

# Gentoo Linux: Filosofía e idea general

José Alberto Suárez López

*Gentoo Linux*

[bass@gentoo.org](mailto:bass@gentoo.org)

versión 0.4-r1, 6 de Septiembre del 2003

## Resumen

**Gentoo Linux** es una Meta-Distribución que ha destacado en el mundo del Software Libre gracias a seguir una filosofía o camino completamente distinto al del resto de distribuciones más conocidas y usadas como puedan ser RedHat, Debian, etc. Esta pretende ser una introducción esta distribución y a un uso básico de esta.

## 1. Índice

1. Introducción
2. Diferencia con el resto
3. Qué es Portage
  - corazón
  - paquetes
  - emerge
  - slots
  - ccache
  - distcc
  - archivos de configuración
  - fuentes
4. Edge Bleeding (al filo)
  - Optimización
  - Flags
  - Prelink
5. Conclusión
6. Bibliografía

## 2. Introducción

[Gentoo](#) Linux nació desde el principio con la idea de la flexibilidad y la optimización, por ello como el creador de Gentoo Linux ([Daniel Robbins](#)), deja muy claro en el [documento de introducción](#) de Gentoo Linux, Gentoo Linux es una Meta-Distribución. Gentoo Linux fue concebida desde un principio para dar un gran rango de flexibilidad desde el mismo momento que se procede a instalarla. Gentoo Linux es un "sabor" especial de Linux que puede ser automáticamente optimizada y adaptada para el uso que necesitemos. El gran rendimiento y flexibilidad son marcas de la casa de esta distribución. Gracias a una tecnología llamada "Portage", Gentoo Linux puede ser un servidor seguro, una estación de desarrollo, un escritorio profesional, un sistema para juegos, una solución empotrada o cualquier otra cosa, esta adaptabilidad es lo que hace que Gentoo Linux sea una Meta-Distribución."

## 3. Diferencia con el resto

La gran diferencia de Gentoo Linux con el resto de distribuciones actuales consiste en facilitar el trabajo, al igual que las otras, pero sin olvidar que hay usuarios más avanzados que requieren un mayor nivel de complejidad, flexibilidad, y por que no, de divertimento. Todo esto se consigue basando Gentoo Linux en LFS ([Linux from scratch](#)) añadiéndole herramientas y tecnologías que facilitan su administración, mantenimiento y uso. Esta diferencia hace que aquellos usuarios avanzados o con ganas de adentrarse aún más en el mundo de Linux (también llamados Geeks) prefieran esta distribución, ya que permite hacerte tu propia distribución adaptándola a tus necesidades y de una manera más simple que empezando desde cero. Y siguiendo su estupenda [documentación](#) se aprende realmente lo que se está haciendo, no simplemente se hace (aunque también existe esta posibilidad). Gentoo Linux fue por este camino en lugar de seguir el camino de las interfaces gráficas, e instalaciones preconfiguradas que restan flexibilidad y en algunos casos entorpecen ciertas tareas. Esto no quiere decir que no pueda existir un instalador gráfico para Gentoo Linux, de hecho existe, así como otros proyectos de interfaces gráficas pero son de un uso minoritario y totalmente opcional. También existe un proyecto para la creación de un [script](#) que ayuda a dar todos los pasos necesarios en la instalación de Gentoo Linux. Así mismo también existen proyectos de creación de frontends para ciertas características del sistema, como por ejemplo el fantástico "etc-update" que permite un manejo simple y flexible de los archivos de configuración actualizados de Gentoo Linux.

Además Gentoo Linux, tiene muy claro que sus características, tanto presentes como futuras, son las que los usuarios piden. [La lista de desarrollo](#) esta abierta para que cualquier usuario pueda sugerir, o implementar nuevas características, que si son de buen agrado para la comunidad y no perjudican al funcionamiento de Gentoo Linux serán incorporadas a su debido tiempo.

Cómo toda distribución Gentoo Linux tiene cientos de fallos o bugs de distintas categorías conocidos, y otros tantos no conocidos, pero siempre a disposición del

usuario en el sistema [bugzilla de Gentoo](#). Desde este sistema se puede acceder a toda la información sobre fallos, no solo en el sistema, ni en paquetes concretos, sino también sobre la documentación o la agregación de nuevas características.

Además todo el software desarrollado o modificado por Gentoo Linux se encuentra accesible a través del [cvs de Gentoo](#) para poder descargar o consultar sus códigos fuentes.

La comunidad es uno de los puntos fuertes de Gentoo y el crecimiento de esta ha sido espectacular. Prueba de ello es el hecho de la tremenda base de datos que se ha originado en los [foros](#) de Gentoo con casi medio millón de entradas con una media desde que se crearon de casi 1000 al día, y con casi 30.000 usuarios y más de 2 Gb de información.

## 4. ¿Qué es "Portage"?

"Portage" es el corazón de Gentoo Linux, y se encarga de muchas funciones claves. En primer lugar "Portage" es el sistema de distribución de Software de Gentoo Linux. Para obtener el último software para Gentoo Linux, solo es necesario ejecutar un comando: 'emerge sync'. Este comando le pide a "Portage" que actualice la copia local del "Árbol de Portage" desde internet. La copia local del árbol de Portage contiene una gran colección de "scripts" que pueden ser usados por Portage para crear e instalar los últimos paquetes de Gentoo. Actualmente Gentoo Linux tiene:

- (comando ejecutado el 22 de Junio del 2003)

```
minime root # gentool-package-count
Number of categories :      76
Number of ebuilds    :    9106
Unique packages     :    4540
```

- (comando ejecutado el 6 de Septiembre del 2003)

```
asuka root # gentool-package-count
Number of categories :      85
Number of ebuilds    :   10904
Unique packages     :    5272
```

Es decir, casi 5300 paquetes únicos y cerca de 11000 si contamos los paquetes con varias versiones disponibles, en un total de 85 categorías. Y nuevos paquetes son añadidos continuamente al árbol de Portage para el deleite de los usuarios de Gentoo Linux.

Portage es también un sistema de creación e instalación de paquetes. Cuando quieres instalar un paquete solo necesitar hacer :

```
emerge nombre_paquete
```

Y Portage automáticamente descargará y creará una versión adaptada del paquete para tus especificaciones exactas, optimizándolo para el hardware que sea necesario y asegurándote que las características extras del paquetes que necesites serán también instaladas y por supuesto las que no necesites no serán instaladas. ¿Por qué instalar el soporte para arts en zinf si prefieres usas esd?

Portage además es capaz de mantener tu sistema completamente actualizado. Escribiendo:

```
emerge --update world
```

Te asegura que todos los paquetes que se engloban en "world" serán automáticamente actualizados. Gracias a Portage mantener el sistema es mucho más ameno y sencillo.

Además Portage usa una tecnología denominada "sandbox" que hace que un paquete no sea realmente instalado en el sistema si no puede ser instalado correctamente. Complementando a esto los archivos importantes de los paquetes (osea los archivos de configuración) no serán sobrescritos sino almacenados a la espera de una decisión del administrador por cada uno de ellos. Así como otra que permite tener varias versiones de un mismo software trabajando juntas sin conflictos y de una forma transparente al usuario. Esto es los llamados "slots". Con los "slots" podemos tener instalado simultáneamente y funcionando, por ejemplo las librerías qt-2.x y las librerías qt-3.x y a la vez distintas versiones de programas que utilizan estas librerías, usando una u otra según hallamos especificado en el USE (que se verá más adelante) o lo requiera el programa.

Portage es capaz de usar otras tecnologías como [ccache](#) que permite rebajar tremendamente el tiempo de compilación. Y [distcc](#) (ver [documentación](#)) que permite distribuir una compilación a través de una red para que el trabajo de compilación sea hecho por varios procesadores simultáneamente por lo que el tiempo de compilación puede llegar a ser realmente reducido.

Portage también es capaz de gestionar dependencias intrínsecas y a distintos niveles, así como de diferenciar entre dependencias necesarias para la creación del paquete y dependencias necesarias para su ejecución. Y por supuesto es capaz de elegir que dependencia necesita un paquete en función del tipo de sistema y de las características de este.

Portage ha bebido de muchos otros sistemas similares tomando lo mejor de ellos, y añadiéndoles nuevas características y capacidades. Así como posee la facilidad y la robustez del comando 'apt-get' también toma algunas de sus características del sistema de paquetes de FreeBSD, y como no del sistema de paquetes de RedHat. Además cualquier usuario familiarizado con alguno de ellos se encontrará cómodo con el comando "emerge" de Portage o en su defecto con su similar del rpm el ["epm"](#). Con lo que la migración de estos sistemas a Portage es realmente cómoda e intuitiva.

Para más información acerca de Portage se puede leer el [manual](#), además de la [guía rápida](#).

## 5. Edge Bleding (al filo)

Gracias a las peculiaridades de Gentoo Linux, los usuarios son capaces de mantener un sistema realmente muy actualizado. Gentoo Linux poseía paquetes para la versión 2 de GNOME a los 5 minutos de esta ser liberada. Así como de kde 3.1 u otras piezas de software. Pero Gentoo Linux no necesita tener varios repositorios con distintos paquetes de software para conseguir un sistema estable y flexible a al vez. En un mismo árbol de portage se encuentran TODAS las ramas necesarias desde la inestable hasta la estable para procesadores x86 (intel) hasta para Mac o sparc. Una simple configuración permite usar mezcla y separar estas ramas sin ningún tipo de problema.

## 6. Optimización

Gracias a la flexibilidad de Gentoo y de la configuración centralizada del Portage (make.conf). Somos capaces de construir un paquete para las características exactas de nuestro sistema y de nuestras necesidades. Este es el llamado sistema de "FLAGSz de ÜSE".

### 6.1. USE

El sistema de ÜSE" nos permite especificar cuales son nuestras necesidades, desde si preferimos usar gnome a kde hasta si no queremos tener soporte para gis en el sistema. Gracias a esto se puede construir el mozilla contra las librerías gtk2 y tan solo el navegador, por lo que esta aplicación en concreto estará adaptada a nuestras necesidades y optimizada para nuestro uso que será claramente perceptible. Gracias al sistema USE somos capaces de especificar una gran cantidad de características configurables por paquete, siempre adaptado al sistema que deseemos tener y tan solo con lo que necesitemos usar. Esto facilita en mucho la administración y mantenimiento del sistema y su flexibilidad a la hora de dedicar un sistema para una tarea en concreto.

- Ejemplo de uso de varias características de Portage y del USE entre ellas

```
USE="x gnome"
emerge --pretend -v bitchx
Calculating dependencies ...done!
[ebuild N ] media-libs/audiofile-0.2.3-r1
[ebuild N ] media-sound/esound-0.2.29 +tcpd -alsa
[ebuild N ] gnome-base/ORBit-0.5.17 +nls
[ebuild N ] dev-util/intltool-0.25
[ebuild U ] media-libs/freetype-2.1.4 [1.3.1-r3] +doc +zlib -prebuilt
[ebuild N ] x11-misc/ttmkfdir-3.0.9
[ebuild N ] x11-base/opengl-update-1.5
[ebuild N ] media-libs/fontconfig-2.2.0-r2 +doc
[ebuild N ] app-arch/cabextract-0.6
```

```

[ebuild N ] x11-base/xfree-4.3.0-r2  -3dfx -sse +mmx -3dnow +xml +truetype +nls -cjk +
[ebuild N ] x11-libs/gtk+-1.2.10-r10  +nls
[ebuild N ] media-libs/giflib-4.1.0-r3  -X +gif
[ebuild N ] media-libs/imlib-1.9.14-r1
[ebuild N ] app-text/docbook-sgml-1.0
[ebuild N ] gnome-base/gnome-libs-1.4.2  +doc +nls -kde
[ebuild R ] net-irc/bitchx-1.0.19-r5  +ssl -esd +gnome -xmms +ncurses -ipv6 -gtk -cjk

USE="-x"
emerge --pretend -v bitchx
[ebuild R ] net-irc/bitchx-1.0.19-r5  +ssl -esd +gnome -xmms +ncurses -ipv6 -gtk -cjk

```

Como se observa en el ejemplo hay una gran diferencia que consiste en que uso le vayamos a dar al sistema. Esto no es una simple resolución de dependencias, sino que con un solo paquete (bitchx en este caso) podemos tener varias posibilidades. En el primer comando se usa el USE por defecto, por lo que al intentar instalar BitchX (cliente IRC de consola) intenta instalar librerías gráficas, ¿Por qué? muy simple, existe una versión de BitchX que usa las librerías gráficas "gtkz esta depende de otras tantas. Simplemente especificando en el "USE" que no queremos usar las opciones gráficas (USE="x") estamos delimitando que se compilará, y no solo esto, sino que también su comportamiento y relación entre si. Un ejemplo más caro puede ser el uso de la variable "java" en el "USE".<sup>al</sup> compilar mozilla. al incluir tal variable (USE="java"), se incluirá una nueva dependencia en mozilla, que es la máquina virtual java. Esto descargará e instalara la máquina virtual java, no solo eso, sino que hará lo que necesite para que mozilla tenga el soporte java. La variable "USE" (así como cualquier otra variable), puede definirse de una manera global en el archivo principal de configuración (/etc/make.conf) o en la propia línea de comandos. Se puede obtener un gran nivel de afinamiento gracias al USE y de configuración de características, no solo de ciertos paquetes sino también del propio sistema.

Aquí otro ejemplo de cuan configurable puede llegar a ser un paquete tan "básico como un editor:

```

emerge vim -pv
[ebuild R ] app-editors/vim-6.1-r21  -gnome +gpm -gtk -gtk2 +ncurses +nls +perl +pytho

```

Existen varias utilidades de Gentoo que nos permiten un control más intuitivo de las variables USE. Como por ejemplo [ufed](#) un buen gui que nos permite seleccionar los USE a introducir en el make.conf con una descripción de cada uno. Así mismo existen dos tipo de variables USE. Las que se pueden aplicar a todos los ebuilds, o las que son específicas de un ebuild. Por lo que el nivel de configuración puede ser tan profundo como se desee.

## 6.2. Flags (variables de compilación)

Así como el USE, en Gentoo Linux es muy fácil especificar que variables debe usar nuestro compilador para crear un paquete, y esto esta integrado totalmente con

el sistema Portage. Gracias al uso de compiladores y librerías de última generación como el gcc-3.x o las glibc-2.3.x el sistema es capaz de optimizarse para un tipo de hardware muy concreto y así sacar el mayor rendimiento de este.

Esto se hace de una forma muy simple, tan solo hay que usar la variable CFLAGS= la cual es pasada al gcc (o al compilador elegido como por ejemplo icc). Un ejemplo de esta configuración sería:

```
CFLAGS="-mcpu=pentiumpro -O3 -pipe"
```

Lo cual aprovecha todas las instrucciones y tecnologías de este procesador en concreto rompiendo la compatibilidad del binario generado con otros procesadores. Hay un gran número de opciones, pero no son específicas de Gentoo, dependen del compilador (en este caso gcc). Para más información acerca de esto visitar la [web de opciones de gcc](#) o de una forma más clara está [referencia](#) en freehackers.

Para facilitar esta tarea existe una pequeña utilidad en Gentoo llamada [genflag](#) la cual consta de una serie de comandos que nos ayudan a configurar el sistema:

```
asuka root # info2flags
CHOST="i686-unknown-gnu-linux"
CFLAGS="-march=athlon -O3 -pipe"
CXXFLAGS="-march=athlon -O3 -pipe"
asuka root # info2host
MAINARCH="x86"
SUBARCH="duron"
```

### 6.3. Prelink

Básicamente [el prelink](#) modifica los ejecutables consiguiendo que se inicien hasta un 50% más rápido. ¿Magia? no. Para que prelink funcione debemos tener un sistema con una versión de glibc mayor o igual a la 2.3.1. Mandrake Linux usa esta tecnología de serie. Prelink funciona especialmente bien en aplicaciones hechas en c++ (léase KDE). Prelink modifica el ejecutable deseado añadiéndole el linkado de librerías que antes necesitaba hacer exteriormente. Aunque tiene cierto parecido, no es lo mismo que un ejecutable estático. Además Prelink es completamente reversible y seguro para el sistema. Tanto Mandrake como RedHat confían en él para acelerar sus sistemas de escritorio.

## 7. Conclusión

Con Gentoo Linux nos creamos un sistema completamente adaptado a nuestras necesidades y a las características de nuestro hardware. Consiguiendo una gran flexibilidad y optimización de paso. Gracias a Portage nuestro sistema es tremendamente sencillo de administrar y automatizar. Sin perder un ápice de flexibilidad.

Pero aunque nuestro sistema este muy optimizado y compilado desde cero la mejora de velocidad no tiene porque notarse ni ser obvia, pero el rendimiento del sistema será tremendamente aumentado.

No hace mucho fue publicado una comparativa de velocidad (usando el comando time) entre Debian, Gentoo y Mandrake. En esta comparativa se ejecutaron varias aplicaciones y se midió el tiempo en cada una por separado. En esta comparativa Mandrake conseguía las mejores marcas de velocidad (algo que no extraña teniendo en cuenta que en la comparativa fue la única que usó prelink), además la Gentoo que se usó no fue compilada desde cero. Entre Debian y Gentoo el resultado fue reñido, obteniendo resultados equivalentes, demostrando que no es factible hacer una instalación de Gentoo Linux ya que no se consigue una velocidad superior a una distribución ordinaria. Pues bien aprovechando el paso de un usuario de RedHat desde tiempos inmemoriales a Gentoo hicimos nuestra propia prueba.

En un mismo ordenador con una Mandrake 9.1 muy bien configurada se ejecutaron una serie de aplicaciones y se midieron con el comando time. Más tarde se hizo una instalación de Gentoo Linux 1.4 GRP (una versión de Gentoo Linux totalmente binaria exceptuando el kernel) y se ejecutaron los mismos comandos. Y por último se aplicó el prelink a esas aplicaciones (no a todo el sistema) y se volvió a medir. Cabe destacar que la persona que realizó estas pruebas es un recién llegado a Gentoo por lo que el sistema seguramente se podría haber optimizado mucho más, Además el sistema Gentoo usado no fue compilado desde cero para obtener el máximo rendimiento. Aquí están los resultados:

- Características de la máquina:

Pentium III (Coppermine) 733 Mhz 256 Kb de Caché. 1461 bogomips  
128 Mb de RAM

- Mandrake 9.1 (prelink de serie):

Mozilla	5,34	5,45	5,32
NetBeans	38,37	14,58	20,43
Kmail	0,42	0,42	0,46

- Gentoo 1.4 GRP Pentium3 :

Mozilla	2,88	2,82	2,88
NetBeans	18,98	18,35	18,51
Kmail	0,39	0,97	0,33

- Gentoo 1,4 GRP Pentium3 (prelink selectivo) :

Mozilla	2,8	2,78	2,82
NetBeans	(no aplicable)		
Kmail	0,03	0,02	0,01

En definitiva, los resultados hablan por sí solos. Si en la comparativa anteriormente mencionada, Mandrake se situaba por encima de Debian y una Gentoo un tanto malograda, al superar en esta comparativa una Gentoo básica a Mandrake, también supera a Debian ;)

Nota: no, lo del 0,03 no es una errata. Gracias a [Juán Jesús Prieto](#) de Eneo tecnología por la comparativa objetiva.

## 8. Bibliografía

- <http://www.gentoo.org> - Web principal
- <http://www.gentoo.org/doc/es/gentoo-x86-install.xml> - Guía de instalación de x86
- <http://www.gentoo.org/doc/en/gentoo-ppc-install.xml> - Guía de instalación de PPC
- <http://www.gentoo.org/doc> - Documentación en varios idiomas
- <http://forums.gentoo.org> - Foros de Gentoo en varios idiomas
- <http://www.gentoo.org/main/en/about.xml> - Acerca de Gentoo
- <http://www.linuxfromscratch.org> - Linux from scratch
- <http://glis.sourceforge.net> - Script ayuda instalación
- <http://www.gentoo.org/main/en/lists.xml> - Listas de correo
- <http://bugs.gentoo.org> - bugzilla de Gentoo
- <http://www.gentoo.org/doc/es/portage-user.xml> - Guía de usuario de Portage
- <http://www.gentoo.org/doc/es/portage-manual.xml> - Manual de portage
- <http://www.gentoo.org/doc/es/use-howto.xml> - Guía de USE
- <http://www.gentoo.org/doc/es/distcc.xml> - Guía de Distcc
- <http://www.gentoo.org/main/en/contract.xml> - Contrato social de Gentoo
- <http://www.gentoo.org/news/en/gwn/gwn.xml> - Boletín semanal de Gentoo