID_NO_PAD" )
vconfig_vlan1=( "set_flag 1" "set_egress_map 2 6" )

# Configure el interfaz como siempre
config_vlan1=( "172.16.3.1 netmask 255.255.254.0" )
config_vlan2=( "172.16.2.1 netmask 255.255.254.0" )
```

Important: Para usar algunas configuraciones de VLAN, tal vez haga falta consultar la documentación de Configuración por medio de variables.

Redes Inalámbricas

“Las redes inalámbricas no son tan sencillas. ¡Esperamos poder ponerlas a trabajar!”

1: Introducción

Actualmente soportamos la configuración de redes inalámbricas con wireless-tools o wpa_supplicant. Lo importante para recordar es que una red inalámbrica se configura de manera global y no por interfaz.

La mejor opción es wpa_supplicant, pero no soporta todos los controladores. Para una lista completa de controladores soportados, lea la [página de wpa_supplicant](http://hostap.epitest.fi/wpa_supplicant) [http://hostap.epitest.fi/wpa_supplicant]. Además, actualmente wpa_supplicant sólo puede conectarse a SSIDs para los cuales haya sido configurado.

Los wireless-tools soportan casi todas las tarjetas y controladores, pero no puede conectarse a PAs (puntos de acceso) con WPA solamente.

Warning: Actualmente el controlador linux-wlan-ng no está soportado por baselayout. Esto es debido a que linux-wlan-ng tiene su propio configurador y su configuración es completamente diferente a las demás. Se rumora que los desarrolladores de linux-wlan-ng están pensando en cambiar su configuración a wireless-tools. Al suceder esto, podremos utilizar linux-wlan-ng con baselayout.

2: WPA Supplicant

El [WPA Supplicant](http://hostap.epitest.fi/wpa_supplicant) [http://hostap.epitest.fi/wpa_supplicant] es un paquete que permite conectarse a puntos de acceso habilitados con WPA. Su configuración está en estado cambiante, por estar todavía en beta, pero generalmente funciona a la perfección.

Code Listing 1: Instalación de wpa_supplicant

```
# emerge net-wireless/wpa_supplicant
```

Important: Debe activar CONFIG_PACKET en su kernel para que funcione wpa_supplicant.

Ahora debemos configurar `/etc/conf.d/net` para preferir wpa_supplicant a wireless-tools (si ambos están instalados, wireless-tools será usado por defecto).

Code Listing 2: Configurar `/etc/conf.d/net` para wpa_supplicant

```
# Preferimos wpa_supplicant sobre wireless-tools
modules=( "wpa_supplicant" )

# Es importante decirle a wpa_supplicant cuál controlador
# debemos usar ya que todavía no adivina bien
wpa_supplicant_eth0="-Dmadwifi"
```

Note: Si está usando el controlador host-ap, hará falta colcoar el interfaz en modalidad manejada antes de poder usarlo correctamente con wpa_supplicant. Para esto puede usar `iwconfig_eth0="mode managed"` en `/etc/conf.d/net`.

Eso fue sencillo, ¿verdad? Sin embargo, todavía nos queda configurar wpa_supplicant, que es algo más difícil, dependiendo de cuán seguros son los PAA a los cuales tratamos de conectarnos. A continuación mostramos un ejemplo simplificado del archivo `/etc/wpa_supplicant.conf.example`, parte del paquete wpa_supplicant.

Code Listing 3: Un ejemplo de /etc/wpa_supplicant.conf

```
# No cambie la siguiente línea, a riesgo que no funcione
ctrl_interface=/var/run/wpa_supplicant

# Asegúrese que sólo root puede leer la configuración de WPA
ctrl_interface_group=0

# Permítale a wpa_supplicant encargarse del barrido y selección de los PA
ap_scan=1

# Caso sencillo: WPA-PSK, PSK como contraseña-frase ASCII,
# permitiendo todas las encriptaciones válidas
network={
    ssid="sencillo"
    psk="contrasenha-frase muy secreta"
    # Mientras más alta la prioridad, más rápido nos conectaremos
    priority=5
}

# Igual que el ejemplo anterior, pero barriendo por un SSID
# específico (para los PAs que rechazan transmitir el SSID)
network={
    ssid="segundo ssid"
    scan_ssid=1
    psk="contrasenha-frase muy secreta"
    priority=2
}

# Solamente estamos usando WPA-PSK. Se aceptada cualquier encriptación válida
network={
    ssid="ejemplo"
    proto=WPA
    key_mgmt=WPA-PSK
    pairwise=CCMP TKIP
    group=CCMP TKIP WEP104 WEP40
    psk=06b4be19da289f475aa46a33cb793029d4ab3db7a23ee92382eb0106c72ac7bb
    priority=2
}

# Conexión sin encriptar (sin WPA, o IEEE 802.1X)
network={
    ssid="prueba-noencriptada"
    key_mgmt=NONE
}

# Conexión con clave WEP compartida (sin WPA, o IEEE 802.1X)
network={
    ssid="prueba-estatica-wep"
    key_mgmt=NONE
    wep_key0="abcde"
    wep_key1=0102030405
    wep_key2="1234567890123"
    wep_tx_keyidx=0
    priority=5
}
```



```
# Conexión con clave WEP compartida (sin WPA, o IEEE 802.1X)
# usando autenticación con clave compartida IEEE 802.11
network={
    ssid="prueba-estatica-wep2"
    key_mgmt=NONE
    wep_key0="abcde"
    wep_key1=0102030405
    wep_key2="1234567890123"
    wep_tx_keyidx=0
    priority=5
    auth_alg=SHARED
}

# Red IBSS/ad-hoc con WPA-None/TKIP
network={
    ssid="prueba adhoc"
    mode=1
    proto=WPA
    key_mgmt=WPA-NONE
    pairwise=NONE
    group=TKIP
    psk="contrasenha-frase secreta"
}
```

3: Wireless Tools

Los [Wireless Tools](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html) [http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html] proveen una manera genérica para configurar los interfaces inalámbricos básicos hasta el nivel de seguridad WEP. Aunque WEP es un sistema de seguridad débil, también es el más prevalente.

La configuración de Wireless Tools es controlado por algunas variables principales. El archivo ejemplo de configuración a continuación deberá describir todo lo requerido. Algo para mantener presente es que ninguna configuración significa "conectarse al punto de acceso sin encriptación que tenga la señal más fuerte" - de manera que siempre tratará de conectarse con lo que sea.

Code Listing 4: Instalando los wireless-tools

```
# emerge net-wireless/wireless-tools
```

Note: Aunque pueda almacenar su configuración inalámbrica en /etc/conf.d/wireless, esta guía le recomienda hacerlo en /etc/conf.d/net.

Important: Hará falta consultar la documentación acerca de nombres de variables.

Code Listing 5: Configuración ejemplo con iwconfig en /etc/conf.d/net

```
# Prefiera iwconfig sobre wpa_supplicant
modules=( "iwconfig" )

# Configure las claves WEP para los PAs ESSID1 y ESSID2
# Pueden configurarse hasta 4 claves WEP, pero solo una puede
# estar activa a la vez, así que tenemos un índice por defecto
# de [1] para la clave [1] y luego otra vez para cambiar la clave activa a [1]
# Hacemos esto en caso que defina otros ESSIDs para usar claves WEP que no sean 1
# El prefijar la clave con s: significa que es una clave ASCII,
# porque sino es una clave HEX
# enc open especifica seguridad abierta (máxima seguridad)
# enc restricted especifica seguridad restringida (menos segura)
key_ESSID1="[1] s:suclaveaquí key [1] enc open"
key_ESSID2="[1] aaaa-bbbb-cccc-dd key [1] enc restricted"

# Los ejemplos a continuación solo funcionan una vez que se
# haya barrido para los PAs disponibles.

# A veces se registra más de un PA, así que hará falta definir
# un orden de preferencia para conectarse
preferred_aps=( "ESSID1" "ESSID2" )
```

. . . : Afinamiento en la selección de puntos de acceso

Puede agregar opciones adicionales para afinar la selección de puntos de acceso, pero normalmente esto no hace falta.

Puede decidirse si conectarse solamente a PAs preferidos o no. Por defecto, si falla todo lo configurado y nos podemos conectar a un PA no encriptado, entonces se hará. Esto puede ser controlado con la variable `associate_order` (orden asociado). Sigue una tabla de valores y como estos ejercen este control.

Valor	Descripción
any	(cualquiera) Comportamiento por defecto
preferredonly	(solo preferidos) Solamente nos conectaremos a PAs visibles en la lista de preferidos
forcepreferred	(obligado a preferidos) Nos conectaremos obligatoriamente a los PAs en el orden preferido aunque no se encuentren en un barrido
forcepreferredonly	(solamente preferidos obligados) No barrer buscando PAs - solo tratar de conectar a cada uno en orden
forceany	(obligar a cualquiera) Igual que forcepreferred y conectar a cualquier otro disponible

Finalmente podemos hacer una lista negra de PAs (`blacklist_aps`) y seleccionar un

único PA (unique_ap). blacklist_aps funciona de manera similar a preferred_aps. unique_ap es un valor si o no (yes o no) que determina si un segundo interfaz inalámbrico se puede conectar al mismo punto de acceso que el primer interfaz.

Code Listing 6: Ejemplo de blacklist_aps y de unique_ap

```
# A veces nunca queremos conectarnos a algunos PAs
blacklist_aps=( "ESSID3" "ESSID4" )

# Si tiene más de una tarjeta inalámbrica, puede determinar
# que cada interfaz se asocie al mismo PA o no
# Los valores son "yes" y "no"
# El valor por defecto es "yes"
unique_ap="yes"
```

... : Modos Ad-Hoc y Master

Si desea establecerse como un nodo Ad-Hoc al no lograr conectarse a algún PA en modo manejado, puede hacerlo también.

Code Listing 7: Usar modo ad-hoc en caso de no conectar en modo manejado

```
adhoc_essid_eth0="This Adhoc Node"
```

Y ¿qué tal conectarse a redes Ad-Hoc o funcionar en modo Master para convertirse en PA? ¡Aquí tenemos una configuración justo para eso! Tal vez tenga que especificar alguna clave WEP como se muestra arriba.

Code Listing 8: Ejemplo de configuración ad-hoc/master

```
# Establezca el modo - que puede ser manejado (por defecto), ad-hoc o master
# No todos los manejadores soportan todos los modos
mode_eth0="ad-hoc"

# Establezca el ESSID del interfaz
# En modo manejado, esto obliga al interfaz intentar conectarse al
# ESSID especificado solamente
ssid_eth0="This Adhoc Node"

# Usamos el canal 3 si no se establece otro
channel_eth0="9"
```

Important: A continuación citamos la documentación wavelan de BSD encontrado en la documentación de NetBSD Existen 14 canales posibles; Se nos informa que los canales 1-11 son legales para Norteamérica , los canales 1-13 para la mayor parte de Europa, los canales 10-13 para Francia y sólo el canal 14 para el Japón. En caso de duda, por favor refiérase a la documentación que acompaña la tarjeta o el PA. Asegúrese que el canal que selecciona es el mismo del PA (o de la otra tarjeta en caso de ser red ad-hoc). Para las tarjetas vendidas en Norteamérica y la mayor parte de Europa, es el canal 3 por defecto; en Francia el 11 y en el Japón es el 14.

. . . : Solucionando problemas con Wireless Tools

Existen más variables que sirven para ayudar a configurar y operar una red inalámbrica por problemas ambientales o del manejador. Aquí presentamos una tabla de otras cosas para intentar.

<i>Variable</i>	<i>Valor por defecto</i>	<i>Descripción</i>
iwconfig_eth0		Vea la página man de iwconfig para detalles acerca de qué enviar a iwconfig
iwpriv_eth0		Vea la página man de iwpriv para detalles acerca de qué enviar a iwpriv
sleep_scan_eth0	0	La cantidad en segundos a dormir antes de intentar un barrido. Esto hace falta cuando el manejador/firmware requiere de más tiempo para activarse antes de poder usarlos.
sleep_associate_eth0	5	La cantidad en segundos a esperar para que el interfaz se asocie con el PA antes de continuar al siguiente.
associate_test_eth0	MAC	Algunos manejadores no resetean la dirección MAC asociada a una inválida cuando pierden la señal o intentan asociarse. Algunos manejadores no resetean el nivel de calidad cuando pierden la señal o intentan asociarse. La configuraciones válidas son MAC, quality y all.
scan_mode_eth0		Algunos manejadores tienen que hacer el barrido en modo ad-hoc, así que, si falla el barrido, intente configurar aquí como ad-hoc.
iwpriv_scan_pre_eth0		Envía algunos comandos iwpriv al interfaz antes del barrido Vea la página man de iwpriv para más detalles.
iwpriv_scan_post_eth0		Envía algunos comandos iwpriv al interfaz después del barrido Vea la página man de iwpriv para más detalles.

4: Definiendo la configuración de la red por ESSID

Cuando se conecta a ESSI1 requiere una dirección IP estática y cuando se conecta a ESSID2, requiere DHCP. De hecho, casi todos las variables de módulo pueden cambiarse por ESSID. A continuación mostramos cómo se hace.

Note: Esto funciona sin importar si usa WPA Supplicant o Wireless Tools.

Important: Necesitará consultar la documentación variable name.

Code Listing 9: Reemplazar configuración de red por ESSID

```
config_ESSID1=( "192.168.0.3/24 brd 192.168.0.255" )
routes_ESSID1=( "default via 192.168.0.1" )

config_ESSID2=( "dhcp" )
fallback_ESSID2=( "192.168.3.4/24" )
fallback_route_ESSID2=( "default via 192.168.3.1" )

# Podemos definir servidores DNS y otras cosas también
# NOTA: DHCP reemplazará esta configuración si no se
# le pide que no lo haga
dns_servers_ESSID1=( "192.168.0.1" "192.168.0.2" )
dns_domain_ESSID1="algun.dominio"
dns_search_domains_ESSID1="busque.eneste.dominio busca.enaquel.dominio"

# Reemplace usando la dirección MAC del PA
# Esto es muy útil si usa distintas ubicaciones con el mismo ESSID
config_001122334455=( "dhcp" )
dhcpcd_001122334455="-t 10"
dns_servers_001122334455=( "192.168.0.1" "192.168.0.2" )
```

Agregando Funcionalidad

“Si tiene espíritu de aventura, puede agregar sus propias funciones para trabajar con redes.”

1: Puntos de enlace standard en las funciones

Cuatro funciones pueden ser definidas, llamables antes y después de operación de arranque/parada. Sus nombres se componen usando el nombre de interfaz primero, para que una función puede controlar múltiples interfaces.

Los valores de retorno de las funciones "preup" y "predown" deben ser cero (éxito) indicando que la configuración o la des-configuración de los interfaces pueden continuar. Si "preup" devuelve un no-cero, se detendrá la configuración del interfaz. Si "predown" devuelve un valor no-cero, no se permite que el interfaz continúe desconfigurándose.

Los valores de retorno de las funciones "postup" y "postdown" son ignorados, ya que no hay nada por hacer si indican algún fallo.

`${IFACE}` es el interfaz que será encendido/apagado, `${IFVAR}` es `${IFACE}` convertido a un nombre de variable válido según bash.

Code Listing 1: Ejemplo de funciones pre/post up/down

```
preup() {
    # Comprobación del enlace de la interfaz antes de activarla.
    # Esto solamente funciona con algunos adaptadores de red y necesita
    # que este instalado el paquete mii-diag.
    if mii-tool ${IFACE} 2> /dev/null | grep -q 'no link'; then
        ewarn "No link on ${IFACE}, aborting configuration"
        return 1
    fi

    # Comprobación del enlace en la interfaz antes de activarla.
    # Esto solamente funciona con algunos interfaces de red y necesita
    # que el paquete ethtool este instalado.
    if ethtool ${IFACE} | grep -q 'Link detected: no'; then
        ewarn "No link on ${IFACE}, aborting configuration"
        return 1
    fi

    # Recordamos devolver cero si todo fue correcto
    return 0
}

predown() {
    # Por defecto en el script de inicio se comprueba si
    # el sistema (root) está montado por NFS y en ese caso no permitir
    # que las interfaces se desactiven. Tenga en cuenta que, si
    # especifica una función predown() sobrecribirá esta regla.
    # Aquí la tiene en caso de que todavía quiera utilizarla...
    if is_net_fs /; then
        error "root filesystem is network mounted -- can't stop
${IFACE}"
        return 1
    fi

    # Recordamos devolver cero si todo fue correcto
    return 0
}

postup() {
    # Esta función podría utilizarse, por ejemplo, para
    # registrarse en un servicio dinámico de DNS. Otra posibilidad
    # sería enviar un correo cada vez que la interfaz se conecta.
    return 0
}

postdown() {
    # Esta función está en gran parte por completar ...
    # Aún no hemos pensado en nada interesante que pueda hacer ;-)
    return 0
}
```

2: Puntos de enlace para las herramientas inalámbricas

Note: Esto no funcionará con WPA suplicante - pero las variables `#{ESSID}` y `#{ESSIDVAR}` están disponibles en la función `postup()`.

Se pueden definir dos funciones ejecutables antes y después de la función asociada referida. Estas funciones se nombran antecediéndole el nombre del interfaz, de manera que la misma función puede controlar múltiples interfaces.

El valor retornado por la función pre-asociada debe ser 0 (éxito) para indicar que la configuración o desconfiguración del interfaz puede continuar. Si la función pre-asociada retorna un valor no-cero, se interrumpirá la configuración del interfaz.

El valor retornado por la función post-asociada se ignora, ya que no hay nada que hacer si llegase a indicarse una falla.

Al `#{ESSID}` se le asigna el ESSID exacto del PA al cual se está conectando ya que `#{ESSIDVAR}` es `#{ESSID}`, convertido a un nombre de una variable permitido por bash.

Code Listing 2: Funciones pre/post asociación

```
preassociate() {
    # A continuación se agregan 2 variables
    # de configuración leap_user_ESSID y leap_pass_ESSID. Cuando
    # ambos esten configurados para el ESSID al que se conectan,
    # corremos el guión LEAP de CISCO.

    local user pass
    eval user="\${leap_user_${ESSIDVAR}}\\"
    eval pass="\${leap_pass_${ESSIDVAR}}\\"

    if [[ -n ${user} && -n ${pass} ]]; then
        if [[ ! -x /opt/cisco/bin/leapscript ]]; then
            eend "For LEAP support, please emerge
net-misc/cisco-aironet-client-utils"
            return 1
        fi
        einfo "Waiting for LEAP Authentication on \"${ESSID//\\\/}\\\"
if /opt/cisco/bin/leapscript ${user} ${pass} | grep -q 'Login
incorrect'; then
            ewarn "Login Failed for ${user}"
            return 1
        fi
    fi

    return 0
}

postassociate() {
    # Esta función existe mayormente para completar ...
    # aunque no he pensado en algo interesante para hacer acá
    # todavía ;-))

    return 0
}
```

Note: Las variables `${ESSID}` y `${ESSIDVAR}` no están disponibles en las funciones `pre-down()` y `post-down()`.

Manejo de Redes

“Para usuarios de portátiles o usuarios que mueven sus computadores entre distintas redes.”

1: Gestión de la red

Si el ordenador está en continuo movimiento, no siempre tendrá un cable ethernet conectado o un punto de acceso disponible. También, puede querer que la red funcione automáticamente cuando un cable ethernet se conecte o se encuentre un punto de acceso.

Aquí se encuentran algunas herramientas que puede ayudar a gestionar esto.

Note: Este documento solamente tiene habla sobre ifplugd, pero hay alternativas que también se pueden tener en cuenta como puede ser quickswitch

2: ifplugd

[ifplugd](http://0pointer.de/lennart/projects/ifplugd/) [http://0pointer.de/lennart/projects/ifplugd/] es un demonio que arranca y para las interfaces de red cuando un cable se conecta o se desconecta. También puede gestionar la detección asociándose a un punto de acceso o cuando uno nuevo entra dentro del radio de detección.

Code Listing 1: Instalar ifplugd

```
# emerge sys-apps/ifplugd
```

La configuración de ifplugd es bastante clara. El archivo de configuración se encuentra en `/etc/conf.d/ifplugd`. Para más detalles o conocer para qué se utilizan las variables, [man ifplugd](#).

Code Listing 2: Ejemplo de configuración de ifplug

```
# Definir que interfaces vamos a monitorizar
INTERFACES="eth0"

AUTO="no"
BEEP="yes"
IGNORE_FAIL="yes"
IGNORE_FAIL_POSITIVE="no"
IGNORE_RETVAL="yes"
POLL_TIME="1"
DELAY_UP="0"
DELAY_DOWN="0"
API_MODE="auto"
SHUTDOWN="no"
WAIT_ON_FORK="no"
MONITOR="no"
ARGS=""

# Parámetros adicionales de ifplugd para una interfaz específica.
# Tenga en cuenta que se ignora la variable global, cuando una variable como
# esta es configurada para un interfaz concreto.
MONITOR_wlan0="yes"
DELAY_UP_wlan0="5"
DELAY_DOWN_wlan0="5"
```