



# Gentoo && Portage

LINUX /usr/portage/\*

SERGIO ALEIXANDRE SOMOZA

eMail: [seralso@polinux.upv.es](mailto:seralso@polinux.upv.es)  
JID: [galidor@jabber.org](mailto:galidor@jabber.org)



# // INDEX

- **ABOUT**
  - Pygoscelis Papua
  - Daniel Robbins
  - Larry (The Cow)
  - BUG#: 27727
- **GENTOO Linux**
  - Intro
  - Instalación
  - Stages
  - emerge
  - GRP
  - +/-
- **PORTAGE**
  - Intro
  - CFLAGS
  - USE Flags
  - /etc/make.conf
  - Ebuilds
  - Se queja...
- **CURIOSIDADES**
- **THE END**



# // ABOUT

## *PYGOSCELIS PAPUA*

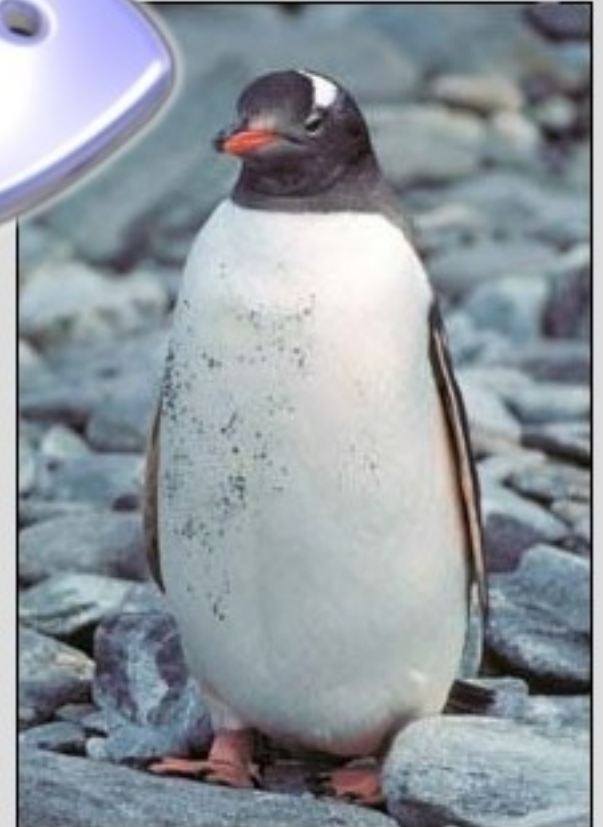
### **Pygoscelis Papua a.k.a. Gentoo Penguin**

Se pronuncia “**Yen-tú**” y es el nombre que le dieron los habitantes de las Islas Malvinas a este curioso pingüino.

En español es conocido como **Pingüino de pico rojo** y es el pingüino que da nombre a la distribución.

**Peso:** 5 ~ 6.2 Kg

**Altura:** 51 ~ 71 cm



# // ABOUT

DANIEL ROBBINS

## // \* DANIEL ROBBINS \* //

Fundador de la distribución Gentoo Linux y escritor de varios artículos.

Trabajó como desarrollador de **Stampede Linux** y más tarde creó su propia distribución **Enoch Linux** antes de crear Gentoo Linux en el 2002.

Un fatídico **26 de Abril de 2004 anunció** que abandonaba su puesto como jefe de desarrollo de Gentoo debido a una deuda de 40.000\$. Desde este momento la propiedad intelectual sobre Gentoo pasó a la **Gentoo Foundation**.

Actualmente es un hombre “libre”, pero para saldar la deuda antes comentada se vio obligado a realizar tareas altamente denigrantes como... trabajar para Microsoft entre el 13 de Junio de 2005 y 9 de Enero de 2006.

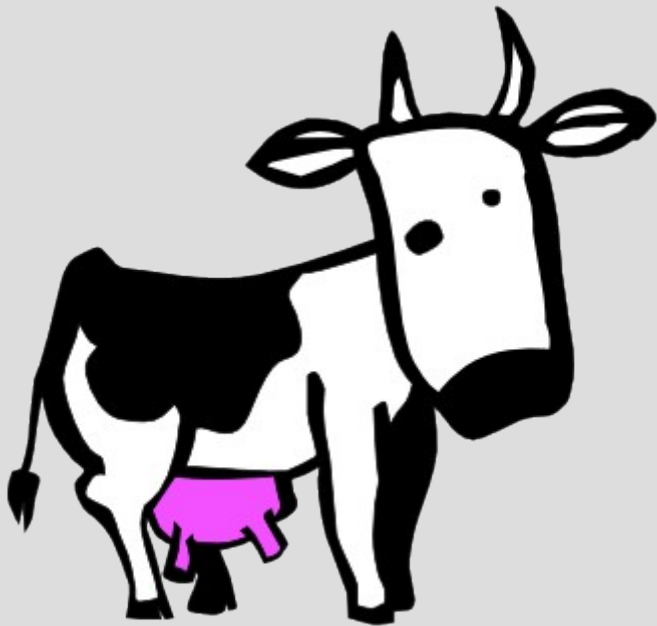
Según Robbins esto lo hizo para:  
*“helping Microsoft to understand open source and community-based projects”*

En la actualidad y desde el 30 de Agosto de 2006 trabaja de nuevo como desarrollador de Gentoo Linux en los proyectos Grub y vmware-workstation-tools.



# // ABOUT

## LARRY (THE COW)



### Larry (The Cow)

“Larry The Cow was a bit frustrated at the current state of Linux distributions...

...until he tried GENTOO LINUX” [gentoo.org](http://gentoo.org)

# // ABOUT

(BUG#: 27727)

**BUG#:** 27727

**Título:** Larry can NOT be a Cow.

**Fecha:** 2003-09-01 13:19 PST

**Estado Actual:** RESOLVED WONTFIX

→ **Dictionary.com** define Cow como: “A domesticated bovine of either sex or any age”. Es por esto por lo que el bug se ha resuelto como WONTFIX.

→ **WONTFIX** = “The problem described is a bug which will never be fixed.”

**De esta forma, Larry se convierte en la primera mascota en tener un BUG.**

## Descripción:

A Cow is femail.  
A Bull is mail.

Reproducible: Didn't try

Steps to Reproduce:

1. Go to [www.gentoo.org](http://www.gentoo.org).
2. Click on About in the navigation bar.
3. View the poster.

## Actual Results:

You see that Larry is a cow.

## Expected Results:

Larry should have been a bull.

# // GENTOO LINUX

## INTRO



- Distribución orientada a usuarios con cierta experiencia en el sistema operativo GNU/Linux.
- Diseñada para ser **modular, portable, fácil de mantener, flexible** y **OPTIMIZADA para la máquina del usuario**.
- Genera sus aplicaciones y utilidades desde el **código fuente**. Pero además, podemos encontrar binarios (GRP) para diversas arquitecturas de aplicaciones muy pesadas de compilar.
- Está siendo desarrollada activamente.
- La versión **1.0** de esta distribución se lanzó el **21 de Marzo del 2002**.
- A partir de la versión de mantenimiento **1.4** (11 de Septiembre de 2003) cambió la nomenclatura de las versiones. Ahora tienen el formato del año en que se libera la versión seguido del ordinal de la release en ese año. La versión más reciente es la **2006.1** (30 de Agosto de 2006).

**NOTA:** La numeración de las versiones se mantiene a pesar de la nueva nomenclatura por años. Podemos comprobarlo en el archivo `/etc/gentoo-release` que contiene la siguiente cadena de caracteres: **"Gentoo Base System version 1.12.6"**

# // GENTOO LINUX

## INSTALACION



La instalación de GENTOO viene explicada al detalle en un documento llamado **HANDBOOK** el cual se puede seguir casi al detalle. Este documento viene adjunto en el LiveCD de instalación.

Para realizar dicha instalación podemos utilizar distintos métodos, el más frecuente consiste en utilizar el LiveCD de Gentoo, pero también se podría realizar desde una instalación previa de otra distribución de GNU/Linux, un LiveCD como puede ser Knoppix, etc... Estos métodos alternativos de instalación están comentados en el siguiente documento: <http://www.gentoo.org/doc/en/altinstall.xml>

El LiveCD de Gentoo podemos obtenerlo aquí: <http://www.gentoo.org/main/en/where.xml>

- **Minimal Install CD:** Contiene un kernel con gran cantidad de módulos que deberemos cargar dependiendo de nuestras necesidades, también contiene un sistema base. Requiere conexión a Internet para completar la instalación.
- **Universal Install CD:** Contiene todo lo necesario para completar la instalación de Gentoo sin necesidad de acceder a Internet. (No disponible para x86 desde la v2006.0)
- **Package CD:** No es un CD de instalación, sino que contiene paquetes precompilados (GRP) para facilitarnos la instalación de algunas aplicaciones como OpenOffice.org, KDE, GNOME...

La instalación de Gentoo puede ser comenzada desde uno de los 3 Stages base que comentamos a continuación.

**NOTA:** 27-2-06 gentoo-es *"También es la primera vez que se libera el instalador gráfico de Gentoo Linux, el cual debuta en el LiveCD x86 y amd64 que reemplazará por completo a los CDs Universal y de paquetes precompilados."*

# // GENTOO LINUX

## STAGES



Los **stages** de instalación son puntos a partir de los cuales se realiza la instalación del sistema.

Son específicos para cada arquitectura.

Podremos descargar los tarballs de Internet o copiarlos del Universal Install CD en el momento de la instalación.

→ **Stage 1:** Particionamiento/Configuración del sistema de archivos, Sincronización del árbol de Portage, Bootstrap, Compilación del sistema, Compilación del kernel y Configuración final.

→ **Stage 2:** Stage 1 con el Bootstrapping realizado.

→ **Stage 3:** Stage 2 con la compilación del sistema realizada.

El handbook de instalación tan sólo da soporte para instalar desde stage 3, si deseamos obtener información sobre otro stage podemos recurrir a las **FAQ** de instalación.

Opcionalmente se puede recompilar el sistema con flags de compilación personalizadas (CFLAGS y USE flags) una vez finalizada la instalación. Los tarballs de los stages 1 y 2 continúan disponibles.

**NOTA:** Es necesario comprender que podemos comenzar la instalación desde cualquiera de los stages no teniendo que realizar los anteriores.

# // GENTOO LINUX

## EMERGE



### /\* EMERGE \*/

Es la utilidad principal para controlar el sistema de paquetes de Gentoo llamado **Portage**. Se trata de un comando bastante complejo con gran cantidad de parámetros y modificadores y está programado en **Python**.

**Funcionamiento básico:** Cuando queremos instalar una aplicación o librería...

- emerge descarga el código fuente de dicha aplicación, así como el de otras aplicaciones o librerías necesarias para satisfacer dependencias.
- Una vez todo ha sido descargado, se descomprime, se compila y se instala en un entorno de seguridad llamado **sandbox**.
- Finalmente, una vez la instalación se ha realizado correctamente en el sandbox, los archivos son instalados en el sistema.

**Parámetros útiles:**

- ◆ **-a --ask:** Solicita confirmación antes de emerger.
- ◆ **-v --verbose:** Muestra información de los paquetes que se van a instalar.
- ◆ **-u --update:** Actualiza los paquetes a la versión más actualizada disponible en stable.
- ◆ **-p --pretend:** Muestra los paquetes que se van a instalar pero no realiza la instalación.
- ◆ **-D --deep:** En conjunción con --update fuerza a actualizar, si es posible, también el árbol de dependencias del paquete que se desea actualizar.
- ◆ **-s --search:** Busca en Portage la cadena de caracteres que pasemos como parámetro.
- ◆ **-C --unmerge:** Borra el paquete indicado.

# // GENTOO LINUX

## EMERGE



```
hersye@kotoko# emerge -s grub
```

\* sys-boot/grub

Latest version available: 0.96-r2

Latest version installed: 0.96-r2

Size of downloaded files: 1,042 kB

Homepage: <http://www.gnu.org/software/grub/>

Description: GNU GRUB boot loader

License: GPL-2

\* app-admin/grubconf

Latest version available: 0.5.1

Latest version installed: [ Not Installed ]

Size of downloaded files: 357 kB

Homepage: <http://grubconf.sourceforge.net/>

Description: Gnome2 based GRUB configuration editor

License: GPL-2



```
hersye@kotoko# emerge -av grub
```

These are the packages that I would merge, in order:

```
Calculating dependencies ...done!
```

```
[ebuild R ] sys-boot/grub-0.96-r2 -custom-cflags -netboot  
-static 1,042 kB
```

```
Total size of downloads: 1,042 kB
```

```
Do you want me to merge these packages? [Yes/No]
```



**Gentoo Reference Platform** fue introducido en la v**1.4** de Gentoo. Se trata de precompilados de paquetes particularmente grandes y populares (OpenOffice.org, KDE, GNOME, Xorg...) e incluso de algunas aplicaciones menores y librerías.

Estos paquetes podemos encontrarlos en el Package CD y en diversos repositorios de Internet, así como ahora en el nuevo LiveCD para x86 y amd64.

Para hacer uso de ellos:

→ **Package CD:** Debemos exportar la variable **PKGDIR** apuntándola a la ruta donde tengamos montado dicho CD de datos y emerger con el parámetro **-k** (--usepkg) o **-K** (--usepkgonly).

```
# export PKGDIR="/mnt/cdrom"  
# emerge -k nombre_del_paquete
```

**NOTA:** Con el parámetro -k el paquete se instalará desde el código fuente si la versión de este es posterior a la del GRP. El parámetro -K fuerza a instalar desde el GRP haya o no versión posterior en código fuente.

→ **Repositorio de Internet:** Exportaremos la variable **PORTAGE\_BINHOST** apuntándola a la URL del servidor donde estén alojados los binarios y emergeremos con los parámetros **-gk** (--getbinpkg --usepkg).

```
# export PORTAGE_BINHOST="ftp://buildhost/gentoo"  
# emerge -gk nombre_del_paquete
```

# // GENTOO LINUX

(+/-)



## /\* A FAVOR \*/

- Permite una mayor optimización de los paquetes gracias a los **USE Flags** y **CFLAGS**.
- Gentoo es una **meta-distribución**, lo que significa que tenemos el completo control sobre qué paquetes son o no instalados y de esta manera podemos instalar el sistema a nuestro gusto.
- Gran cantidad de documentación en Internet:
  - <http://www.gentoo.org/doc/es/list.xml>
  - <http://gentoo-wiki.com>
- Una vez instalada, GENTOO es "**Versionless**". **i.e.** Si instalamos desde la versión 2006.0, se libera la 2006.1 y actualizamos el sistema, entonces la versión también se actualiza. La finalidad de las versiones es para mantener actualizados los paquetes de instalación de GENTOO.

## /\* EN CONTRA \*/

- La instalación es muy larga y compleja... ¡YA NO! GENTOO proporciona instaladores gráficos y asistentes de instalación. Como alternativa podemos instalar una distribución derivada de Gentoo llamada **VidaLinux (VLOS)** que utiliza el instalador Anaconda y dispone de Portage como sistema de paquetes.
- La compilación de algunos paquetes supone una importante carga para el sistema y una importante espera para el usuario. Como posible solución, disponemos de paquetes precompilados **GRP**, un sistema de caché para la compilación llamado **ccache** y de un sistema de compilación distribuida llamado **distcc**.

# // PORTAGE

## INTRO



### // \* PORTAGE \* //

Es el sistema de administración de paquetes que utiliza Gentoo. Está basado en el sistema de paquetes de **BSD** llamado **ports**.

Portage está desarrollado en **Python** y para acceder a sus prestaciones lo hacemos mediante el comando emerge antes mencionado.

Posee una serie de características avanzadas tales como:

- Gestión de dependencias.
- Falsas instalaciones.
- Sandbox durante la compilación.
- Desinstalación segura.
- Paquetes virtuales (virtual/syslog).
- Gestión de ficheros de configuración (etc-update).
- Slots para instalar distintas versiones de un mismo paquete.
- “Personalización” de los paquetes (USE Flags y CFLAGS).

La versión actual de Portage y sobre la que vamos a dar detalles es la **2.1**.

# // PORTAGE

## CFLAGS



### // \* CFLAGS \* //

Se trata de flags de compilación de C, se asocian normalmente con opciones de compilación de GCC. Se utilizan para “personalizar” y optimizar aplicaciones cuando las construimos desde su código fuente (compilación). Este dato es relevante a la hora de hablar de GENTOO pues la gran mayoría de los paquetes se construyen así.

En GENTOO se utilizan frecuentemente para especificar la arquitectura de nuestro computador, esta información es importante para GCC porque le indica cómo optimizar la compilación.

#### **CFLAGS Frecuentes:**

- **-O** : Activa la optimización básica.
- **-O2** : Activa el siguiente nivel de optimización. Es tan seguro como -O y mejora algunos aspectos de la compilación.
- **-O3** : Nivel de optimización superior al -O2. No debe ser activado en sistemas con poca memoria o escaso espacio en disco. Incrementa el tamaño de los binarios resultantes por lo que tardan más en ser cargados desde disco aunque luego en memoria sean más eficaces. Este nivel es conocido por generar errores en algunos paquetes.
- **-march** : Aquí se especifica el nombre por el que conoce GCC la arquitectura de nuestro computador.
- **-pipe** : Indica al GCC que no debe crear archivos temporales sino que debe utilizar pipes para pasar datos a las funciones.

# // PORTAGE

## USE FLAGS



### **// \* USE \***

Portage nos permite, mediante la variable USE, seleccionar qué características del paquete queremos incluir en su compilación de tal manera que podremos deshabilitar aquellas que no nos resulten útiles obteniendo un paquete más ligero.

Para hacer ésto hacemos uso de las **USE Flags** que podremos activar y desactivar mediante diversos métodos.

Encontraremos descripciones de cada USE Flag en `/usr/portage/profiles/use.desc` o en este enlace: <http://www.gentoo.org/dyn/use-index.xml>

A la variable USE podemos acceder a través del archivo `/etc/make.conf` (donde introducimos las USE que serán globales a todos los paquetes) o bien a la hora de hacer un emerge de la siguiente manera:

```
#USE="-xine dvd" emerge -av mplayer
```

Existe también la posibilidad de crear el archivo `/etc/portage/package.use` donde introduciremos una lista de los paquetes y las USE que desearemos activar/desactivar para cada uno con respecto a las globales de `/etc/make.conf`

**NOTA:** Para activar/desactivar las USE de `/etc/make.conf` disponemos de aplicaciones como **ufed** o **euse** (perteneciente al paquete gentoolkit).

# // PORTAGE

## USE FLAGS



```
hersye@kotoko# USE="X" emerge -pv mplayer
```

These are the packages that would be merged, in order:

Calculating dependencies... done!

```
[ebuild R ] media-video/mplayer-1.0_rc1 USE="X* aac aalib alsa cpudetection directfb dts dv dvd encode fbcon gif  
jpeg mad openal opengl png samba sdl speex theora truetype unicode v4l vorbis xv xvid xvmc (-3dfx) (-3dnow)  
(-3dnowext) (-altivec) -amr -arts -bidi -bindist -bl -cdparanoia -custom-cflags -debug -dga -doc -dvb -dvdread -enca  
-esd -ggi -gtk -iconv -ipv6 -jack -joystick -libcaca -lirc -live -livecd -lzo -matrox (-mmx) (-mmxext) -musepack -nas  
-oss (-real) -rtc (-sse) (-sse2) (-svga) -tga -v4l2 (-win32codecs) -x264 -xanim -xinerama (-subtitles%)" LINGUAS="-bg  
-cs -da -de -el -en -es -fr -hu -ja -ko -mk -nl -no -pl -pt_BR -ro -ru -sk -tr -uk -zh_CN -zh_TW" 8,906 kB
```

Total size of downloads: 8,906 kB

# // PORTAGE

## /ETC/MAKE.CONF



```
hersye@kotoko# cat /etc/make.conf
```

```
CFLAGS="-O2 -march=k8 -pipe -fomit-frame-pointer"
```

```
CHOST="x86_64-pc-linux-gnu"
```

```
CXXFLAGS="${CFLAGS}"
```

```
USE="-* alsa bash-completion bzip2 ccache cdrom cups emul-linux jabber mp3  
ncurses opengl openssh qt rar samba ssl usb vorbis xine"
```

```
ACCEPT_KEYWORDS="amd64"
```

```
# Portage
```

```
PORTDIR=/usr/portage
```

```
# Ruta del árbol de portage.
```

```
PORTDIR_OVERLAY=/usr/local/portage
```

```
# Ruta del árbol de ebuilds de terceros.
```

```
PKGDIR=${PORTDIR}/packages
```

```
# Ruta del árbol de precompilados (GRP)
```

```
DISTDIR=${PORTDIR}/distfiles
```

```
# Ruta de los fuentes.
```

```
AUTOCLEAN="yes"
```

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

```
FEATURES="ccache buildpkg parallel-fetch userfetch"
```

```
CCACHE_SIZE="1G"
```

```
GENTOO_MIRRORS="http://linuv.uv.es/mirror/gentoo/ "
```

# // PORTAGE

## /ETC/MAKE.CONF



### // \* /ETC/MAKE.CONF \* //

Dentro de este archivo podemos configurar el comportamiento de Portage y activar/desactivar algunas de sus funcionalidades.

- **ACCEPT\_KEYWORDS** : En esta variable podemos definir si queremos instalar paquetes de la rama testing añadiendo ~ delante de la arquitectura que utilicemos. **i.e.** **ACCEPT\_KEYWORDS="~x86"**  
Con este método todos los paquetes del sistema se instalarán de la rama testing.
- **AUTOCLEAN = "yes | no"** : Limpia automáticamente los paquetes que estén obsoletos o hayan sido actualizados.
- **CCACHE\_SIZE = "size"** : Establece el valor de la caché de compilación. Imprescindible instalar la utilidad ccaché. **emerge ccache**
- **FEATURES** : Define el comportamiento de Portage.
  - **ccache**: Utiliza la caché de compilación.
  - **buildpkg**: Almacena en \$PKGDIR los binarios de todos los paquetes que compilemos.
  - **parallel-fetch**: Permite descargar los fuentes de los paquetes a instalar durante la compilación.
  - **userfetch**: Permite que Portage se ejecute sin permiso de root durante la obtención de fuentes.
- **MAKEOPTS** : Permite especificar el número máximo de compilaciones que se llevarán a cabo de forma simultánea. Es recomendable establecer este valor entre **nCPU+1** y **2\*nCPU+1**
- **GENTOO\_MIRRORS = "url"** : Indica el/los lugar/es de donde debe intentar descargar los archivos antes de probar con la ruta que indica el ebuild.

**NOTA:** Estas son tan sólo algunas de las opciones de las que dispone Portage, para más información podemos acudir a las páginas `man make.conf`



## */\* EBUILDS \*/*

*“En GENTOO no se empaqueta software, se define cómo se construye el software.”* **Ferdy**

Un ebuild es un **script de bash** escrito para que Portage pueda instalar software en el sistema. Cada paquete tiene un ebuild que define cómo debe ser compilado, sus dependencias (si las tiene), la URL de descarga de los fuentes en caso de fallar el mirror por defecto, una descripción del paquete, etc...

Los nombres de los ebuilds consisten en 4 secciones:

`pkg-ver{ _suf{#} }{-r#}.ebuild`

**NOTA:** Las { } delimitan campos opcionales y la # representa un entero distinto de 0.

- **pkg:** Nombre del paquete. **i.e.** `gentoo-sources`
- **ver:** Versión del paquete. **i.e.** `2.6.18`
- **\_suf:** Sufijo que indica el estado de desarrollo. **i.e.** `_rc`
- **-r:** Número de revisión del paquete. **i.e.** `-r3`

`gentoo-sources-2.6.18-r3`

# // PORTAGE

*EBUILDS*



```
DESCRIPTION="the advanced PC speaker beeper"  
HOMEPAGE="http://www.johnath.com/beep/"  
SRC_URI="http://www.johnath.com/beep/${P}.tar.gz"  
  
LICENSE="GPL-2"  
SLOT="0"  
KEYWORDS="x86 ppc alpha amd64 ppc64 sparc"  
IUSE=""  
  
DEPEND="virtual/libc"  
  
src_compile() {  
    emake FLAGS="${CFLAGS}" || die "compile problem"  
}  
  
src_install() {  
    dobin beep  
    # do we really have to set this suid by default? -solar  
    fperms 4711 /usr/bin/beep  
    doman beep.1.gz  
    dodoc CHANGELOG CREDITS README  
}
```

# // PORTAGE

## SE QUEJA...



### **// \* MASKED \* //**

Es posible que queramos instalar un paquete que no esté disponible para nuestra arquitectura y recibamos un error de enmascaramiento.

- **-arch keyword** implica que la aplicación no está probada lo suficiente para ser parte de la rama estable.
- **-arch keyword** o **-\* keyword** implica que la aplicación no funciona en su arquitectura.
- **missing keyword** implica que la aplicación aún no ha sido probada en su arquitectura. Podemos solicitar al equipo encargado de portar arquitecturas que pruebe el paquete o probarlo nosotros mismos y reportar la experiencia en bugzilla.
- **package.mask** implica que el paquete se ha encontrado corrupto, inestable o peor y ha sido marcado deliberadamente para que no se use. `/usr/portage/profiles/package.mask`
- **profile** implica que el paquete no está disponible para nuestro perfil. La aplicación podría corromper el sistema si se instala o no es compatible con el perfil que estamos usando.

# // PORTAGE

SE QUEJA...



```
hersye@kotoko# emerge -av =sys-apps/portage-2.1.2_rc3-r6
```

These are the packages that I would merge, in order:

Calculating dependencies

!!! All ebuilds that could satisfy "`=sys-apps/portage-2.1.2_rc3-r6`" have been masked.

!!! One of the following masked packages is required to complete your request:

- `sys-apps/portage-2.1.2_rc3-r6` (masked by: `~amd64` keyword)

For more information, see MASKED PACKAGES section in the emerge man page or refer to the Gentoo Handbook.

# // PORTAGE

## SE QUEJA...



### // \* **BLOCKED** \* //

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.

Con `--pretend`:

```
[blocks B ] gnome-base/bonobo-activation (from pkg gnome-base/libbonobo-2.4.0)
```

Sin `--pretend`:

```
!!! Error: the gnome-base/bonobo-activation package conflicts with another package.
```

```
!!! both can't be installed on the same system together.
```

```
!!! Please use 'emerge --pretend' to determine blockers.
```

//

# CURIOSIDADES

¡ANDA YA!



```
hersye@kotoko# emerge -av shoutcast-server-bin
```

These are the packages that I would merge, in order:

Calculating dependencies ...done!

```
[ebuild N F ] media-sound/shoutcast-server-bin-1.9.7 57 kB
```

Total size of downloads: **57 kB**

Do you want me to merge these packages? [Yes/No]

```
>>> emerge (1 of 1) media-sound/shoutcast-server-bin-1.9.7 to /  
!!! shoutcast-1-9-7-linux-glibc6.tar.gz not found in /usr/portage/distfiles
```

```
!!! media-sound/shoutcast-server-bin-1.9.7 has fetch restriction turned on.  
!!! This probably means that this ebuild's files must be downloaded  
!!! manually. See the comments in the ebuild for more information.
```

- \* Please download shoutcast-1-9-7-linux-glibc6.tar.gz from:
- \* <http://www.shoutcast.com/download/license.phtml>
- \* and move it to /usr/portage/distfiles



# **CURIOSIDADES**

**¡DE SABORES!**



## ***/\* DISTROS BASADAS EN GENTOO LINUX \*/***

- **Flash Linux** -> Distribución Linux de 256Mb con entorno gráfico para instalar en PEN Drives.
  - <http://flashlinux.org.uk/>
- **GentooX** -> Distribución Linux para instalar en consolas XBOX.
  - <http://gentoox.shallax.com/>
- **Knopperdisk** -> Distribución Linux especial para instalar en USB PEN Drives o Floppy Disks. Destinada a tareas de recuperación. No dispone de entorno gráfico.
  - <http://knopperdisk.knopper.tk/>
- **Kororaa / Gororaa** -> Distribución Linux en formato LiveCD para probar XGL o AIGLX.
  - <http://kororaa.org/>
- **Pentoo** -> Distribución Linux para instalar en USB PEN Drive con Enlightenment y con gran cantidad de herramientas para pen testings y seguridad.
  - <http://www.pentoo.ch/-PENTOO-.html>
- **Ututo**
- **Vidalinux VLOS**

// **THE END**

*¡GRACIAS!*



**gentoo linux**

**# emerge -verbose -nocolor questions**