

# USERS

TU PUERTA DE ACCESO AL MUNDO DIGITAL

Apto para  
todas las  
DISTROS!

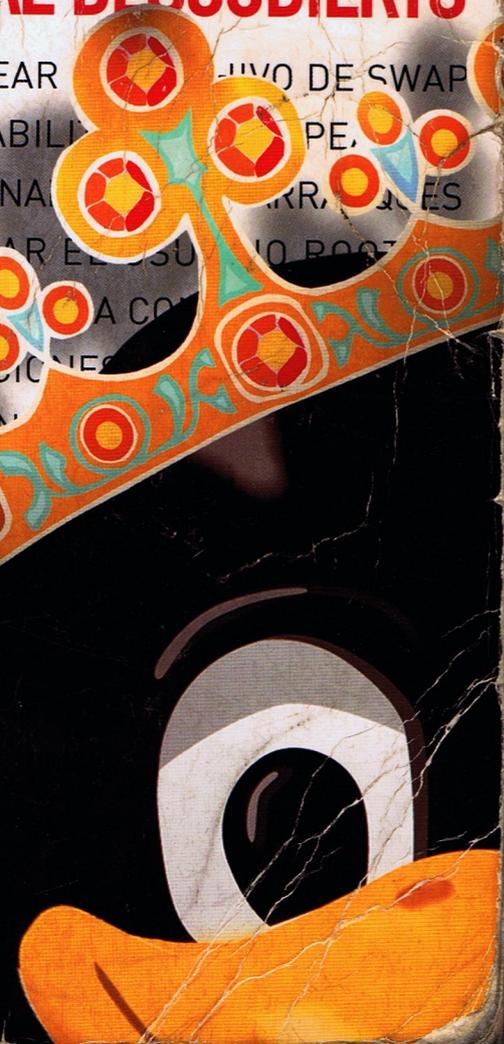
# 202 SECRETOS DE LINUX

## LOS TESOROS OCULTOS, AL DESCUBIERTO

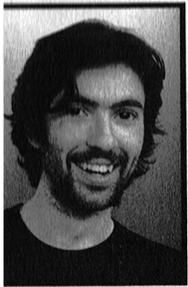
EVITAR BORRADOS ACCIDENTALES | CREAR ARCHIVO DE SWAP  
ARCHIVOS DE CONFIGURACIÓN | DESHABILITAR EL TIPO DE  
BORRAR ARCHIVOS SEGÚN UNA DETERMINADA FECHA | BORRAR ARCHIVOS  
PERSONALIZADOS CON LILO | HABILITAR EL USUARIO ROOT EN  
UBUNTU | ESCUCHAR ARCHIVOS MP3 DE UN DISCO DURA CON  
ARCHIVOS GRANDES / INSTALAR CASI CUALQUIER SISTEMA  
| CREAR IMÁGENES ISO | MEDIR EL RENDIMIENTO DE  
EQUIPO | INICIAR LINUX SIN PASSWORD | PASOS PARA  
| CONFIGURAR IPTABLES | ACCEDER A LINUX EN  
EQUIPOS CON POCOS RECURSOS | EVITAR EL BOMBARDEO  
CREAR NUESTRA PROPIA DISTRIBUCIÓN DE LINUX | EL  
MAKE | AUMENTAR LA SEGURIDAD EN EL SERVIDOR  
TODO SOBRE NETCAT | ESCANEAR EL SISTEMA  
| OTKITS | TRANSFERIR ARCHIVOS

MANUALES USERS MANUALES USERS  
MUNDO ARENA

## ¡DOMINIO DEFINITIVO DEL PINGÜINO!



# Héctor Facundo Arena



Consultor de temas relativos al sistema operativo GNU/Linux y otras aplicaciones de Software Libre desde 1998. A lo largo de su carrera editó diez libros, lideró varios proyectos de desarrollo relacionados con el sistema y sirvió de consultor a usuarios, empresas y organismos gubernamentales.

Es el principal editor de la primera revista de GNU/Linux de Argentina (**Linux USERS**) y posee su propio sitio de noticias tecnológicas en <http://alternaria.blogspot.com>. Allí publica semanalmente un **podcast** y un programa de TV por Internet sobre tecnología y vida digital.

Además, es director de su propio centro de capacitación en tecnologías de Software Libre (**Linux A Distancia**) y ha dictado decenas de cursos y seminarios en diferentes puntos del país.

En su tiempo libre gusta de la investigación de las tecnologías alternativas, como el sistema operativo Mac OS X y las computadoras Atari.

Aparte de las computadoras, tiene otras pasiones como la música (es autor de un libro de producción musical publicado por esta misma editorial), el Karate-DO y otros deportes. Puede ser contactado en su dirección de correo electrónico, [hfarena@gmail.com](mailto:hfarena@gmail.com).

## Agradecimientos y dedicatoria

A todos los colaboradores de la revista Linux USERS por haber compartido sus conocimientos y ayudar a la existencia de este libro.

A todos los lectores que nos han enviado sus trucos.

A Daniel Benchimol y Diego Spaciuk por la paciencia que me han tenido durante el proceso de redacción.

A mis amigos y a mi familia, por haberme soportado durante algunos meses sumido en la más aislada concentración seleccionando el contenido de este libro.

## PROLOGO

Hace aproximadamente siete años que trabajo en la generación y selección de contenidos (notas para revistas, libros, sitios web) relacionados con GNU/Linux y su conjunto de herramientas de software libre. Un día me di cuenta de que tenía un montón de material interesante y se me ocurrió la idea de hacer una recopilación de todo ese material. Cuando mi editor me propuso escribir un libro de trucos y soluciones para los usuarios de Linux, consideré que ésa era una gran oportunidad para llevar a cabo ese proyecto.

Por mi trabajo como generador de contenidos y profesor en cursos, tengo una idea bastante clara de cuáles son los puntos más críticos para los usuarios de Linux. Preguntas del tipo ¿cómo habilito la clave de root en Ubuntu?, ¿cómo extiendo la memoria swap? o ¿hay alguna forma de imprimir lo que me aparece en pantalla? llegan diariamente a mi casilla de correo electrónico. Es por eso que en este libro intenté reunir las soluciones a los problemas más populares que tienen los usuarios de Linux hoy en día. Además, está condimentado con experiencias propias, sugerencias y tutoriales para que los usuarios mejoren exponencialmente su experiencia de usabilidad del sistema operativo GNU/Linux.

Quiero destacar que, si bien el contenido de este libro está compuesto en gran parte por material de mi autoría, también he seleccionado los mejores trucos y soluciones de los colaboradores de la revista que edito hace varios años (Linux USERS) y de los lectores que tan gentilmente nos han enviado sus trucos. Por eso quiero agradecer a todos los colaboradores y usuarios que en algún momento han compartido sus soluciones y de esa forma ayudaron a dar lugar a la existencia de este libro. ¡Gracias a todos por compartir sus conocimientos!

Finalmente, permítanme hacerles una recomendación. Hay dos formas de leer este libro. En primer lugar, está la forma tradicional, comenzando por el primer capítulo y avanzando hasta el último, como lo haríamos con cualquier libro convencional. La segunda, es abrirlo en cualquier página y ver si la solución o el truco allí explicado nos resulta de utilidad. Me gusta pensar en los libros de trucos como grandes tesoros de soluciones, y la sensación de descubrir algo nuevo cada vez que abro una página al azar es realmente emocionante. Ténganlo en cuenta.

*Héctor Facundo Arena*

# L LIBRO DE UN VISTAZO

este libro encontraremos trucos para las distintas distribuciones de Linux. Están organizados según el tipo de tarea que nos permiten realizar o por su utilidad específica. En algunos casos, incluye un breve tutorial al comienzo del capítulo para que los usuarios primero se familiaricen con el lenguaje y luego puedan llevar a cabo los trucos. Para finalizar, veremos la forma de vencer a otros de migrar a Linux y una serie de sitios y aplicaciones recomendados.

## Capítulo 1

### TRUCOS PARA LA CONSOLA

Este primer capítulo del libro descubriremos aquellas funcionalidades ocultas en los comandos de la consola que nos permitirán sacarle el máximo provecho al sistema operativo. Se trata de una selección de trucos ideales para comenzar.

## Capítulo 2

### TRUCOS AVANZADOS PARA LA CONSOLA

Este capítulo es una selección de trucos para la consola de Linux que requieren un poco más de experiencia (y atención) por parte de los usuarios, pero que resultan muy útiles.

## Capítulo 3

### TRUCOS PARA LA CONSOLA PARA EL ADMINISTRADOR

Este capítulo incluye una selección de los mejores y más útiles trucos especialmente orientados a los administradores de sistemas GNU/Linux en entornos de red.

## Capítulo 4

### TRUCOS PARA KDE Y GNOME

En este capítulo encontraremos una completa selección de trucos especiales para mejorar el aspecto visual y para optimizar el rendimiento de los dos entornos de escritorios más populares del mundo de GNU/Linux.

## Capítulo 5

### TRUCOS PARA LAS APLICACIONES

Selección de trucos para sacarles el máximo provecho a las aplicaciones más populares de Linux como Firefox, OpenOffice, Amarok, Thunderbird, etcétera.

## Capítulo 6

### TRUCOS PARA VI Y EMACS

Se trata de los dos editores de texto más populares en el ámbito UNIX. En este capítulo incluimos dos tutoriales para aprender a utilizarlos rápidamente, y una completa selección de trucos y sugerencias para dominar cada uno de ellos.

## Capítulo 7

### OPTIMIZAR EL RENDIMIENTO Y LA SEGURIDAD

Consejos y sugerencias para optimizar el rendimiento y el funcionamiento del sistema operativo. Además, en este capítulo incluimos soluciones (como un tutorial de iptables) para aumentar la seguridad del sistema.

## Capítulo 8

### SCRIPTS EN BASH PARA AUMENTAR LA PRODUCTIVIDAD

Bash es el lenguaje de scripting por excelencia. Por eso, este capítulo incluye un tutorial rápido para dar nuestros primeros pasos en él y una completa selección de

trucos y scripts para automatizar al máximo las tareas más tediosas que debemos realizar.

## Capítulo 9

### SCRIPTS EN PERL

PERL es el otro lenguaje de scripting elegido por los administradores. Este capítulo incluye un tutorial de primeros pasos y una completa selección de sugerencias para generar mejores scripts de administración.

## Capítulo 10

### CONSEJOS PARA LA MIGRACIÓN A LINUX

Selección de consejos, trucos y soluciones paso a paso para llevar a cabo un proceso exitoso de migración a Linux, tanto en el hogar como en la empresa.

## Capítulo 11

### DISTRIBUCIONES

Como hay muchas distribuciones disponibles, en este capítulo haremos un repaso de las más conocidas o recomendadas según el tipo

de uso que les quiera dar el usuario. Así, veremos distros especiales para el hogar o la oficina, para los servidores y para llevar en un CD o pendrive.

## Apéndice A

### SITIOS WEB DE INTERÉS

En este apéndice conoceremos las direcciones de muchos sitios recomendados para conocer más sobre Linux y para informarse sobre las novedades de este sistema operativo, además de aquellos en los que podemos realizar descargas.

## Apéndice B

### APLICACIONES QUE NO TODOS CONOCEN

Aquí veremos una serie de aplicaciones que, aunque no sean muy conocidas, resultan de mucha utilidad para realizar tareas cotidianas y para simplificar aquellas que consideramos muy complejas o que pensamos que no se pueden hacer.



## INFORMACIÓN COMPLEMENTARIA

A lo largo de este manual encontrará una serie de recuadros que le brindarán información complementaria: curiosidades, trucos, ideas y consejos sobre los temas tratados. Cada recuadro está identificado con uno de los siguientes iconos:



CURIOSIDADES  
E IDEAS



ATENCIÓN



DATOS ÚTILES Y  
NOVEDADES



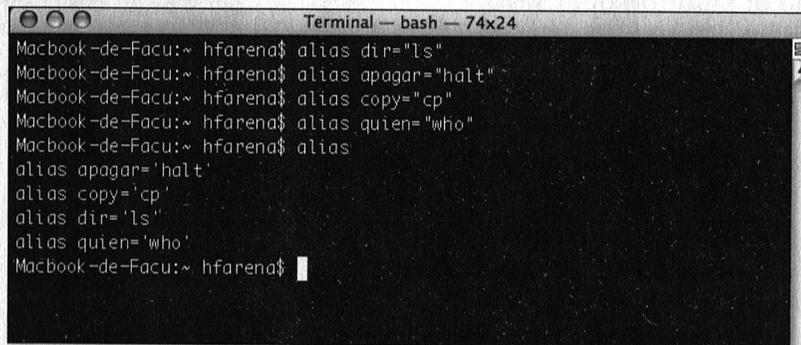
SITIOS WEB

## Desplazamiento por páginas en la terminal

Toda la información que aparece en la pantalla de nuestra terminal se almacena en un **buffer** que se puede consultar como **historial** en cualquier momento. Para hacerlo, se debe usar la combinación de teclas **SHIFT+AVPAG** o **SHIFT+RETRPAG** para navegar por todas las líneas que han aparecido en la terminal como si se tratara de un archivo de texto. Luego, con tan sólo presionar la tecla **ENTER**, estaremos de nuevo en la línea de comandos del sistema.

## Evitar borrados accidentales

El comando **rm** (borrar archivos y carpetas) es un comando bastante insolente que actúa por sí solo sin pedirle confirmación al usuario sobre el crimen que va a cometer. Para evitar inconvenientes, muchos usuarios lo utilizan en conjunto con el parámetro **-i**, que se encarga de **pedir confirmación** antes de llevar a cabo la tarea de eliminación del archivo o la carpeta en cuestión.



```
Terminal — bash — 74x24
Macbook-de-Facu:~ hfarena$ alias dir="ls"
Macbook-de-Facu:~ hfarena$ alias apagar="halt"
Macbook-de-Facu:~ hfarena$ alias copy="cp"
Macbook-de-Facu:~ hfarena$ alias quien="who"
Macbook-de-Facu:~ hfarena$ alias
alias apagar='halt'
alias copy='cp'
alias dir='ls'
alias quien='who'
Macbook-de-Facu:~ hfarena$
```

Figura 1. El comando **alias** nos permite crear nuevos nombres para los comandos

Es posible definir que cada vez que se ingrese el comando **rm** en realidad se ejecute **rm -i**. ¿Cómo se logra esto? Por medio de un alias como el siguiente:

```
alias rm="rm -i"
```

Este alias sólo funcionará mientras la sesión del usuario esté abierta y cuando se cierre la sesión, el alias será eliminado. Para hacer que se cargue cada vez que inicia el sistema, sólo debemos agregar la línea al archivo **.bash\_profile** (o **.bashrc**) ubicada en nuestro directorio personal.

## Corrección ortográfica de archivos de texto desde la consola

El mundo de los comandos de Linux no tiene límites, y una de las tantas aplicaciones disponibles para los usuarios de este entorno operativo es el comando **ispell**, que permite corregir la ortografía de los archivos de texto (Figura 2). Su forma de utilización es la siguiente:

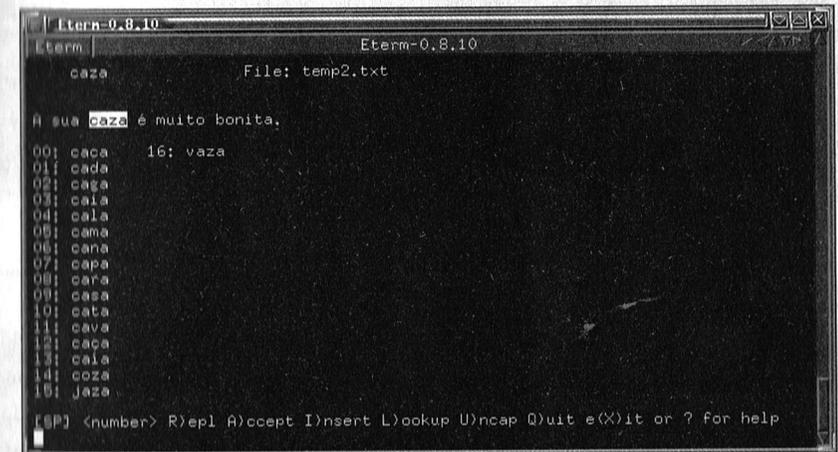
```
ispell -d spanish [archivo]
```

Además, es útil la posibilidad de cambiar el parámetro **spanish** por cualquier otro idioma en el que se quiera corregir el archivo de texto. Una vez cargado, se presentarán en pantalla las opciones de corrección (tal como lo hace, por ejemplo, Microsoft Word) y con sólo indicar el número que identifica a una de las palabras sugeridas podremos corregir la palabra que se encuentra escrita de forma incorrecta.

## Contador de palabras en archivos de texto

Otro de los comandos que nos facilita la tarea de trabajar con archivos de texto en la consola es el comando **wc**, que nos ofrece estadísticas sobre su contenido. Veamos un ejemplo:

```
Macbook-de-Facu:~ hfarena$ wc .bash-history
500  988  8663 .bash_history
Macbook-de-Facu:~ hfarena$
```



```
Eterm-0.8.10
Eterm-0.8.10
caza File: temp2.txt

A sua caza é muito bonita.
00: caca 16: vaza
01: cada
02: caga
03: caia
04: cala
05: cama
06: cana
07: capa
08: cara
09: casa
10: cata
11: cave
12: caca
13: caie
14: coze
15: Jaze

[SP] <number> R)epl A)cept I)nsert L)ookup U)ncap Q)uit e(X)it or ? for help
```

Figura 2. En Linux contamos con **ispell**, una herramienta para la consola

## BUSCAR ARCHIVOS RÁPIDO

Muchas veces, en el uso de nuestro sistema nos encontramos con la necesidad de buscar algún archivo alojado en un directorio específico que no tenemos la más mínima idea de cuál es. Para esto, el comando **find** viene a nuestra ayuda. Utilizarlo es muy simple y sólo basta con hacerlo como en el siguiente ejemplo:

```
find / -name "nombre"
```

El símbolo **/** hace referencia al directorio inicial de la búsqueda, que en el caso del ejemplo es la raíz del sistema (no sólo buscará en la raíz sino en todos los directorios allí contenidos, a forma de árbol). Finalmente, el parámetro **-name** nos permite definir el nombre del archivo a buscar.

El primer valor que ofrece corresponde a la cantidad de líneas que hay en el archivo de texto. El segundo valor indica la cantidad de palabras y final-

mente, el tercer valor corresponde a la cantidad de caracteres.

## Cómo formatear disquetes rápidamente

GNU/Linux nos ofrece la facilidad de dar formato a disquetes para diferentes sistemas operativos. Por ejemplo, para dar formato MS-DOS a un disquete (lo que nos permitirá utilizarlo en un equipo con Microsoft Windows), se puede utilizar del siguiente modo el comando **mformat** que se encuentra en el paquete **mtools** (incluido en casi todas las distribuciones):

### mformat a:

Formatear un disquete para ser utilizado en GNU/Linux requiere de dos pasos:

```
fdformat /dev/fd0H1440
mkfs.ext2 /dev/fd0
```

El primer comando se encarga de formatear el disquete A (**/dev/fd0**) en 1.44MB (el tamaño estándar). Por último, el segundo comando se encarga de crear un sistema de archivos **EXT2** (el estándar para disquetes) en el disco recientemente formateado.

## III DEFINIR LA REGLA EN VI

El editor de textos **VI** es muy completo, aunque a veces su interfaz es muy austera y no podemos obtener datos fácilmente, como el número de línea en el que estamos trabajando. Para modificar esto, sólo debemos modificar el archivo **.vimrc** y agregar al final la línea **set ruler**. Esto mostrará la regla la próxima vez que se abra el editor.



Formatear disquetes y eliminar procesos son dos tareas que generalmente requieren de permisos de administrador (root) para funcionar.

## Cómo matar procesos que no responden

En el uso cotidiano de GNU/Linux se presenta la particularidad de tener algún proceso corriendo al cual necesitamos **matar** (cerrar, eliminar, etcétera). Esto se puede llevar a cabo de varias formas. En nuestro ejemplo, la aplicación Mozilla está colgada y queremos cerrarla porque no responde. Para esto procedemos de la siguiente manera: vamos a una consola y buscamos su entrada correspondiente en el registro de procesos.

```
$ ps -aux|grep mozilla
root      16914  1.0 16.7 146836
40264 ?        S1   Dec06
15:32 /mozilla-1.7.2/mozilla
bin
root      20241  0.0  0.3  1916
736 pts/1    S+   16:24
0:00 grep mozilla
```

En el ejemplo nos encontramos con dos líneas: una que nos devuelve el valor de proceso con el cual está corriendo y la otra que nos muestra la consulta que ingresamos. Para poder cerrar Mozilla debemos hacer uso del comando **kill** seguido del número **PID** o **Process Id** (en el caso del ejemplo, **16914**):

```
# kill -9 16914
```

Otra opción es utilizar el comando **killall** para matar a todos los procesos que tengan un determinado nombre:

```
# killall mozilla
```

Existe una herramienta llamada **xkill** que nos permite matar una aplicación con sólo hacer clic sobre su ventana en el modo gráfico. Podemos ejecutarla desde el panel de ejecución de comandos (con la combinación de teclas **ALT+F2** en KDE o GNOME).

## Otra alternativa para buscar archivos

Buscar archivos por su nombre es una de las tareas que con más frecuencia se lleva a cabo en un sistema Linux. Y como siempre, tenemos varias opciones para hacerlo. En Linux contamos con un comando poco utilizado llamado **locate**, que nos permite consultar en una base de datos con los nombres y las ubicaciones de todos los archivos de nuestro sistema. Su uso es muy simple: con sólo ingresar **locate** y el nombre del archivo, el sistema comenzará a buscar.

## Cómo cambiar la pantalla de bienvenida en modo texto

Cuando nos registramos en una terminal virtual de texto, un mensaje de bienvenida aparece frente a nosotros antes del **login** y luego, una vez que

```
top - 13:59:13 up 0 min, 1 user, load average: 0.34, 0.11, 0.04
Tasks: 33 total, 1 running, 32 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0% us, 0.4% sy, 0.0% ni, 90.9% id, 0.0% wa, 0.0% hi, 8.7% si
Mem: 252964k total, 21812k used, 231152k free, 2016k buffers
Swap: 522104k total, 0k used, 522104k free, 10768k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2341	root	16	0	2016	1024	812	R	0.3	0.4	0:00.02	top
1	root	16	0	1524	540	476	S	0.0	0.2	0:00.63	init
2	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	events/0
5	root	20	-5	0	0	0	S	0.0	0.0	0:00.02	khelper
6	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
8	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kblockd/0
11	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	khubd
13	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kseriod
124	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pdflush
125	root	15	0	0	0	0	S	0.0	0.0	0:00.00	pdflush
126	root	25	0	0	0	0	S	0.0	0.0	0:00.00	kswapd0
127	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
129	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	jfs10
130	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	jfsCommit
131	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	jfsSync
132	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	xfslogd/0

Figura 3. Top es una herramienta que nos permite monitorear el estado de los procesos desde la consola. Si presionamos la tecla **K** mientras están en ejecución, podemos eliminar un proceso determinado.

nos registramos en el sistema con usuario y contraseña, otro mensaje de bienvenida puede aparecer. Esto en realidad depende de la distribución, ya que algunas no muestran un mensaje en absoluto. Pero las herramientas para definir estos dos comunicados están presentes en todas las distribuciones.

Ahora bien, ¿para qué sirve esto? Como administradores de sistemas, el mensaje que podemos definir antes del **login** nos servirá para brindarles **comunicados colectivos** a todos los usuarios que se conecten al sistema. Por su parte, el mensaje que podemos definir para después del **login** nos permitirá brindarles comunicados sólo a los usuarios que entraron correctamente al sistema (Figura 4).

Para ambos mensajes tenemos archivos disponibles. Para el mensaje general tendremos que editar el archivo **/etc/issue**, y para el mensaje exclusivo para los usuarios tenemos a nuestra disposición el archivo **/etc/motd** (**Message of the day**). Con sólo editar esos archivos con nuestro editor de texto favorito podremos modificarlos.

```
Welcome to GoboLinux 013.

System.....: Linux 2.6.17.11-Gobo i686
Compilation...: #12 SMP Tue Oct 24 12:38:45 BRST 2006
Processor.....: Genuine Intel(R) CPU 1400 @ 1.83GHz
Clock.....: 2086.810 MHz
Bogomips...: 2999.29
Memory.....: 252964k
Terminal.....: tty2

GoboLinux login: root
Password:
Bienvenido al sistema!
root@GoboLinux ~1_
```

**Figura 4.** Podemos utilizar el archivo **motd** para dar la bienvenida al sistema, por ejemplo, en **GoboLinux**.

## LA LÍNEA DE COMANDOS

Para definir la línea de comandos completa podemos usar la variable de entorno **PS1** del siguiente modo: **export PS1="[u@\h \W]\\\$ "**. Pero, si tenemos en cuenta que **u** es el nombre de usuario, **h** es el nombre del equipo y **W** es el directorio de trabajo, podemos armar la línea de comandos como más nos guste. Luego podemos ingresar esta línea al final del archivo **.bashrc** dentro de nuestro directorio personal para que los cambios se hagan permanentes.

## Tuberías: conexión de flujos de procesos

GNU/Linux provee un método muy sofisticado para la comunicación de datos entre procesos denominado **pipes** (tuberías). Estas tuberías permiten tomar la salida de un proceso y enviársela a otro como entrada. Para entenderlo mejor, veamos un ejemplo de esto en la siguiente línea:

```
ls /usr/doc | more
```

Nótese el símbolo **|** entre los dos comandos. El resultado de esta operación será la devolución del directorio **/usr/doc** con pausas en cada pantalla. El **shell** analiza expresión por expresión en toda la línea de comandos. La primera expresión es el comando **ls** y la segunda es el parámetro **/usr/doc**. La tercera expresión con la que se encuentra es el símbolo **|** que le indica que deberá crear una tubería entre el comando **ls** y el siguiente (**more**), que se encarga de hacer las pausas en la pantalla.

Otro buen ejemplo para entender este procedimiento es el uso de los comandos **cat** y **grep** combinados. El comando **cat** visualiza un archivo en pantalla y el comando **grep** busca coincidencias entre cadenas. De esta manera, si queremos buscar la palabra **Other** dentro del archivo **/etc/lynx.cfg**, sólo habrá que tipear:

```
cat /etc/lynx.cfg | grep "Other"
```

Y la salida será:

```
# lynx. Other settings are
  ignored. Note: INCLUDE is
  also treated as a
# you can connect to reliably.
  Otherwise users will become
  confused and think
```

```
# Other codes not listed above
  may be available for
  additional keys,
# LYMessages_en.h Otherwise, it
  will be NOVICE_LINE_TWO.
# A double percent yields a
  literal percent on output.
  Other characters
# Other source-view related
  options:
```

La salida del comando **cat** sirvió como entrada del comando **grep** para buscar coincidencias de la palabra **Other** dentro de ese archivo.

## Cómo instalar paquetes RPM

**RPM** es el formato de empaquetamiento creado por **Red Hat**, y actualmente se utiliza en distribuciones como Fedora, SuSE y Mandriva. Siempre hay que tener en cuenta lo siguiente antes de instalar un paquete en formato RPM:

- Que la distribución que usemos tenga **soporte** para ese paquete (Mandriva, Red Hat y Fedora lo tienen por ejemplo).
- Que los **componentes** necesarios para ese programa (librerías u otros programas) estén instalados. Si no lo están

## CÓMO DESINSTALAR PROGRAMAS RPM

Para desinstalar un programa que instalamos en RPM, simplemente debemos usar el comando **rpm -e programa**, donde **programa** es la primera palabra que aparece en el nombre del paquete (para **abiword-2.6.0-x86**, el comando sería **rpm -e abiword**).

entonces deberemos empezar por instalar esos componentes.

- Que el archivo RPM que vamos a instalar sea para la **versión** de la distribución que estamos usando.

Una vez que todo esto se cumple, podemos proceder a instalar el archivo RPM. En el modo gráfico, con tan sólo hacer clic sobre él, KDE nos abrirá un programa de instalación que hará todo el trabajo de manera automática. En el modo texto, podemos instalarlo con el siguiente comando:

```
rpm -i archivo.rpm
```

Si el programa que hemos descargado viene en varios archivos RPM y estamos seguros de que en el directorio de paquetes está todo lo necesario para que el programa funcione, entonces podemos realizar una **instalación forzosa** de todos los paquetes de la siguiente manera:

```
rpm -i --force --nodeps *.rpm
```

Por su parte, **Mandriva** tiene un sistema llamado **urpmi** que permite instalar aplicaciones desde los servidores repositorios de archivos muy fácilmente. Para utilizarlo, debemos configurar nuevos medios de paquetes porque por defecto el sistema sólo buscará paque-

tes de aplicaciones en los CDs de la distribución. Podemos acceder a un listado de repositorios interactivo en el sitio <http://easyurpmi.zarb.org> (Figura 5). Allí se nos consultará por la plataforma, los tipos de paquetes que queremos y la versión de Mandriva que tenemos. Luego nos dará un comando para que ejecutemos en la consola que tiene la siguiente estructura:

```
urpmi.addmedia [tipo_de_paquetes]
[direccion_fuente]
```

Cuando seleccionemos los tipos de paquetes que queremos, tendremos las siguientes alternativas:

- **Main:** éstos son los paquetes oficiales de la distribución.
- **Update:** paquetes de actualización (también oficiales).
- **Contrib:** son programas no oficiales de la distribución y que se encuentran allí como colaboración.

Podemos agregar repositorios a los tres tipos de paquetes para tener más alternativas a la hora de instalar un programa si no sabemos en cuál de ellos se encuentra. Luego, para instalar paquetes de manera automática, usamos el comando **urpmi** del siguiente modo:

```
urpmi [paquete.rpm]
```

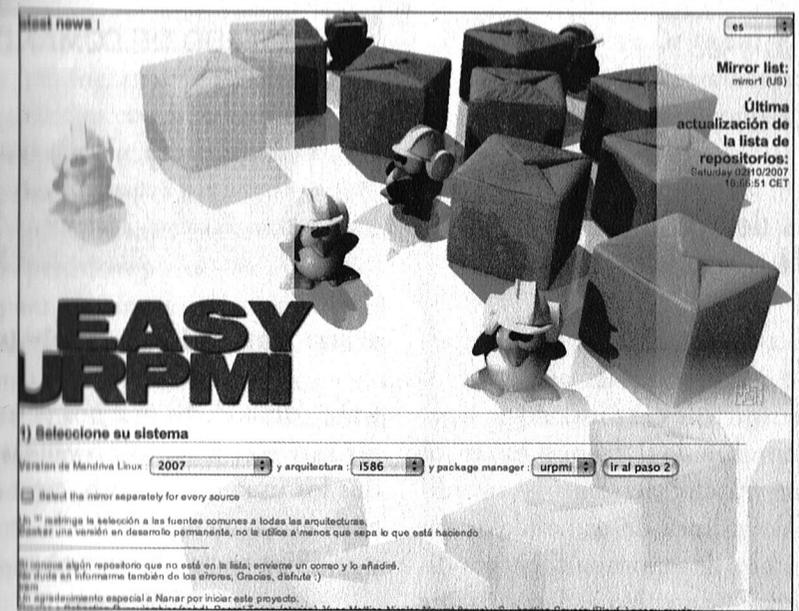


Figura 5. En <http://easyurpmi.zarb.org> podemos encontrar un listado de los repositorios del sistema urpmi.

El sistema se conectará al servidor de archivos, bajará todas las dependencias necesarias y se encargará de instalar el paquete solicitado.

## Cómo instalar paquetes DEB

En las distribuciones **Debian**, **Ubuntu** y sus derivados, tenemos varias herramientas de instalación de paquetes. La más simple de ellas es **dpkg**, que se utiliza de manera similar al RPM de Red Hat:

```
dpkg -i archivo.deb
```

Esto se encargará de manera automática de descomprimir el paquete y copiar cada cosa a su lugar. Para desinstalar, simplemente debemos tipear:

```
dpkg -i nombre_paquete
```

Además, los usuarios de **Debian** cuentan con un comando **apt-get** que funciona de manera similar al **urpmi** que vimos antes. Para instalar aplicaciones se usa del siguiente modo: **apt-get install [paquete]**. Pero antes, siempre es conveniente actualizar la base de datos de paquetes disponibles con el comando **apt-get update**.

También existe una aplicación llamada **dselect** para el modo texto que nos permite navegar por toda la base de aplicaciones y seleccionar las que queramos instalar, desinstalar o actualizar (Figura 6). Lo primero que debemos hacer cuando trabajamos con paquetes en Debian y sus derivados es acceder a la opción **Access** y definir las **fuentes de paquetes** (pueden ser los CDs de Debian, un sitio web o servidores FTP). Luego, debemos seleccionar la opción **Update** para descargar la lista de aplicaciones ac-

## III DESINSTALAR PAQUETES

Para borrar aplicaciones debemos utilizar el comando **urpme** seguido del nombre del paquete sin la extensión. Por ejemplo: **urpme kdebase**.

tualizada y luego sí, proceder a la selección **Select** para seleccionar las aplicaciones que queremos instalar.

## Archivos de configuración de BASH

BASH (Bourne Again SHell) lee un archivo de configuración ubicado en el directorio personal de cada usuario para generar un entorno de trabajo. ¿Qué es un entorno de trabajo? Sencillo: una configuración de la línea de comandos y un set de variables de entorno útiles para dicho usuario. Este archivo se llama **.bashrc** (es un archivo oculto ya que comienza con un punto) y como mencionamos antes está ubicado en el directorio personal de cada usuario. Es posible editarlo con cualquier editor de textos y crear o eliminar configuraciones de variables de entorno según lo deseado. Para definir variables de entorno nuevas, usamos:

```
bob@
dselect - inspection of package states (avail., priority)
EIDM Pri Section Package Inst.ver Avail.ver Description
- All packages -
- Up to date installed packages -
- Up-to-date Required packages -
- Up-to-date Required packages in section base -
*** Req base base-files 3.0.8 3.0.8 Debian base system miscellar
*** Req base base-passwd 3.5.1 3.5.1 Debian base system master pa
*** Req base bash 2.05b-7 2.05b-7 The GNU Bourne Again SHell
*** Req base bsdtails 2.11z-1 2.11z-1 Basic utilities from 4.4BSD-
*** Req base coreutils 4.5.10-1 4.5.10-1 The GNU core utilities
*** Req base debianutils 2.4.1 2.4.1 Miscellaneous utilities spec
*** Req base diff 2.7-29 2.7-29 File comparison utilities
*** Req base dpkg 1.10.10 1.10.10 Package maintenance system f
*** Req base dselect 1.10.10 1.10.10 a user tool to manage Debian
*** Req base e2fsprogs 1.32-2 1.32-2 The EXT2 file system utilit
*** Req base findutils 4.1.7-2.1 4.1.7-2.1 utilities for finding files-
debianutils installed; install (was: install). Required
debianutils - Miscellaneous utilities specific to Debian
This package provides a number of small utilities which are used primarily by the
scripts of Debian packages, although you may use them directly.
The specific utilities included are: installkernel mkboot mktemp run-parts savelog
sensible-browser sensible-editor sensible-pager tempfile which.
```

Figura 6. Dselect es un gestor de paquetes para Debian que está incluido en la instalación básica.

## LISTADO DE COMANDOS

Para obtener un listado completo de todos los comandos presentes en la distribución sólo tenemos que pulsar dos veces la tecla **TAB** y a continuación la letra **Y**.

**export [nombredelavariabile]= [valor]**

Existe un archivo que nos permite definir variables de entorno válidas para todos los usuarios de bash, cuya estructura es exactamente igual a la recién vista y se ubica en **/etc/bash.bashrc**.

## Cómo cambiar los permisos de los archivos sin recordar el significado de los bits

Seguramente nos ha pasado que nos olvidamos la forma de armar el parámetro de permisos del comando **chmod**:

777, 666, 776 ¡demasiado difícil! Aquí tenemos una forma más fácil.

Las posibilidades que tiene el usuario dueño o el grupo dueño de realizar diferentes acciones con el archivo son tres: **Lectura (r)**, **Escritura (w)** y **Ejecución (x)**. Estos permisos pueden ser personalizados para el usuario dueño, para el grupo dueño y para los otros usuarios del sistema. Para cambiar los permisos de un archivo sin usar la nomenclatura de bits, tenemos que utilizar el comando **chmod** del siguiente modo:

**chmod [quien][+/-][permiso] [usuario]**

El primer parámetro (quien) nos permite definir a quién vamos a modificarle los permisos de acceso. El usuario dueño es la letra **u**, el grupo dueño es la letra **g**, los demás usuarios del sistema son la letra **o** y todos los usuarios del sistema (incluidos el usuario dueño y el grupo dueño) son la letra **a**. Luego, con los signos **+** y **=** asignamos o quitamos el permiso definido en el tercer parámetro. Es muy importante que no haya espacios entre los parámetros. Veamos algunos ejemplos para aclarar las cosas:

**chmod u+x archivo.pl**

```
root@Gobolinux ~]# ls -l
total 28
drwxr-xr-x 5 root root 4096 2007-04-02 16:26 Choices
drwxr--r-- 3 root root 4096 2007-04-02 16:28 Desktop
drwxr-xr-x 2 root root 4096 2007-04-02 16:28 Documentos
drwxr-xr-x 2 root root 4096 2007-04-02 16:28 Imagenes
drwxr-xr-x 2 root root 4096 2007-04-01 15:45 lib
drwxr-xr-x 2 root root 4096 2007-04-02 16:26 Trash
-rw-r--r-- 1 root root 2198 2007-04-01 22:39 xorg.conf.new
root@Gobolinux ~]#
```

Figura 7. Podemos visualizar los permisos de los archivos con el comando **ls -l** (es la primera columna).

## PROCESOS CON NOHUP

Para lanzar un proceso y luego poder matar el shell padre, debemos hacerlo de la siguiente manera:

**nohup proceso &**

Ahora, el proceso seguirá funcionando por más que la terminal en la que se encuentra se haya cerrado. La salida del comando podemos verla en el archivo **nohup.out**.

Es importante tener en cuenta que si eliminamos el archivo **nohup.out** mientras el proceso sigue activo, no volverá a crearse.

Mediante la línea de este ejemplo el usuario dueño del archivo podrá ejecutar el archivo **archivo.pl**.

**chmod go-w datos.txt**

Con este comando, el grupo dueño del archivo y los demás usuarios del sistema no podrán escribir el archivo **datos.txt**.

**chmod a+w datos.txt**

Ahora, todos los usuarios del sistema podrán escribir el archivo **datos.txt**.



Tenemos la posibilidad de generar archivos comprimidos desde Midnight Commander (el gestor de archivos clon de Norton Commander) de manera sumamente sencilla. Para hacerlo, sólo hay que seleccionar los archivos que se quieren comprimir, presionar la tecla F2 (que despliega el menú de utilidades) y seleccionar la opción Crear archivo comprimido .tar.gz.

## Cómo generar un archivo comprimido .tar.gz

Para generar este tipo de paquetes se utilizan dos herramientas. La primera de ellas es **tar**, que se encarga de empaquetar muchos archivos en uno solo, pegándolos todos cola con cabeza uno debajo del otro. **Tar** no comprime absolutamente nada, por lo que necesitaremos de otra herramienta para realizar esta tarea: hablamos de **gzip**. Con **gzip** comprimimos el paquete generado con **Tar** para que ocupe menos espacio. Para generar este tipo de paquetes debemos ubicarnos en el directorio que contiene los paquetes que queremos empaquetar. Allí tenemos que tipear el siguiente comando:

```
tar -cvf [archivo.tar] *
```

## CÓMO CAPTURAR EL CONTENIDO DE UNA CONSOLA

Para capturar el contenido de una terminal en GNU/Linux sólo debemos introducir una pequeña sentencia del comando **setterm**, que en realidad lo que hace es mostrar el contenido del archivo **/dev/vcsaN** (donde **N** es el número de la terminal). La línea de comando es la siguiente:

```
# setterm -file [nombre de archivo] -dump N
```

Esto generará un archivo empaquetado, que además podemos comprimir con el siguiente comando:

```
gzip archivo.tar
```

## Cómo deshabilitar el PC-Speaker

En muchas ocasiones (por no decir siempre) es bastante molesto el muy conocido **beep** del PC-Speaker que suena con eventos, con el autocompletado, cuando nos equivocamos de tecla, de comando, y en tantas otras ocasiones. La solución es muy simple. Sólo tendremos que agregarle una línea a nuestro **inicio** o al sistema por medio de un script al **rc.local**, **boot.local**. La línea es:

```
set bell-style none
```

Cuando ingresemos otra vez en el sistema, éste no molestará con el **beep**.

## Cómo instalar el plug-in de Java

Primero debemos descargar la última versión de uno de los componentes

esenciales para todo usuario de Linux que navegue por Internet. Para eso tenemos que dirigirnos a la dirección [www.java.com/en/download/linux\\_manual.jsp](http://www.java.com/en/download/linux_manual.jsp) (Figura 8).

Allí, si hacemos clic en la **versión no RPM**, descargaremos el archivo binario de instalación con un peso que ronda los 18 MB. Una vez descargado, procedemos a copiarlo en un nuevo directorio llamado **/usr/java**. Siempre como usuario **root** tipeamos:

```
# mkdir /usr/java
```

Otorgamos permisos para ejecutarlo:

```
# chmod 777 jre-1_5_0_04-linux-i586.bin
```

Ejecutamos el binario:

```
# ./jre-1_5_0_04-linux-i586.bin
```

Esto va a crear un directorio llamado **jre1.5.0\_04**. Dentro del subdirectorio **/plugin/i386/ns7** encontraremos el plug-in específico para cualquier navegador web. Ahora debemos buscar la ubicación en la que nuestro navegador guarda el plug-in y crear un enlace:

**whereis mozilla**

```
mozilla: /usr/bin/mozilla /usr/lib/mozilla /usr/share/man/man1/mozilla.1.gz
```

Entramos en el directorio:

```
cd /usr/lib/mozilla
ls
libnssckbi.so plugins
```

Y creamos el enlace:

```
ln -s /usr/java/jre1.5.0_04/plugin/i386/ns7/libjavaplugin_oji.so
```

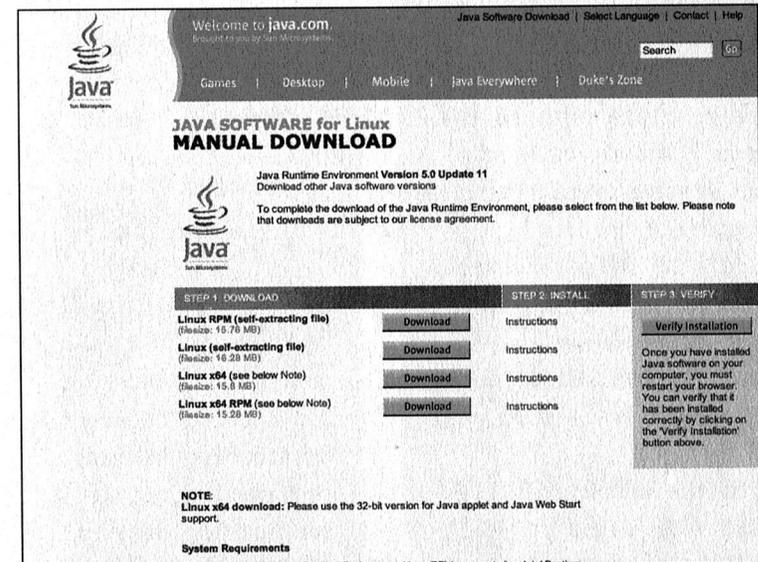


Figura 8. La última versión de Java para Linux se puede descargar de [www.java.com/en/download/linux\\_manual.jsp](http://www.java.com/en/download/linux_manual.jsp).

De esta forma, el plug-in queda instalado. Sólo resta cerrar nuestro navegador y volverlo a abrir para ver su correcto funcionamiento.

## Cómo instalar el soporte de RealPlayer

Para instalarlo, primero debemos descargar la versión para GNU/Linux de [www.real.com/linux?pcode=rn&src=freeplayer\\_partner&opage=freeplayer\\_partner](http://www.real.com/linux?pcode=rn&src=freeplayer_partner&opage=freeplayer_partner) (Figura 9). Una vez que lo hayamos descargado, debemos proceder a asignar permisos totales para **root**:

```
chmod 755 RealPlayer10GOLD.bin
```

Y luego tenemos que ejecutar el binario:

```
./RealPlayer10GOLD.bin
Extracting files for RealPlayer
installation.....
.....

Enter the complete path to the
directory where you want
RealPlayer to be installed. You
must specify the full
pathname of the directory and
have write privileges to
the chosen directory.
Directory: [//local/Paquetes/SH/
RealPlayer]: /usr/local/real

Welcome to the RealPlayer
(10.0.5.756) Setup for UNIX
Setup will help you get
RealPlayer running on your
```

```
computer.
Press [Enter] to continue...
```

Cuando se nos consulte sobre la **ruta destino**, podemos optar por **/usr/local/real** o cualquier otra que nos guste.

```
You have selected the following
RealPlayer configuration:
```

```
Destination:
  /usr/local/real
```

```
Enter [F]inish to begin copying
files, or [P]revious to go
back to the previous prompts:
[F]:
```

Ahora presionamos la **F** para que el sistema comience a copiar todos los archivos, cada uno en su lugar:

```
Copying RealPlayer
files...configure system-wide
symbolic links? [Y/n]:
.....y
enter the prefix for symbolic
links [/usr]:
.....
Setting up realplay symlinks in
/usr...
configuring icons...
configuring document icons...
configuring pixmapes...
configuring locale...
configuring desktop...
configuring applications...
configuring GNOME mime types...
```

```
Configuring realplay script...
RealPlayer installation is
complete.
Cleaning up installation
files...
Done.
```

Para usar la aplicación, la ejecutamos desde una terminal en modo gráfico con el comando:

```
! /usr/bin/realplay
```

Si tenemos dudas sobre la ubicación del binario ejecutable, podemos consultarle al sistema con el comando **whereis**:

```
! whereis realplay
realplay: /usr/bin/realplay /usr/lib/
realplay /usr/share/realplay
```

## Borrado de archivos según la fecha

Seguramente, en alguna oportunidad se nos presentó la necesidad de borrar archivos según una determinada fecha y, por supuesto, recurrimos al comando **ls -l** para verlas y luego proceder a borrar con **rm**. Este método es útil con pocos archivos ya que con varios, la cosa se complica. Es por eso que incluimos un truco con el que podremos borrar los archivos de forma simple y con tan sólo una línea.

```
# find -mtime +60 -type f -exec rm "{}" ;
```

Este comando buscará en nuestro sistema los archivos que tengan más de **60 días** y procederá a borrarlos uno a uno con el comando **rm**. Debemos reemplazar el **60** por la cantidad de días que necesitemos y listo.

real

My Account | Customer Support | Subscriber Upgrade | In the News!

Guide Radio Premium Video Free Games

### RealPlayer 10 for Linux is here

Includes these great features and more:

- Play popular datatypes**  
RealPlayer® 10 supports RealAudio, RealVideo 10, MP3, Ogg Vorbis and Theora, H263, AAC and more. Get ready for accelerated video, full screen playback, and a lot more to play.
- Mozilla-compatible plug-in**  
You can now watch and listen to embedded video right in your Web browser without opening RealPlayer. Enjoy media from your favorite music and news sites with just one click.
- New UI adopts your theme**  
The elegant UI design is based on GTK technology. That means the player adopts the theme you've chosen, and blends itself accordingly into your desktop environment.

Support Information  
Mailing List  
Helix Community Forums  
Nightly Player Builds

[Download RealPlayer](#)

Advanced Installation  
RedHat Package

Figura 9. La última versión de RealPlayer para Linux se puede descargar de [www.real.com/linux?pcode=rn&src=freeplayer\\_partner&opage=freeplayer\\_partner](http://www.real.com/linux?pcode=rn&src=freeplayer_partner&opage=freeplayer_partner).

## Arranques personalizados con LILO

Muchas veces tenemos la necesidad de **bootear** en forma específica en un determinado nivel (**runlevel**) para realizar tareas de administración (como chequear la parte física o lógica del disco, operación riesgosa en los **niveles 3 y 5**). A veces puede ser necesario entrar en el **nivel 1** para cambiar la clave de **root** olvidada. Esta operación se realiza de forma muy simple con LILO, ya que sólo debemos teclear al inicio de nuestro sistema el número de nivel en el que se quiere iniciar:

**boot: Linux 1**

Además, desde el inicio podemos definir una determinada cantidad de memoria operativa en nuestro sistema. Esta operación se utiliza cuando queremos testear

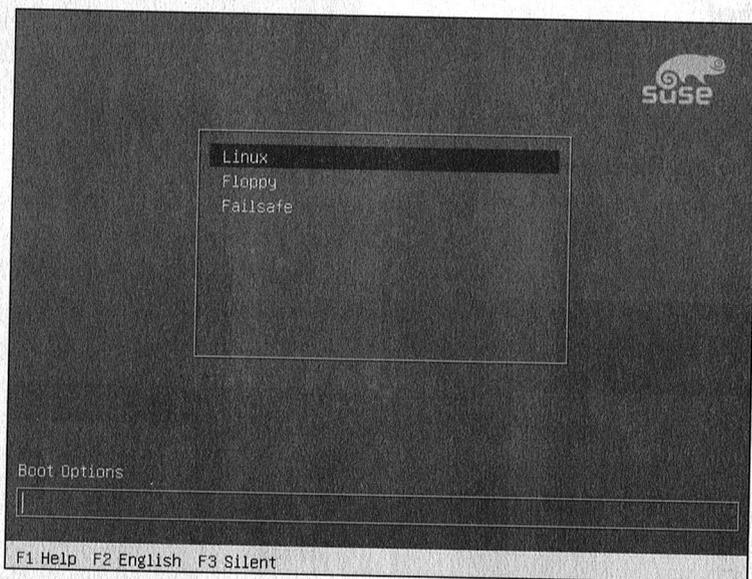
## WORD EN LA CONSOLA

Existe una herramienta llamada **antiword** (<http://freshmeat.net/projects/antiword>), que permite visualizar el contenido de un archivo **.DOC** en la pantalla. Para usarla, se debe tipear **antiword [archivo]**. Si no poseemos la herramienta instalada y somos usuarios de **Ubuntu** o cualquier otra versión de **Debian**, debemos instalarla con **apt-get install antiword**.

una determinada distribución con menos recursos de los que realmente tiene el sistema. La forma de hacerlo es simple:

**boot: Linux mem=128M**

De esta manera, le decimos al sistema que inicie con **128 RAM** de memoria sin importar cuánta sea realmente la cantidad de memoria física.



**Figura 10.** Si nuestra distribución nos muestra un menú gráfico de inicio, debemos presionar la tecla **ESC** para pasar al modo de línea de comandos de LILO.

## Cómo configurar fecha y hora desde la consola

Con tres simples líneas podemos fijar la hora de nuestra distribución según nuestro hardware, utilizando el formato **MM/DD/hh:mm**.

```
# date MMDDhhmmCCYY
# hwclock --utc --systohc
# hwclock --show
```

Por el contrario, quizá debamos sincronizar nuestro equipo para que tome la fecha y hora de un determinado servidor remoto y esto se logra con el siguiente comando:

```
# ntpdate server
```

Si contamos con una conexión permanente a Internet, podemos incluir la línea al archivo **/etc/cron.daily**. Esta operación es muy utilizada en redes donde la sincronización horaria es muy importante.

## Cómo habilitar el usuario root en Ubuntu

Muchos usuarios de las distribuciones **Ubuntu** no saben cuál es la clave de



Algunas distribuciones, como **Ubuntu**, deshabilitan la posibilidad de usar la cuenta **root** para evitar que los usuarios con poca experiencia puedan cometer errores que pongan en peligro la estabilidad del sistema. Aunque nos consideremos usuarios con experiencia, recordemos que es conveniente utilizar el sistema operativo de manera cotidiana con una cuenta de usuario normal.

**root**, que parece ser una cuenta de usuario no habilitada. La cuestión es que esta distribución utiliza un sistema llamado **sudo** que permite que los usuarios ejecuten un comando con privilegios de **root** sin usar la cuenta del administrador (para evitar problemas). De todas formas, también podemos activar y asignar una clave al usuario **root** de forma muy simple y con un solo comando:

```
sudo passwd root
```

## Cambiar la imagen de Grub

Cambiar la imagen que muestra **Grub** al inicio puede ser una buena práctica cuando empezamos a aburrirnos del estado general de nuestro sistema (**Figura 11**). El proceso no es complejo y requiere de

## CÓMO CONFIGURAR EL TECLADO EN ESPAÑOL

Algunas distribuciones no nos preguntan por la configuración de nuestro teclado al momento de instalarlas. Esto hace que el uso del sistema operativo una vez iniciado se vuelva bastante complicado, especialmente en la consola. Para configurar el teclado en español, simplemente hay que usar el comando **loadkeys es**.

algunas modificaciones a nuestra foto favorita. Para lograrlo, debemos abrir **Gimp** y seleccionar una imagen que nos guste. Luego debemos cambiarle la resolución a 640 píxeles de ancho por 480 píxeles de alto. Después de eso debemos hacer un clic con el botón derecho del mouse sobre esa imagen y seleccionar la opción **Imagen/Modo/Indexado**.

Dentro de la pantalla de indexado seleccionamos del lado izquierdo la opción **Generar gama de colores óptima** y, en el lado derecho, (**Número de colores**) ingresamos el número **14**. Hacemos clic en el botón **Aceptar** y guardamos el archivo con extensión **XPM**.

Si por cualquier razón Gimp no nos permitiera guardar el archivo con esta extensión, entonces debemos guardarlo como **PNG** y luego convertirlo con el siguiente comando:

```
convert nombre_imagen.png
nombre_imagen.xpm
```

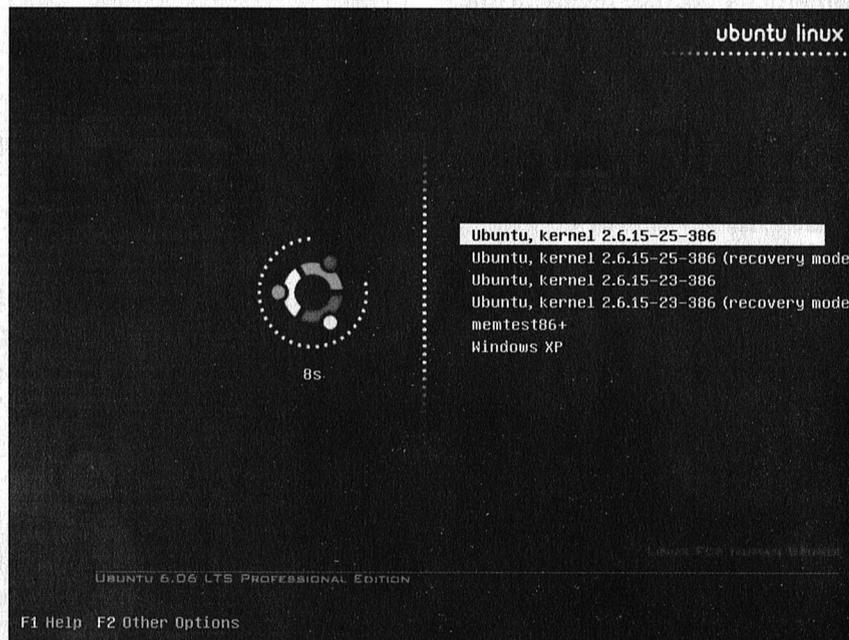
Una vez creada la imagen, como **root** ejecutamos los siguientes comandos:

```
gzip nombre_imagen.xpm
mv nombre_imagen.xpm.gz /boot/grub/
```

Paso seguido, editamos el archivo **menu.lst** ubicado en el directorio **/boot/ grub/** en el que indicaremos el nombre de nuestra imagen.

```
# Splash Image
splashimage=(hd0,0)/grub/nombre_
imagen.xpm.gz
```

Ahora sólo resta que reiniciemos nuestro equipo para ver el inicio de Grub con nuestra imagen preferida.



**Figura 11.** Es posible modificar el look de Grub para hacer, por ejemplo, que el menú de inicio de Ubuntu se vea como el de SuSE.

## Cómo crear un disco de arranque con menú de GRUB

Siempre es bueno tener un disco de inicio de nuestra distribución para tener a mano en caso de que aparezcan problemas. Para esto, primero debemos crear un sistema de ficheros en un disquete con el comando **mke2fs /dev/fd0** y montar el disco en algún directorio (por ejemplo **/mnt** o **/media**). Luego debemos copiar las imágenes de GRUB al directorio **/mnt/boot/grub**. Cabe aclarar que sólo **stage1**, **stage2** y **menu.lst** son necesarios. Finalmente, debemos desmontar el disco flexible y ejecutar los siguientes comandos:

```
/sbin/grub -batch -device
map=/dev/null <<EOT
device (fd0) /dev/fd0
root (fd0)
setup (fd0)
quit
EOT
```

## Cómo escuchar archivos MP3 desde la consola

Este truco está pensado para las personas que no utilizan el entorno gráfico de forma continua. Con él podremos escuchar archivos MP3 y OGG sin ningún tipo de problemas desde una consola. Para ello, sólo debemos tener instalada en nuestro equipo la aplicación **mpg123** e ingresar la siguiente línea:

```
mpg123 [archivo.mp3]
```

## RPM FORZADO

A veces sucede que queremos instalar un paquete RPM y el sistema nos devuelve un montón de mensajes de errores de **dependencias y componentes faltantes**. Con frecuencia, esos componentes están instalados en el sistema y el problema se debe a que RPM no puede encontrarlos. En esos casos, es bueno probar instalar el paquete de forma forzada al menos para saber si funcionará. Esto puede ser realizado con el siguiente comando:

```
rpm -i --force --nodeps [archivo.rpm]
```

Si nuestro archivo MP3 tiene espacios en blanco en el nombre, deberemos usar comillas al inicio y al final. De cualquier manera, podemos escribirlas en todos nuestros archivos. Para dejar el archivo reproduciéndose y seguir usando la consola podemos usar:

```
mpg123 -q "canción.mp3" &
```

Si pulsamos una vez la tecla **ENTER**, empezaremos a escuchar la canción, pero la consola quedará liberada para la introducción de nuevos comandos.

Con este formato no veremos los nombres de las canciones pero sí el número de proceso que está corriendo. Si quisiéramos interrumpir la canción, deberíamos ingresar el siguiente comando:

```
kill -9 <número-proceso>
```

Seguramente, en algún momento desearemos escuchar toda una lista com-

pleta de canciones. Para eso, tendremos que usar este comando:

```
mpg123 -@ "lista.m3u"
```

Si queremos dejar liberada la consola:

```
mpg123 -@ "lista.m3u" -q &
```

Para crear una lista de un directorio completo de archivos:

```
ls -1 *.mp3 > lista.m3u
```

Luego podemos cambiar el orden de las canciones con cualquier editor de texto. Por último, para manejar el volumen de salida debemos utilizar el comando **aumix** (Figura 12).

## Instalar paquetes .DEB en Red Hat y .RPM en Debian

En muchos casos, puede pasar que no encontremos paquetes compilados para nuestra distribución preferida. Como bien lo informa el título, vamos a ver cómo instalar paquetes .DEB (de De-

## BÚSQUEDA DE COMANDOS

**Command not found** es un error típico de la terminal de Linux cuando escribimos el nombre de un comando que no existe. Las distribuciones **Ubuntu** incluyen una herramienta llamada **command-not-found** que, al instalarla, reemplaza ese mensaje por un sistema de búsqueda del comando en los repositorios de Ubuntu. Entonces, si tipeamos el nombre de un comando que existe pero no está instalado en el sistema, Ubuntu intentará descargarlo e instalarlo automáticamente.

bian) en distribuciones basadas en .RPM y el caso contrario, instalar paquetes RPM en Debian. Esto lo solucionamos con un sistema llamado **alien**, que viene incluido en muchas distribuciones. Si no está en la nuestra, lo podemos descargar de <http://kitenet.net/~joey/code/alien.html>.

Una vez instalado, se utiliza del siguiente modo. Para generar un .DEB a partir de un .RPM:

```
alien --to-deb paquete.rpm
```

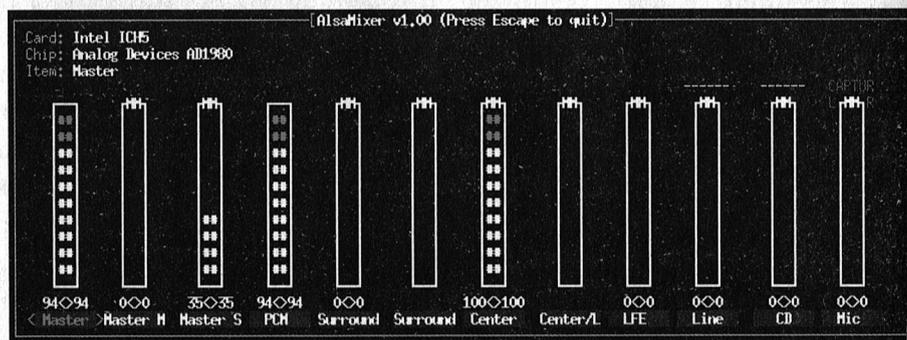


Figura 12. Aumix es una herramienta para controlar los niveles de la placa de sonido desde la terminal.

Y para generar un .RPM a partir de un .DEB ingresamos la siguiente línea:

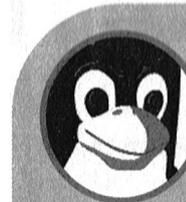
```
alien --to-rpm paquete.deb
```

Luego, sólo es cuestión de instalar los paquetes como lo haríamos si los hubiéramos bajado directamente en el formato de nuestra distro.

## Cómo instalar los archivos man en español

Es muy desalentador para los nuevos usuarios de GNU/Linux el hecho de encontrarse con los manuales del sistema en inglés. Para bajarlos en español, tendremos que ingresar a la siguiente URL para seleccionar la descarga correspondiente a español: [www.win.tue.nl/~seb/linux/man](http://www.win.tue.nl/~seb/linux/man) (Figura 13).

Luego de descargarlo, debemos descomprimir el archivo en un directorio como



El proyecto de traducción de las páginas del manual al español es llevado a cabo por colaboradores independientes. Podemos realizar un seguimiento de la evolución del trabajo si visitamos la página oficial del proyecto TLDP en español en <http://es.tldp.org>. Allí encontraremos la última versión y podremos acceder a la lista de correo especial para este trabajo.

/usr/local. Para todo esto necesitamos estar trabajando con la cuenta **root**:

```
tar -jxvf man-pages-es-1.55.tar.bz2
```

Ingresamos en el directorio que se creó y lo instalamos con los comandos **make** y **make install**. Por último, tendremos que editar el directorio **/etc/profile** para especificar que brinde prioridad al lenguaje español agregando la siguiente línea:

```
export LANG=es_ES
```

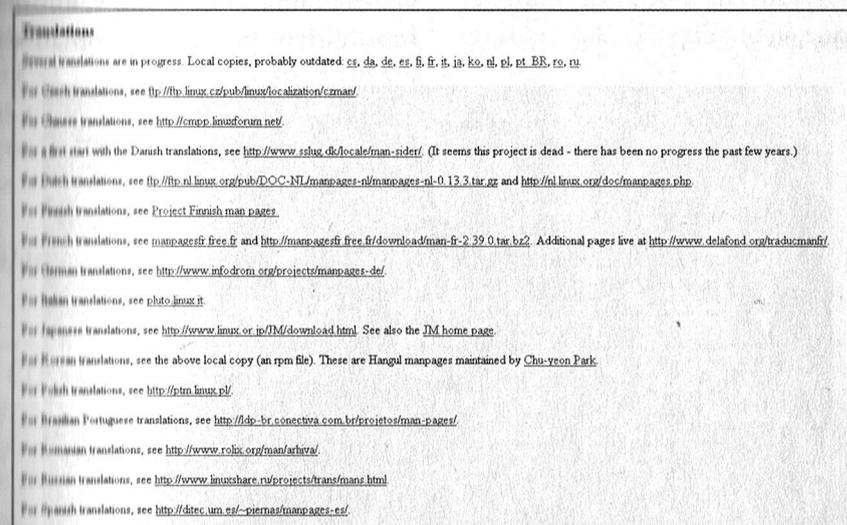


Figura 13. Entre todos los idiomas disponibles, podemos encontrar el español al final de la lista.

Luego debemos salir del shell con el comando **exit** y volver a entrar para ver los resultados: todos los manuales deberían estar en español (Figura 14).

## Capturar el texto de la consola

En muchos casos, puede ser muy útil poder enviar a la impresora los datos contenidos en la consola o imprimir un archivo de texto directamente sin abrir primero un editor. Para poder llevar a cabo estas tareas, debemos usar el siguiente comando para guardar los datos en un archivo:

```
script archivo.txt
```

En cambio, para enviar los datos a una impresora conectada al puerto lp0 debemos ingresar el siguiente comando (se debe reemplazar lp0 por el archivo de conexión que corresponda):

```
script /dev/lp0
```

```
LS(1) LS(1)
NOMBRE
ls, dir, vdir - listan los contenidos de directorios
SINOPSIS
ls [opciones] [fichero...]
dir [fichero...]
vdir [fichero...]
Opciones de POSIX: [-CFRacdilqrstu]
Opciones de GNU (en la forma más corta): [-labcdfghiklmnopqrstuwvXBCD-
FGHLMQRSUX] [-w cols] [-T cols] [-l patrón] [--full-time] [--show-con-
trol-chars] [--block-size=tamaño] [--format={long,verbose,com-
mas,across,vertical,single-column}] [--sort={none,time,size,extension}]
[--time={atime,access,use,ctime,status}] [--color={none,auto,always}]
[--help] [--version] [--]
DESCRIPCIÓN
El programa ls lista primero sus argumentos no directorios fichero, y
luego para cada argumento directorio todos los ficheros susceptibles de
listarse contenidos en dicho directorio. Si no hay presente ningún
argumento aparte de las opciones, se supone un argumento predeterminado
```

Figura 14. Las páginas del manual (accesibles con el comando **man**) son indispensables para aprender a dominar al máximo todos los comandos.

## Cómo dividir grandes archivos para que entren en discos floppy

Suele suceder que debemos transferir cierta cantidad de información de una computadora a otra y el espacio ofrecido por un disco floppy no es suficiente. Pero hay una solución: comprimirlos en un solo archivo y luego dividirlos con el comando **split**. Primero debemos comprimir todo como un paquete **.tar.gz**:

```
tar -cvf nombre_archivo.tar *
```

Luego lo tenemos que dividir en varios archivos más pequeños de 1.44MB (el tamaño de un disco floppy) utilizando el comando **split**:

```
# split -b 1440k nombre_archivo.tar.gz
```

De esta manera, se dividirá el archivo **nombre\_archivo.tar.gz** en varios archivos del tamaño definido y, ahora sí, estos archivos podrán ser copiados a dis-

quetes para su transferencia. Para volver a unirlos, podemos utilizar el comando **cat** del siguiente modo:

```
cat nombre_archivo1.tar.gz nombre_
archivo2.tar.gz > nombre_archivo.tar.gz
```

## Cómo crear imágenes ISO

En general, los usuarios de GNU/Linux no tenemos en cuenta que, cuando ejecutamos determinadas aplicaciones de forma gráfica, en realidad lo que estamos haciendo es utilizar entornos gráficos de programas que corren de forma nativa en Linux desde la consola.

En el caso de cualquier aplicación gráfica para grabar CDs o DVDs estamos en la misma situación. Por ejemplo, cuando utilizamos **K3B** estamos usando **cdrecord**, **mkisofs**, **cdrdao**, etcétera. Esto quiere decir que podremos utilizar nuestra grabadora de forma remota con tan sólo un logueo vía ssh o incluso desde nuestra consola. Entonces, vamos a generar una imagen ISO de un CD con el comando **dd** o el comando **cat** (se llega al mismo resultado).

```
dd if=/dev/cdrom of=imagen.iso
```

## LISTAR DISPOSITIVOS PCI

Un comando muy útil a la hora de configurar el hardware es **lspci**, que muestra en pantalla un listado completo de todas las tarjetas conectadas a la computadora. Incluye datos de hardware y del fabricante.

o:

```
cat /dev/cdrom > imagen.iso
```

Luego sólo resta grabar la imagen a otro CD con el comando **cdrecord**:

```
cdrecord -v speed=VELOCIDAD dev=GRABADORA imagen.iso
```

## Obtener información de paquetes .DEB

Para obtener información sobre algún programa instalado, sólo debemos teclear en la consola el siguiente comando:

```
$ apt-cache show "nombre del paquete"
```

Por ejemplo:

```
$ apt-cache show postfix
```

## CÓMO DESCARGAR ARCHIVOS DESDE LA CONSOLA

Existe un comando llamado **wget** que sirve para descargar archivos e incluso sitios web completos. Funciona muy bien y permite continuar descargas previamente interrumpidas. Su forma de utilización es la siguiente: **wget -c [url\_archivo]**. Para obtener más información sobre este comando podemos utilizar el comando **man wget**.

De forma automática, obtendremos una respuesta como la siguiente:

```
$ apt-cache show postfix
Package: postfix
Priority: optional
Section: mail
Installed-Size: 2172
Maintainer: LaMont Jones
<lamont@debian.org>
Architecture: i386
Version: 2.2.10-1ubuntu0.1
Replaces: postfix-doc (<< 1.1.7
0), postfix-tls, mail
transport-agent...
```

## Convertir texto seleccionado en código HTML

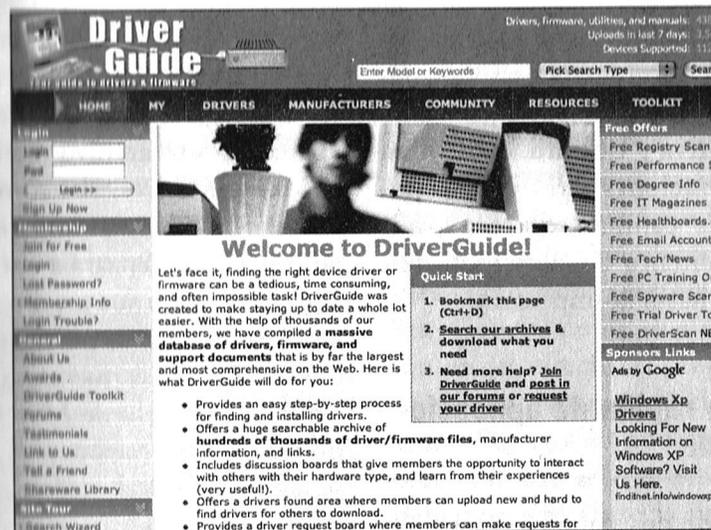
Convertir una porción de texto seleccionado en código HTML es muy útil para crear sitios web. Para hacerlo, sólo tipeamos desde la consola de **Vim**:

```
so \$(VIMRUNTIME/syntax/2html.vim
```

Sólo resta guardarlo con `:w archivo.html` y luego `:q`.

## Dónde buscar controladores de dispositivos

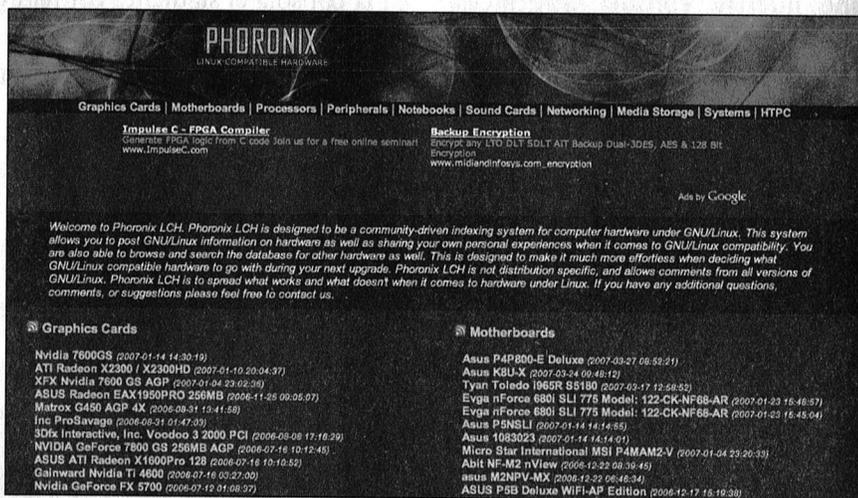
Hay varias herramientas que nos permiten averiguar dos datos esenciales: el primero de ellos es saber si el dispositivo que nosotros queremos configurar tiene soporte en el mundo de GNU/Linux. En tal caso, debemos averiguar el segundo dato esencial: dónde encontrar el **driver** que funciona con nuestra distribución. El proyecto **Phoronix** ([www.phoronix.com/lch/?k=home](http://www.phoronix.com/lch/?k=home)) es una inmensa base de datos creada por usuarios, que tiene información sobre miles de dispositivos (**Figura 15**). Entre otras cosas, los usuarios pueden poner comentarios como, por ejemplo, dónde encontraron el driver, cómo lo



**Figura 16.** DriverGuide es un gran repositorio de controladores de hardware para Linux y Windows.

configuraron en su distribución, los datos que detecta el kernel sobre el dispositivo en cuestión y advertencias sobre el funcionamiento y el rendimiento, entre otras cosas. El sitio incluye un buscador para encontrar más fácilmente lo que estamos buscando. Si tipeamos, por ejemplo, la palabra **iPod**, encontraremos un link que nos derivará hacia comentarios de usuarios que han logrado conectar ese popular dispositivo con GNU/Linux, entre las herramientas más populares de sincronización y mucho más. Este sitio indispensable a la hora de encontrar drivers para GNU/Linux es

**DriverGuide** ([www.driverguide.com](http://www.driverguide.com)) que, si bien es un portal que incluye una base de datos de drivers para múltiples plataformas, la sección Linux es muy amplia e incluye un buscador que nos permitirá llegar rápidamente al preciado archivo con sólo poner algunas palabras clave (**Figura 16**). Antes de instalar una distribución, debemos consultar en su sitio oficial la sección **Compatibilidad con Hardware** (o **Hardware Compatibility**) porque allí podremos informarnos rápidamente si la distribución que deseamos instalar incluye soporte para todos los dispositivos que tenemos en



**Figura 15.** Phoronix es un gran repositorio de información sobre el soporte de hardware en Linux.

## CÓMO BUSCAR TEXTO DENTRO DE ARCHIVOS

Existe una forma muy simple de buscar una cadena de texto dentro de un archivo, y consiste en combinar los comandos **cat** y **grep** del siguiente modo: `cat [archivo] | grep [cadena]`. Lo que veremos en pantalla serán las líneas del archivo en las que aparece la cadena utilizada como parámetro del comando **grep**.

nuestro equipo. Esta base de compatibilidad sólo la encontraremos en los sitios de las distribuciones importantes como **SuSE, Mandriva y Ubuntu**, entre otras. Elegir distribuciones raras y poco populares es un camino seguro a pasar horas en el intento de configurar desde el mouse hasta la impresora.

## Instalación rápida de aplicaciones con klik

Klik consta de dos componentes: uno que se instala en nuestro navegador web y un sitio (<http://klik.atekon.de>) que tiene miles de aplicaciones divididas por categorías, en el que además se incluye un buscador (**Figura 17**).

¿Cómo funciona? Muy simple: una vez que tenemos el sistema klik instalado, lo único que tenemos que hacer es dirigirnos al sitio de klik con nuestro na-

vegador favorito, elegir la aplicación que queremos, **hacer klik** sobre ella y se descargará. En nuestro escritorio aparecerá un archivo con extensión **.CMG**. Entonces, si hacemos clic sobre ese archivo, automáticamente la aplicación comenzará a correr.

Podemos armarnos un directorio **Aplicaciones** y almacenar allí todos los archivos **.CMG** si queremos tener algo similar al modelo de **Mac OS X**. Para borrar las aplicaciones el proceso es sencillo: enviar el archivo **.CMG** a la papelera de reciclaje.

El proceso de instalación (y posterior uso) de klik es muy sencillo. Se recomienda utilizar **Firefox** como navegador y **KDE** o **GNOME** como gestores de escritorios para que, cada vez que descarguemos un **.CMG**, éste aparezca como un nuevo icono en el escritorio. Podemos ver el procedimiento para instalar klik en el **Paso a paso: Instalación de Klik**.

The screenshot shows the klik website interface. At the top, there is a navigation bar with links for Home, Blog, Wiki, Forum, Chat, Maintainers, and Contact, along with a package search box. The main content area features a large download icon and the text: "klik strives to be the easiest way to download and RUN software, without installation". Below this, there are instructions for installing the klik client. A sidebar on the right lists various application categories such as Artistic Software, Communications, Database, Desktop Environment, Documentation, Education, Games/Entertainment, Home Automation, Information Management, Internet, Multimedia, Office/Business, Other/Nonlisted Topic, Printing, Religion, Scientific/Engineering, Security, Software Development, System, Terminals, Text Editors, Text Processing, and Utilities. At the bottom of the sidebar, it states "357080 packages successfully delivered".

**Figura 17.** El proyecto klik posee un gran directorio de aplicaciones de fácil instalación para Linux.

El no queremos que klik descargue los **.CMG** directamente en el escritorio podemos hacer, por ejemplo, que aparezcan en nuestro nuevo directorio **Aplicaciones**. Para eso debemos crear un archivo llamado **.klikrc** en nuestro directorio personal e incluir lo siguiente:

```
# Definimos el destino de los
# archivo .cmg en
# $HOME/Aplicaciones.
export
DESTINATION=$HOME/
Aplicaciones
```

## Agregar PATHs perdidos

Al escribir un comando, debemos tener presente que cualquier Linux debe poder encontrar el directorio que contiene el comando antes de poder ejecutarlo. La variable de entorno **PATH** contiene la lista de directorios en los que deseamos que Linux realice las

búsquedas de comandos. La variable **PATH** debe contener todos los directorios necesarios para localizar todos los comandos que utilicemos.

Es muy común instalar algunas aplicaciones que no ponen sus ejecutables en los directorios estándares para ejecutables como **/bin**, **/usr/bin** y **/usr/local/bin**. Generalmente, sus ejecutables están en su directorio propio y este directorio no aparece en el **PATH** del sistema. Por eso no podemos ejecutar el comando con sólo escribirlo, sino que debemos especificar la ruta absoluta al comando. Además, cabe aclarar que tampoco podremos usar el autocompletar (**TAB**) para el comando.

La solución es agregar el directorio a la variable de entorno **PATH**, algo que se puede realizar con el comando **export** de la siguiente manera:

```
export PATH=$PATH:/usr/local/share/
```

En el ejemplo estamos agregando el directorio **/usr/local/share** al **PATH**. Hay

## Instalación de klik

PASO A PASO

1 En una terminal virtual en modo gráfico, ejecute el siguiente comando para instalar la infraestructura necesaria de klik:

```
wget klik.atekon.de/client/
install -O -|sh
```

2 Luego abra el navegador e ingrese a la dirección <http://klik.atekon.de>.

3 Navegue por el directorio de aplicaciones y haga clic sobre la que quiera instalar.

4 Luego aparecerá en su escritorio un archivo **.CMG**. Si hace clic sobre él, encontrará la aplicación que quiere ejecutar.

que tener en cuenta que los directorios en la variable **PATH** se separan por dos puntos (:). Si queremos que la variable **PATH** siempre esté actualizada con las modificaciones, debemos guardar el comando para que se ejecute en algunos de los dos archivos de configuración de Bash (`~/bashrc` o `/etc/profile`). Basta con agregar el comando al final del archivo.

## Cómo entrar al modo gráfico si el sistema no lo hace automáticamente al inicio

A veces, sucede que luego de instalar el sistema operativo, reiniciamos y éste sólo inicia en modo texto. Es decir, no nos muestra el **entorno gráfico** que tanto queremos. Esto puede deberse a que durante el proceso de instalación del sistema, no hemos configurado correctamente el modo gráfico. Para evitarlo, siempre hay que prestar especial atención a la configuración gráfica en la instalación y probar la configuración antes de continuar.

Sin embargo, una vez que tenemos el problema, lo que podemos hacer es in-



El archivo de configuración de X.Org se encuentra en `/etc/X11/xorg.conf`. Allí podremos modificar los controladores de video y la resolución de la pantalla. Luego podemos

probar la configuración con el comando **X**. Antes de hacer cambios, conviene que conozcamos las características de nuestro hardware y sepamos qué resoluciones y configuraciones pueden soportar.

## III CERRAR EL SISTEMA

Prácticamente, todos los procesos de Linux pueden ser llevados a cabo de más de una forma diferente. Un buen ejemplo de esto es el acto de apagar el sistema, que se puede realizar de las siguientes maneras:

```
halt
shutdown -h now
init 0
```

gresar al modo gráfico con el comando **startx**. Si no funciona, entonces deberemos iniciar la **herramienta de configuración** de la distribución que estemos usando. En el caso de Mandriva, por ejemplo, con sólo tipear **setup** la tendremos en pantalla para configurar el modo gráfico como corresponde.

## Qué hacer cuando una instalación de Linux no finaliza correctamente

A veces, cuando estamos instalando una distribución de GNU/Linux, el sistema de instalación se congela sin respuesta aparente y no avanza más. Esto puede deberse a varias razones, pero las principales son: problema físico en el CD o DVD (que esté rayado o mal grabado), problema del medio de almacenamiento (clusters malos en el disco rígido), problema en las particiones (que estén mal hechas, dañadas o solapadas) y por último, problema en el hardware del equipo (que esté mal la memoria RAM o el procesador es lo más común).

Pero, ¿cómo saberlo? La manera más común es abrir la terminal de texto del sistema de instalación. Todas las distribuciones permiten acceder a ellas por medio de la combinación de las teclas **ALT + FUNCION[n]** (reemplazar la **n** por la que corresponda. Para saber, hay que ir probando todas las posibilidades). Una vez que se encuentra la terminal, se podrán ver todos los mensajes que devolvió el sistema de instalación. Si se cuelga, sólo es cuestión de ver cuál fue el último mensaje de error para tratar de solucionar el problema.

## Comandos de ayuda útiles

Al utilizar el sistema, los usuarios de GNU/Linux no se encuentran solos. Existen algunos comandos y parámetros que permiten obtener ayuda sobre un comando o una aplicación en especial. Lo primero que debe hacer el usuario es chequear que exista una página del manual de dicho comando. Por ejemplo, si queremos obtener ayuda sobre el comando **ls**, sólo tenemos que escribir la siguiente línea:

```
man ls
```

Así aparecerá el sistema de manuales con la ayuda en pantalla. Algunas veces, esta ayuda estará en idioma español y otras veces no. En realidad, eso depende de que el grupo de traducción de aplicaciones y documentos de GNU/Linux haya llegado a los datos que estamos manejando. Para salir del sistema de manuales, sólo tenemos que presionar la tecla **q**.

Si obtenemos un mensaje de error, entonces es muy probable que esa aplicación o comando no incluya páginas del manual. Todavía quedan opciones. Podemos probar agregando el parámetro **-help** al comando, de la siguiente manera:

```
ls -help
```

Lo que obtendremos en pantalla es una escueta guía de uso de ese comando, compuesta por un listado de parámetros y sus respectivas descripciones de acción.

Por último, debemos recordar que en los directorios `/usr/doc` y `/usr/share/doc` se encuentran almacenadas las documentaciones de todos los comandos y aplicaciones que tengamos instalados en el sistema. Por eso, darse una vuelta por esos directorios con algún editor de texto puede ser una buena idea.

## III TRABAJAR CON EL HISTORIAL

Eliminar la herramienta de historial de Bash nos permitirá ahorrar mucho tiempo de tipeo. El comando **history** nos muestra un listado de todos los comandos previamente utilizados en Bash. Cada comando está precedido por un número único que lo identifica y, si queremos volver a ejecutar ese comando, debemos escribir el símbolo **!** seguido del número de comando que tenemos que ejecutar sin espacio intermedio. Por ejemplo: **!508**.

# GUIA DE COMANDOS

A continuación presentamos una guía con los comandos más importantes que se pueden usar en el modo texto de GNU/Linux.

COMANDO	DESCRIPCIÓN
cd	Cambia de directorio.
compress	Comprime archivos.
cp	Copia un archivo.
chmod	Cambia los permisos de un archivo o directorio.
chown	Cambia el propietario de un archivo o directorio.
df	Muestra el espacio libre en disco.
du	Muestra el espacio en disco utilizado.
fdformat	Formatea un disquete.
fdisk	Utilidad para particionar unidades.
find	Encuentra un archivo.
fsck	Chequea el sistema de archivos.
gzip	Descomprime un archivo en formato GZIP.
ln	Crea un enlace simbólico.
ls	Muestra el contenido de un directorio (análogo a <i>dir</i> ).
mkdir	Crea un directorio.
mkfs	Crea un nuevo sistema de archivos.
mknod	Crea un archivo de dispositivo.
mkswap	Crea un espacio de intercambio.
mount	Monta una unidad o partición en el sistema de archivos.
mv	Mueve un archivo (se utiliza también para renombrar).
pwd	Devuelve la cadena correspondiente al directorio actual.
rm	Borra un archivo.
rmdir	Borra un directorio.
split	Divide un archivo en muchos archivos de un tamaño determinado.
swapoff	Desactiva el espacio de intercambio.
swapon	Activa el espacio de intercambio.
sync	Sincroniza los procesos de escritura y lectura.
tac	Da vuelta el contenido de un archivo.
tail	Muestra las últimas líneas de un archivo.
tar	Desempaqueta un archivo en formato TAR.
umount	Desmonta un sistema de archivos ya montado.
uniq	Borra las líneas duplicadas de un archivo.
wc	Cuenta el número de caracteres de un archivo.

**Tabla 1.** Comandos para el manejo de archivos.

COMANDO	DESCRIPCIÓN
bg	Ejecuta un proceso (interrumpido) en segundo plano.
fg	Ejecuta un proceso (interrumpido) en primer plano.
free	Muestra la memoria libre y la utilizada.
halt	Apaga la máquina.
kill	Envía una señal a un proceso.
ldd	Muestra las librerías que se necesitan para ejecutar un programa.
nice	Ejecuta un proceso con menor prioridad de ejecución.
ps	Muestra todos los procesos que se están ejecutando en el sistema.
printenv	Muestra las variables de entorno.
pstree	Como el <i>ps</i> , pero muestra todo en forma de árbol.
reboot	Reinicia el sistema.
shutdown	Cierra el sistema.
top	Una utilidad para monitorear procesos y el estado del sistema.
uname	Muestra información del sistema.

**Tabla 2.** Comandos para el manejo de procesos.

COMANDO	DESCRIPCIÓN
adduser	Crea un nuevo usuario.
chsh	Cambia el shell de un usuario determinado.
groups	Muestra el listado de grupos de usuarios del sistema.
id	Muestra la información de usuario y grupo de un determinado usuario.
passwd	Cambia la clave de acceso de un determinado usuario.
su	Cambia de usuario.
talk	Permite chatear con otros usuarios.
users	Lista los usuarios conectados al sistema.
who	Muestra información de los usuarios actualmente conectados.

**Tabla 3.** Comandos para el manejo de usuarios.

COMANDO	DESCRIPCIÓN
alias	Permite ejecutar comandos utilizando otros nombres (crea alias).
apropos	Encuentra las páginas del manual para un comando determinado.
cal	Muestra el calendario.
date	Permite ver y cambiar el día y la hora actuales.
info	Muestra ayuda de un determinado comando.
man	Muestra las páginas del manual de un determinado comando.
unalias	Borra el alias de un comando.

**Tabla 4.** Otros comandos.

- 1 Es posible ver un listado completo de los comandos tipeados por el usuario si accedemos al archivo **.bash\_history** de su directorio personal.
- 2 Con el signo de admiración (!) y un número de comando del listado que ofrece el comando **history**, es posible volver a ejecutar un comando de manera rápida y sencilla.
- 3 En algunas distribuciones, con **CTRL + PAGE UP** y **CTRL + PAGE DOWN** es posible hacer **scroll** por una terminal.
- 4 Bash guarda su configuración en **/etc/bashrc** (para todos los usuarios) y en **/\$HOME/.bashrc** y **/\$HOME/.bash\_profile** para la configuración particular de cada usuario.
- 5 Se puede usar el comando **dd** para crear una imagen de bajo nivel de un floppy del siguiente modo: **dd if=/dev/fd0 of=/imagen**.
- 6 Con la combinación **CTRL + ALT + RETROCESO** se puede reiniciar el sistema.
- 7 Si editamos el archivo **/etc/inittab** es posible cambiar el comportamiento de la combinación de las teclas **CTRL + ALT + RETROCESO**.
- 8 Las teclas FLECHA ARRIBA y FLECHA ABAJO permiten navegar el historial de los comandos de bash.
- 9 Es posible apagar el sistema con el comando **init 0**.
- 10 Es posible cambiar el mensaje de bienvenida de una distro en modo texto si editamos los archivos **/etc/issue** y **/etc/motd**.

# Trucos avanzados para la consola

Continuamos con los trucos para la consola de comandos de Linux, pero en este apartado hemos reunido aquellos trucos que requieren un poco más de experiencia en el uso del sistema por parte del usuario.

Ejecutar comandos con permisos de otro usuario (suid)	50
Iniciar Linux sin password	50
Ejecución de comandos al inicio de nuestro sistema	51
Cómo crear servicios de inicio en Debian y Ubuntu	52
Cómo crear un archivo de intercambio (swap)	52
Usar make para automatizar tareas	53
Sugerencias para recompilar el núcleo	54
Cómo crear nuestro propio repositorio de paquetes Debian	55
Grabar canciones MP3 en un CD de audio desde la consola	55
Borrar los volcados de memoria Core	56
Cómo ejecutar aplicaciones de MS-DOS bajo Linux	57
Cómo sincronizar una Palm con Linux desde la consola	60
Cómo compartir la conexión a Internet en pocos pasos	60
Multitarea en el modo texto	63
Cómo acceder desde Ubuntu a un directorio compartido en red	65

## Ejecutar comandos con permisos de otro usuario (suid)

En muchas ocasiones, un proceso debe ejecutarse con privilegios diferentes a los que posee el usuario que lo está ejecutando. ¿Cómo es esto? Veamos una aplicación típica del mundo real:

Un usuario puede modificar su propia clave utilizando el comando **passwd**. La simple ejecución de este comando implicará la modificación de cualquiera de los dos archivos de usuarios (**/etc/passwd** o **/etc/shadow**). El usuario común no tiene privilegios para modificar estos archivos directamente y sin embargo el comando **passwd** que utiliza lo puede hacer sin problemas. ¿Por qué sucede esto? Porque el comando **passwd** tiene habilitado el bit **SUID**, que le permite manejarse con privilegios de **root** aun cuando un usuario común lo esté ejecutando.

¿Cómo funciona entonces? Simple, activando el bit **SUID** del comando **passwd**. Para esto, se utiliza el comando **chmod** (cambiar permisos) del siguiente modo:

**chmod u+s [archivo]**

Esto quiere decir que cuando se ejecute, el proceso correspondiente va a tener los privilegios del propietario del comando y no del usuario que lo lanzó. En el caso del comando **passwd**, su propietario es **root** (el administrador del sistema).

No debemos finalizar este truco sin advertir de la peligrosidad de este comando. Si se utiliza de forma incorrecta, puede permitir que cualquier usuario común lleve a cabo operaciones peligrosas en el sistema. Por eso hay que tener

sumo cuidado y, cada vez que se decida habilitar el bit **SUID**, asegurarse de que el comando no se pueda utilizar para poner en riesgo la seguridad del sistema.

## Iniciar Linux sin password

¿A quién no le ha ocurrido el olvido accidental de la clave de acceso alguna vez? Lo que se debe hacer en estos casos es iniciar el sistema en modo **mono usuario** para así tener los privilegios de **root** sin

## RESCATAR A LINUX

Suele suceder que cuando se instala Windows en un equipo que previamente tenía una partición con Linux, el sistema de Microsoft borra la **MBR** (*Master Boot Record*), e imposibilita así que arranque el Linux instalado. Para restaurar el gestor de arranque, debemos bootear con el CD original de la distribución que tengamos instalada o utilizar un disco de inicio (si lo tenemos) para iniciar el sistema Linux original. Luego se debe editar el archivo **/etc/lilo.conf** para agregar una entrada que permita bootear el sistema Windows recién instalado.

```
other = /dev/hda1
label = WindowsXP
table=/dev/hda
```

Se debe cambiar **/dev/hda1** por la partición en la que se encuentra instalado el sistema Windows. Finalmente, debemos guardar el archivo y tipear el comando **lilo** para volver a escribir la **MBR** con el gestor de arranque.

necesidad de tipear su clave. Para llevar esto a cabo en LILO, se debe utilizar el parámetro **single** seguido de la etiqueta que hace referencia a la instalación de Linux en el menú de inicio. Por ejemplo:

**linux single**

Otra forma es cambiar el proceso de inicio por bash:

**linux init=/bin/sh**

Si se utiliza el primer método, sólo hay que usar el comando **passwd** para cambiar la clave. Pero si se utiliza la segunda opción, notaremos que el inicio del sistema lo habrá hecho en modo de sólo lectura, por lo que no podremos modificar la clave (ni ningún archivo del sistema). Por eso, antes de cambiar la clave, debemos volver a montar el sistema de archivos con los privilegios correspondientes:

**mount / -o rw,remount**

## Ejecución de comandos al inicio de nuestro sistema

Suele suceder que al iniciar el sistema debemos tipear varios comandos para configurar ciertos parámetros. Por ejemplo, configurar un entorno de trabajo en red puede tomarnos entre dos y cinco comandos, y sería muy tedioso tener que escribirlos cada vez que Linux bootea. Por eso, lo mejor es crear un archivo de texto (un **script**) que contenga todos esos comandos, otorgarle permisos de ejecución (**chmod +x [archivo]**) y



Iniciar Linux en modo mono usuario es sinónimo de poner en peligro la integridad del sistema. ¡Mucho cuidado cuando habilitamos este nivel de ejecución!

hacer que el sistema lo ejecute de forma automática cada vez que se inicia.

El proceso para llevar a cabo este último paso varía según la distribución, ya que cada una guarda el archivo de inicialización en un directorio diferente y con un nombre distinto. Veamos en dónde guardan el archivo de inicialización las distribuciones más populares:

- RedHat y Mandriva: **/etc/rc.d/rc.local**
- SuSE: **/etc/rc.d/boot.local**
- Debian y Ubuntu: **/etc/rc.boot**

Para editar estos archivos, podemos recurrir al editor **mcedit**, **vi** o **emacs**. Para el ejemplo elegimos el editor **mcedit** y vamos a realizar un script que se ejecute al inicio del sistema y realice una copia del historial de bash de un usuario en particular (tarea muy útil en ambientes corporativos para poder tener un registro completo de los comandos tipeados por un determinado usuario):

```
# mcedit /etc/rc.d/rc.local
#!/bin/sh
#
# This script will be executed
# *after* all the other init
# scripts.
# You can put your own
# initialization stuff in here if
```

```
you don't
# want to do the full Sys V
style init stuff.
```

```
cp /home/usuario/.bash_history
/usr/local/back/usuario
```

Este ejemplo corresponde a una distribución Mandriva. Como se ve, al final del archivo está el comando **cp** que se encarga de llevar a cabo el proceso de copia. Este comando se ejecutará cada vez que se inicie el sistema (Figura 1).

## Cómo crear servicios de inicio en Debian y Ubuntu

Para poder lanzar procesos automáticamente al inicio de sistemas como Debian, Sarge, Ubuntu o cualquier distribución derivada de Debian, debemos seguir los siguientes pasos. En primer lugar tendremos que crear un **script**

dentro de **/etc/init.d/**. Lo guardamos y creamos un enlace a él en el directorio correspondiente al nivel de ejecución (por citar un ejemplo).

```
# /etc/rc.d/rc.2/ln -s /etc/init.d/varios
```

¡Listo! ahora podemos crear cualquier **script** y enlazarlo de la misma manera.

## Cómo crear un archivo de intercambio (swap)

Cuando el tamaño de la partición **swap** no es suficiente (porque se expandió la memoria RAM) y no hay opciones de modificar la tabla de particiones, la mejor opción es extender el espacio de intercambio con un archivo común. Para crear ese archivo, primero debemos usar la herramienta **dd** del siguiente modo:

```
dd if=/dev/zero of=/swap bs=1024
count=8192
```

Con esto habremos creado un archivo ubicado en **/swap** de 8MB. Se debe modificar el parámetro **count=** para asignar el espacio en bytes que se desea. El dispositivo **/dev/zero** es un dispositivo especial en donde todo lo que se lee son bytes nulos. El archivo, entonces, estará vacío. Ahora, debemos darle formato a ese archivo para que tenga la estructura de un archivo de espacio de intercambio. Esto lo hacemos con el comando **mkswap** del siguiente modo:

```
mkswap -c /swap 8192
```

Nuevamente, se debe reemplazar **8192** por el tamaño que corresponda. Por último, habilitamos el espacio de intercambio para que se use de inmediato:

```
swapon /swap
```

## Usar make para automatizar tareas

Cuando se distribuye código fuente, generalmente se ofrece una manera de compilar todos los archivos de forma automática. En pocas palabras, podemos decir que el proceso de **compilación** de un programa consiste en pro-

cesar todos los archivos que lo componen y convertirlos en **archivos binarios objeto** que como están todos sueltos, no es posible ejecutarlos. Luego se realiza un proceso de **ensamblado** que consiste en juntar todos los archivos objeto en un gran **paquete binario ejecutable**. Ese binario se debe copiar a uno de los directorios de archivos binarios de Linux, como por ejemplo **/usr/bin**. Como realizar esta tarea a mano es muy tedioso, se utiliza una herramienta que se llama **GNU Make** que tiene como objetivo principal automatizar comandos. Para esto, lo que hace es manejarse con un archivo de texto llamado **Makefile** que tiene la siguiente estructura:

```
tarea: dependencias
      comandos
tarea2:
      comandos
tarea3:
      comandos
```

Lo que hacemos es crear tareas que pueden tener dependencias unas de otras. Cada tarea tiene su set de comandos para ejecutarse. Veamos un ejemplo del mundo real:

## REEMPLAZO DE BASH

Si nos gusta mucho **Midnight Commander**, podemos utilizarlo como reemplazo de bash. Para esto, sólo debemos reemplazar el último campo de la línea correspondiente a nuestro usuario (**/bin/bash**) por **/usr/bin/mc**. De esta manera, cada vez que ingresemos al sistema, **Midnight Commander** aparecerá ante nosotros.

```
k journald starting. Commit interval 5 seconds
EXT3-fs: mounted filesystem with ordered data mode.
VFS: Mounted root (ext3 filesystem) readonly.
Freeing unused kernel memory: 328k freed
EXT3 FS on hda1, internal journal
i8xx TCO timer: heartbeat value must be 2<heartbeat<39, using 30
i8xx TCO timer: initialized (0x1060). heartbeat=30 sec (nowayout=1)
Intel 810 + AC97 Audio, version 1.01, 12:16:32 Oct 24 2006
PCI: Setting latency timer of device 0000:00:1f.5 to 64
i810: Intel ICH2 found at 10 0x7400 and 0x7000, MEM 0x0000 and 0x0000, IRQ 9
ne2k-pci.c:v1.03 9/22/2003 D. Becker/P. Gortmaker
http://www.scyld.com/network/ne2k-pci.html
i810_audio: Audio Controller supports 2 channels.
i810_audio: Defaulting to base 2 channel mode.
i810_audio: Resetting connection 0
ac97_codec: AC97 Audio codec, id: ALG16 (ALC200/200P)
i810_audio: AC'97 codec 0 supports AMAP, total channels = 2
i810_audio: setting clocking to 20454
agpgart: Detected an Intel i815 Chipset.
agpgart: AGP aperture is 64M @ 0xe0000000
eth0: RealTek RTL-8029 found at 0x4c00, IRQ 10, 00:40:00:70:24:9A.
ts: Compaq touchscreen protocol output
Adding 522104k swap on /System/Kernel/Devices/hda2. Priority:1 extents:1 across
:522104k
```

Figura 1. Si agregamos comandos al inicio, es bueno precederlos de un mensaje informativo con el comando **echo [mensaje]**.

```
copiar: creardir
cp archivo.txt
/etc/configuracion
creardir:
mkdir /etc/configuracion
```

Cuando se tipea **make**, el intérprete abrirá el archivo **Makefile** y ejecutará la primera tarea del archivo (**copiar**). ¿Tiene dependencias? Sí, depende de **creardir** (definido justo después de los dos puntos). Entonces ejecuta primero **creardir**. La tarea **creardir** lo único que hace es ejecutar un comando **mkdir** que crea un directorio. Luego, al finalizar, se ejecuta la tarea original (**copiar**), que copia un archivo en el directorio que fue creado por la tarea anterior. Cabe aclarar que también es posible poner múltiples dependencias por tarea y éstas se ejecutarán en el orden en que sean definidas. De esta forma, podemos utilizar los archivos **Makefile** para crear scripts que ejecuten diferentes tareas administrativas de nuestro sistema.

## Sugerencias para recompilar el núcleo

La mayoría de los usuarios Linux alguna vez se enfrenta con la necesidad de compilar un núcleo (Figura 2). Aquí tenemos una receta ligeramente modificada para llevar a cabo esta tarea:

```
make menuconfig
make dep
make clean
make zImage
cp arch/i386/boot/zImage/boot/
vmlinuz
```

```
cp System.map /boot
make modules
make modules_install
```

Podemos evitar los pasos **make zImage** y **cp** si editamos el archivo **Makefile** que hay en **/usr/src/linux**. Dentro de ese archivo buscamos:

```
#INSTALL_PATH=/boot
```

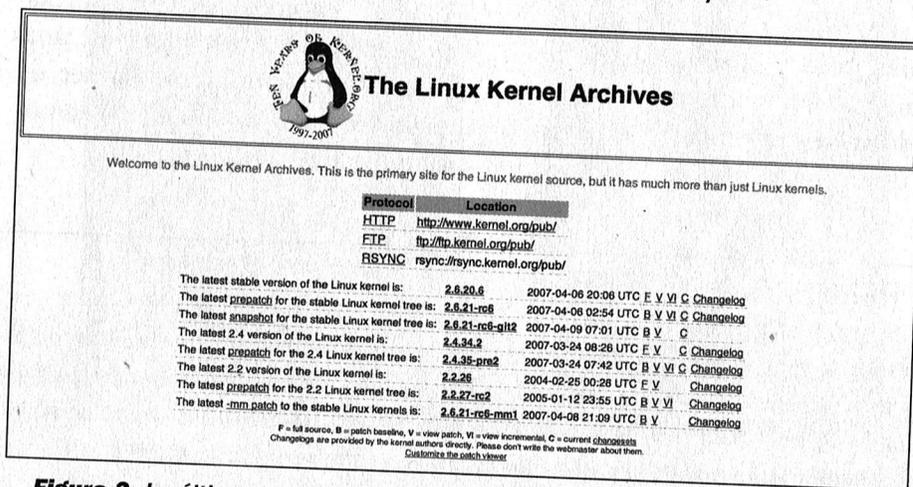


Figura 2. La última versión de kernel se puede descargar de [www.kernel.org](http://www.kernel.org).

Allí debemos quitar el símbolo de comentario a la línea. También podemos poner nuestro nuevo kernel en cualquier directorio, siempre y cuando definamos en nuestro gestor de arranque la ruta correspondiente. Una vez hecho esto, usamos **make zilo** (en reemplazo de **make zimage** y los **cp**) que hará todo el proceso de compilación y copiado de manera automática.

## Cómo crear nuestro propio repositorio de paquetes Debian

Es muy útil contar con repositorios Debian locales para la instalación de aplicaciones de forma más fluida. Un método muy útil es, justamente, copiar los paquetes .DEB de un CD a un directorio local para tenerlos disponibles en cualquier momento y desde cualquier equipo de la red. Para esto, lo primero que debemos hacer es copiar todo el contenido de los paquetes en un directorio específico como el siguiente:

```
/home/ftp/pub/debian/dists/repo/binary-i386.
```

Para ello se debe hacer:

```
$ cd /home/ftp/pub/debian/repo
$ dpkg-scanpackages
dists/repo//binary-i386
/dev/null > dists/repo/binary
i386/Packages
$ gzip -c repo/binary
i386/Packages > repo/binary
i386/Packages.gz
```

Luego de realizar los comandos anteriores obtendremos el archivo **Packages.gz**

(archivo necesario para acceder a cualquier repositorio .DEB) y ¡Listo! Para usarlo, sólo resta incluir la línea correspondiente en el archivo **/etc/apt/sources.list**. Editamos este archivo con un editor de texto cualquiera e incluimos la siguiente línea:

```
deb ftp://mimaquina.midominio.com/
pub/debian binary-i386 repo
```

Ya podemos utilizar el comando **apt-get install** para instalar los paquetes incluidos en el repositorio.

## Grabar canciones MP3 en un CD de audio desde la consola

Para poder grabar archivos MP3 en un CD de audio debemos contar con los siguientes programas: **sox**, **cdrecord** y alguno de los siguientes: **mpg123**, **freeamp** o **alsaplayer**. Como primera medida, debemos convertir el formato MP3 a un formato que el programa **cdrecord** pueda entender para después grabarlo:

```
mpg123 -s archivo.mp3 | sox -t raw
-c 2 -r 44100 -w -s - track.cdr
```

Una vez ejecutada esta línea, tendremos un archivo **track.cdr** en formato CD-audio (frecuencia de 44KHz, estéreo, sin signo). Ahora sólo debemos grabarlo en un CD:

```
cdrecord -nofix -audio track.cdr
```

Repetimos el mismo procedimiento para cada archivo MP3. Cuando haya

mos terminado de copiar todos los archivos, ejecutaremos:

```
cdrecord -fix
```

Ahora podremos escuchar las canciones que antes estaban en formato MP3 en un reproductor de CD hogareño.

## Borrar los volcados de memoria Core

Los archivos **core** son los que genera el sistema cuando alguna aplicación o servicio se cierra de forma inesperada. Hasta ahora, sólo teníamos la opción de

### III SIN CTRL + ALT + SUPR

En cualquier instalación estándar de Linux, si un usuario presiona la combinación de teclas **CTRL+ALT+SUPR**, el sistema se reiniciará de forma automática. Para evitar que esto suceda, debemos editar el archivo **/etc/inittab** y comentar (agregar un símbolo #) la siguiente línea:

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Si queremos que sólo ciertos usuarios puedan reiniciar el sistema de esta forma, definimos la línea de la siguiente manera:

```
ca::ctrlaltdel:/sbin/shutdown -a -t3 -r now
```

Y creamos un archivo llamado **/etc/shutdown.allow** que contenga el listado de usuarios que pueden reiniciar el sistema.

buscarlos con el comando **find** o desde el entorno gráfico y borrarlos de forma manual, una tarea quizás arriesgada e incómoda para cualquier usuario.

Para esto tenemos un pequeño script que se encargará de buscar esos archivos y después borrarlos sin preguntas. Sería interesante que se ejecutara de forma periódica en nuestro sistema Linux favorito. Como **root**, vamos a crear un archivo vacío llamado **limpieza.sh**.

```
# touch limpieza.sh
```

Luego tenemos que editarlo de la siguiente manera:

```
# vi /limpieza.sh
```

E ingresamos el siguiente texto:

```
#!/bin/sh
# Borrar archivos core
find / -name "core.*" -exec rm {} \;
```

Guardamos el archivo con la combinación de teclas **ESC + X** y salimos, y luego le brindamos los permisos suficientes para la ejecución:

```
# chmod 775 limpieza.sh
```

Listo. Ahora sólo debemos ejecutarlo anteponiendo el **./**.

```
# ./limpieza.sh
```

De esta manera, se limpiarán los archivos **core** de todo nuestro sistema. Para definirlo como un servicio más dentro del directorio **init.d**, sólo debe-

mos ingresar el texto siguiente y guardarlo dentro de algún **run level**.

```
#!/bin/sh
# Borrar archivos core

case "$1" in
  'start')
    find / -name "core.*" -exec rm
      {} \;
    ;;
  'stop')
    ;;
  *)
    echo "Usage: $0 { start | stop
      }"
    ;;
esac
exit 0
```

De esta forma, con tan sólo ingresar el comando **/etc/init.d/limpieza start**, el script se pondrá en acción y podremos hacer que éste se ejecute cada vez que se inicie el sistema.

## Cómo ejecutar aplicaciones de MS-DOS bajo Linux

Lo primero que debemos hacer es descargar dos paquetes. Uno corresponde al código fuente de **DOSEMU** (una aplicación que permite ejecutar un entorno completo de DOS bajo Linux) y el segundo corresponde a un sistema operativo **FreeDOS** (una versión GPL de DOS) lista para utilizar. Para eso, nos dirigimos a **www.dosemu.org** (Figura 3) y en la sección **SourceForge project page** descargamos los paquetes **dosemu** y **dosemu-freedos**.

Los archivos que debemos descargar no ocupan mucho lugar, por lo que la descarga será muy rápida. Luego debemos descomprimir ambos paquetes:

```
tar dosemu-1.2.2.tgz
```

```
tar -zxvf dosemu-freedos-b9r5a-bin.tgz
```

Antes de continuar, es importante destacar que es recomendable usar **DOSEMU** bajo el entorno gráfico **X11** si es que va-

DOSEMU.org

| Home | Developer Releases | Stable Releases | Documentation | SourceForge project page

**DOSEMU Main Page**

DOSEMU stands for DOS Emulation, and allows you to run DOS and many DOS programs, including many DPMI applications such as DOOM and Windows 3.1, under Linux.

For problems try [linux-mtdos@open.kernel.org](mailto:linux-mtdos@open.kernel.org) (see [mailing lists](#)) or the links on the left hand side of this page.

**NEWS NEWS NEWS**

DOSEMU 1.4.0 is out and can be obtained [here](#).

**SOURCEFORGE.NET**

Last modified: Sat May 5 13:13:23 EDT 2007

Figura 3. Podemos descargar DOSEMU y FreeDOS de [www.dosemu.org](http://www.dosemu.org).

mos a ejecutar juegos como **DOOM** o aplicaciones como Microsoft **Windows 3.1**. Y para que DOSEMU funcione bajo X11, debemos tener instaladas las librerías de desarrollo de X11 (incluidas en los CDs principales de todas las distribuciones de Linux). Por lo tanto, antes de continuar, hay que buscar esos CDs e instalar los paquetes de desarrollo bajo X11. Luego de hacerlo, vamos a compilar DOSEMU. Para esto, entramos al directorio **dosemu-1.2.2** y tipeamos:

```
./configure --prefix=/usr/local/dosemu
make
make install
```

El parametro **--prefix** nos permite indicar el directorio de instalación, y podemos cambiarlo por aquel que nos quede más cómodo. El proceso completo puede tomar varios minutos según el equipo que tengamos. Ahora debemos copiar el sistema operativo FreeDOS al directorio de instalación de DOSEMU. Para esto, salimos del directorio de fuentes de FreeDOS e ingresamos en el directorio **dosemu/** que creó el paquete **dosemu-freedos**. Allí encontraremos un directorio llamado **freedos/** que debemos mover con el siguiente comando:

```
mv freedos /usr/local/dosemu
```

Una vez que está todo copiado en su lugar, procedemos a configurar algunos parámetros de DOSEMU para que funcione correctamente en nuestro equipo. El archivo de configuración de DOSEMU se encuentra en **/etc/dosemu/dosemu.conf**. Como todo archivo de configuración de Linux, las líneas que aparecen con un símbolo **#** corresponden a líneas deshabilitadas, por lo que deben ser eliminados si queremos habilitar alguna de las opciones que vamos a mencionar a continuación:

```
$_cpu = "80386"
```

En esta cláusula podemos definir el tipo de **procesador** que tenemos. Si tenemos un equipo Pentium o superior, podemos reemplazar el valor por 80586.

```
$_hdimage = "/usr/local/dosemu/freedos"
```

Quizás éste sea el parámetro más importante del archivo. Aquí debemos definir el directorio en el que está alojado el sistema operativo (esto es, la ubicación del directorio de FreeDOS). En caso de tener una partición con un MS-DOS instalado, podemos indicarla aquí (por ejemplo **/dev/hda1**) y DOSEMU booteará automáticamente ese sistema.

```
$_xms = (8192)
```

Este parámetro nos permitirá indicar la cantidad de MB de memoria extendida que tendrá nuestro sistema DOS.

```
$_ems = (2048)
```

Lo mismo para la memoria expandida (EMS) que tantos programas utilizaban durante los años 90.

```
$_layout = "auto"
```

Este parámetro intentará detectar nuestro teclado y utilizar su configuración de teclas bajo DOSEMU. Si no funciona, podemos indicar el lenguaje. Por ejemplo, el parámetro para el teclado en español es **es**.

```
$_com1 = "/dev/ttyS1 irq 5"
```

Si tenemos un dispositivo conectado a un puerto serie (como por ejemplo, un

módem externo), debemos definirlo aquí. Por ejemplo, si tenemos un módem conectado al puerto serie **/dev/ttyS1** debemos definirlo tal como está en el ejemplo.

Lo que vimos hasta ahora es la configuración básica, pero hay muchas más opciones para personalizar la placa de video, la placa de sonido, la de red, etcétera. Todo viene preconfigurado para que funcione de fábrica, por lo que no deberíamos tener problemas.

Por último, debemos ubicarnos en el directorio **/usr/local/dosemu/bin** y ejecutar el archivo **dosemu** o **xdosemu** si estamos bajo el modo gráfico. Ahora, sólo es cuestión de navegar por el sistema de archivos y ejecutar el programa que más nos guste (**Figura 4**). Si tenemos disquetes o CDs con programas de DOS y queremos ejecutarlos bajo DOSEMU, sólo tenemos que copiarlos al directorio del paquete FreeDOS (**/usr/local/dosemu/freedos**).

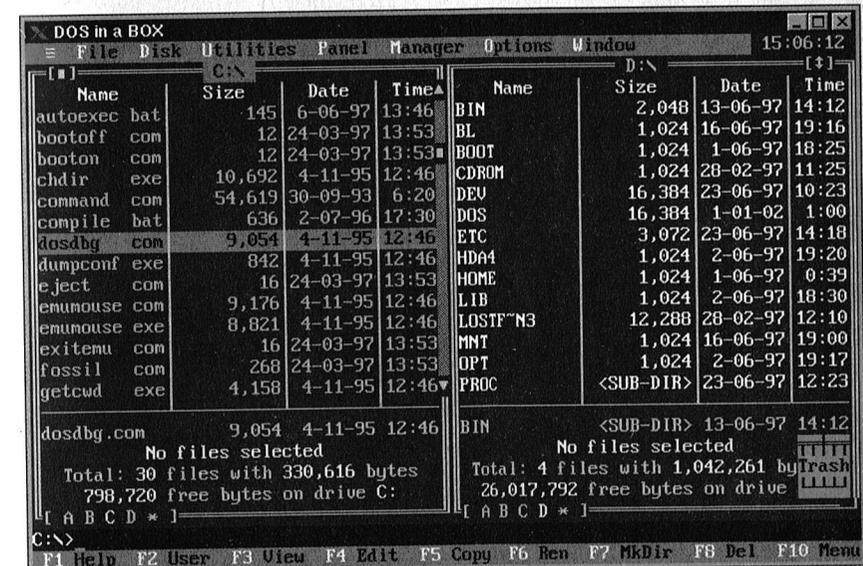


Figura 4. Ejecución de Norton Commander bajo una terminal de DOSEMU en Linux.

## III CÓMO MONTAR UNA IMAGEN ISO

No hace falta grabar una imagen ISO en un CD para ver su contenido. ¡Podemos montarla directamente desde la consola! Para esto, debemos usar el comando **mount** del siguiente modo: **mount -o loop archivo.iso /mnt/cdrom**.

## Cómo sincronizar una Palm con Linux desde la consola

Esta forma de comunicar nuestro dispositivo Palm es muy efectiva y por demás de rápida, y sólo tendremos que acordarnos los comandos y su sintaxis. Pero antes tendremos que configurar la conexión. Esto se logra creando un enlace simbólico entre nuestro dispositivo Palm y el puerto USB o serial (según el modelo de Palm). Por lo general, los dispositivos más modernos se conectan a nuestra PC mediante cables USB, y de esta forma vamos a enlazar el dispositivo a nuestro puerto disponible.

Es bueno verificar si nuestro sistema operativo detecta el dispositivo USB, y para esta operación tenemos dos comandos: **lspci** y **lsusb**. Cualquiera de estas opciones nos lista los dispositivos activos y listos para ser configurados que hay conectados a nuestra PC. Como veremos, no sólo está la información de nuestro dispositivo USB, sino que también están todos los dispositivos de nuestro equipo. Pero nos resta algo por demás de importante: saber cuál es el archivo dispositivo correspondiente. Y para esto tenemos dos opciones:

- Ingresar al lugar donde están los archivos de dispositivos de nuestro sistema y buscarlo nosotros mismos (directorio **/dev**).
- La otra y más fácil es (siempre como usuario **root**) tipear la siguiente línea una vez que nuestro dispositivo Palm quiera sincronizar.

```
# lsusb
```

```
Bus 003 Device 001: ID 0000:0000
```

```
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 003: ID 0830:0070
Palm, Inc.
Bus 001 Device 002: ID 04b8:080e
Seiko Epson Corp.
Bus 001 Device 001: ID 0000:0000
```

Con esta información veremos cuáles son exactamente los dispositivos de nuestro puerto USB. De esta manera, vamos a crear un enlace directo:

```
# ln -s /dev/dev/ttyUSB1 /dev/pilot
```

Ahora podemos hacer uso de las herramientas clásicas de la consola gracias a la aplicación **pilot-link** (que podemos descargar de <http://sourceforge.net/projects/pilot-link>).

Las utilidades de esta aplicación son:

- Backup de todo nuestro dispositivo: **pilot-xfer /dev/pilot -r backpalm**
- Instalación de nuevas aplicaciones: **pilot-xfer /dev/pilot -i programita.prc**
- Listar todos los programas de nuestro equipo: **pilot-xfer /dev/pilot -L**

Para ver ayuda sobre estas herramientas, podemos ver las páginas del manual del comando **pilot** con **man pilot**.

## Cómo compartir la conexión a Internet en pocos pasos

El sistema NAT (*Network Address Translation*) permite que muchos clientes de una red tengan acceso a Internet utilizando sólo una IP real (la que asig-

na el proveedor). La cuestión es la siguiente: entre nuestra red e Internet vamos a instalar un **servidor NAT**. Ese servidor NAT estará compuesto por dos placas de red. Una de ellas estará conectada a Internet y la otra a nuestra red privada. Ahora bien, la función de este NAT será la de tomar todos los paquetes que provengan de la red privada (que tiene direcciones IP privadas que no son válidas para Internet), **enmascararlos** en un solo paquete y enviarlos a Internet. Cuando las peticiones vuelven, el NAT **desenmascara** el resultado y lo distribuye entre las máquinas que hicieron las peticiones. Lógicamente, este paquete enmascarado sale mediante la placa que está conectada a Internet y posee una dirección IP real.

Es importante mencionar que antes debemos tener las dos placas de red configuradas correctamente y una de ellas debe tener la conexión a Internet ya esta-

blecida. Hay varias maneras posibles de conectar una placa de red a una conexión de banda ancha y esto varía según el tipo de sistema que se utilice para acceder a Internet (cablemódem, ADSL, etcétera), por lo que no vamos a entrar en detalles sobre este tema y nos dedicaremos a configurar el NAT.

Supongamos que tenemos una red con direcciones IP privadas de tipo **10.0.0.0/24** (esto es, una red clase C). La referencia a Internet no se hace con la IP real ya que ésta generalmente cambia. Lo que se debe hacer es referenciar Internet como **0.0.0.0/0**. Comencemos entonces por vaciar la tabla de cadenas del **firewall** y del NAT. El comando que vamos a utilizar es **iptables**, que viene incluido con todas las distribuciones de GNU/Linux. **Iptables** permite tener varias reglas en diferentes tablas. La tabla NAT es la tabla en donde se definen las reglas de enmascaramiento. En ella debemos crear una

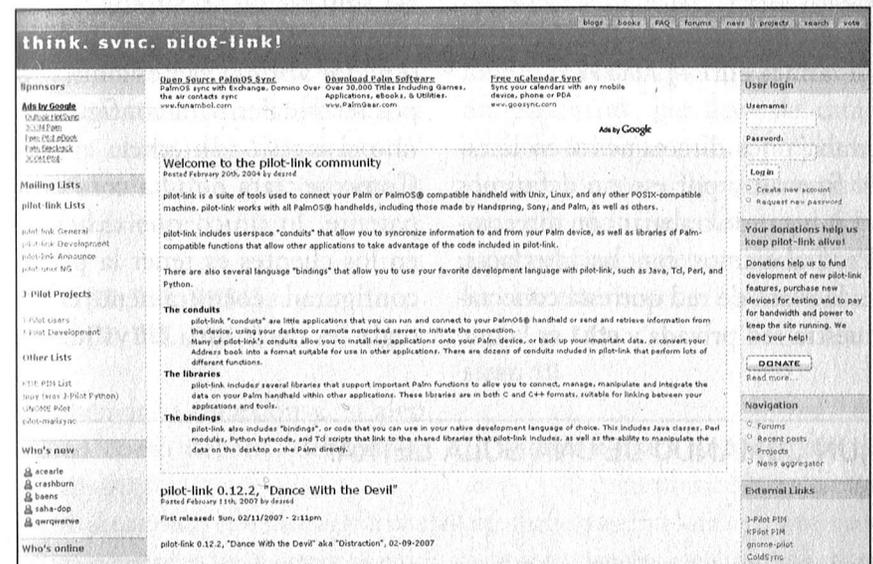


Figura 5. En el sitio de pilot-link ([www.pilot-link.org](http://www.pilot-link.org)) podemos obtener información sobre esta aplicación.

regla que defina que todo lo que sale hacia fuera sea enmascarado. Para eso usamos el siguiente comando:

```
iptables -t nat -A POSTROUTING -p tcp -s 10.0.0.0/24 -d 0.0.0.0/0 -j MASQUERADE
```

Analícemos un poco el comando anterior. Con el primer set de parámetros (-t nat -A POSTROUTING) definimos que vamos a crear una regla para la tabla NAT en la cadena POSTROUTING (todo lo que sale). Luego definimos el protocolo tcp (con -p), las redes fuente (-s) y destino (-d) y finalmente la acción (-j): enmascarar.

Bien, ya tenemos gran parte del problema resuelto. Ahora, en la tabla del firewall debemos crear una regla para permitir el paso entre las dos redes. Esto se llama **forwarding** y se hace del siguiente modo:

```
iptables -A FORWARD --in-interface eth0 --out-interface eth1 -j ACCEPT
```

Aquí trabajamos directamente en la tabla del firewall y por eso no definimos tabla. En lugar de trabajar con direcciones IPs, trabajamos con las interfaces: **eth0** es la placa de red que está conectada a nuestra red privada y **eth1** es la que

está conectada a Internet. En resumen, acabamos de definir dos cosas: la primera de ellas es que todo lo que sale de nuestro NAT sea enmascarado. La segunda fue la de permitir el paso de paquetes entre la red privada e Internet. Bien, como último paso, en el servidor debemos activar el soporte de IP forwarding en el kernel. Esto lo hacemos con el siguiente comando:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Cabe destacar que todos estos comandos no quedan almacenados en ningún archivo, por lo que cada vez que reiniciemos la computadora tendremos que escribirlos si queremos compartir nuestra conexión a Internet. Una buena solución es armarnos un script con todos estos comandos y que ese script se ejecute al iniciar GNU/Linux de forma automática (ya hemos visto cómo realizar esto en un truco anterior). Ya hemos terminado nuestro trabajo en el servidor y ahora vamos a nuestros equipos terminales para configurar en cada uno el acceso a Internet.

Como se trata de una conexión transparente, lo único que debemos hacer en los clientes es tener la placa de red configurada correctamente con la IP en regla (es decir, una IP válida en la red

local) y configurar la dirección del servidor NAT con el comando **route**. Si usamos Windows (o cualquier otro sistema operativo en los clientes), entonces tendremos que ir a la configuración del entorno de red y definir la IP de nuestro nuevo **router** como **Gateway** (o pasarela). Pero volviendo a GNU/Linux, para definir un Gateway usamos el siguiente comando:

```
route add default gw [IP_DEL_SERVIDOR]
```

Si analizamos el comando, veremos que lo que acabamos de hacer es crear una ruta por defecto al gateway (pasarela) en la dirección definida para el servidor. Sencillo, ¿no?

Pero todavía nos falta algo muy importante: definir las direcciones de los servidores de nombres en cada una de las computadoras cliente. Si en el servidor NAT la conexión a Internet funciona correctamente, entonces podemos tomar el archivo **/etc/resolv.conf** y copiarlo en la misma ubicación de cada uno de los clientes. Éste es el archivo en el cual se ubican las direcciones IP de los servidores DNS y el contenido debe ser el siguiente:

```
nameserver [IP_DEL_DNS1]
nameserver [IP_DEL_DNS2]
```

¡Listo! Probemos la conexión y, si algo no funciona como debería, deberemos revisar lo siguiente:

- Asegurarnos de que el router esté conectado correctamente a Internet.
- Asegurarnos de que **eth0** sea la placa de red que está conectada a nuestra



Es muy útil crear nuestros propios scripts de bash para no tener que tipear una y otra vez los comandos de configuración del sistema. Estos scripts deben estar almacenados en **/usr/sbin** ya que allí se almacenan los scripts y comandos que sólo puede utilizar el usuario **root**. En el Capítulo 8 incluimos un montón de trucos y sugerencias para hacer scripts más potentes y efectivos.

red local y **eth1** sea la que está conectada a Internet (podemos invertir los cables si hace falta).

- Asegurarnos de estar escribiendo los comandos correctamente.

## Multitarea en el modo texto

Una forma de aprovechar la multitarea de Linux es ejecutando procesos en **background** o segundo plano. El método no es complicado y son muchos sus beneficios. Para ejecutar un programa de fondo, sólo hay que tipear su nombre y un parámetro que lleva de carácter el símbolo **&**. Hagamos una prueba sencilla: el comando **sleep** lleva como parámetro un valor numérico. Una vez ejecutado, espera el tiempo determinado por ese valor. De este modo, si ejecutamos:

```
sleep 10
```

El sistema entrará en un estado de pausa por 10 segundos. Ahora ejecutemos este mismo comando pero en **background**. Para eso debemos tipear:

```
sleep 10 &
```

## III ¡UN COMANDO DE UNA SOLA LETRA!

El comando **w** nos muestra en pantalla un listado de los usuarios conectados al sistema y bastante información adicional sobre ellos, como la terminal, el tiempo de conexión, el proceso que tienen en ejecución, etcétera.

Y automáticamente el sistema imprimirá el siguiente mensaje:

[1] 638

[root@myhost /root]#

En la primera línea tenemos dos valores: el primer valor nos indica el número de trabajo que se está ejecutando en background. El segundo valor nos indica su **pid** (*process id* o identificación del proceso). En la segunda línea, se devuelve el **prompt** para poder ejecutar otros comandos.

Una vez concluido el proceso, el sistema imprimirá el siguiente mensaje después de presionar la próxima tecla **ENTER**:

[1]+ Done sleep 10

Esto significa que el proceso ha concluido. Para obtener un listado de los procesos que se están ejecutando actualmente en background, sólo hay que tipear el comando **jobs**. La salida de este programa será un listado de procesos semejante al siguiente:

```
jobs
[1]- Running sleep 20 &
[2]+ Running sleep 30 &
```

Esta tabla contiene tres columnas: la primera muestra el número de proceso en background, la segunda el estado (**Running**: ejecutándose, **Stopped**: parado) y la tercera la línea la cadena correspondiente al proceso en cuestión.

Si queremos enviar al fondo un proceso que se está ejecutando en este momento, debemos utilizar la combinación de teclas **CTRL + Z**. El resultado de esta combinación será la siguiente cadena impresa en nuestra pantalla:

[2]+ Stopped sleep 70

Esto nos indica que el trabajo está parado en el fondo pero que no ha sido destruido y puede ser reanudado en cualquier momento. Para traerlo al frente de nuevo, debemos utilizar el comando **fg** seguido de un signo **%** y el número de trabajo. En el caso de nuestro proceso **sleep**, la línea completa es:

fg %2

Por su parte, la combinación de teclas **CTRL + Z** envía un proceso al fondo pero lo detiene. Para reanudarlo (en background), deberemos utilizar el comando **bg** con el mismo formato que el comando **fg**.

## Cómo acceder desde Ubuntu a un directorio compartido en red

Muchas veces nos encontramos con la necesidad de acceder a un servidor remoto que tiene compartido algún tipo de archivo que necesitamos de forma diaria. La manera más común es crear un acceso directo en nuestro escritorio, pero claro, la forma de hacerlo en Linux y más precisamente en Ubuntu no es tan simple como copiar y pegar. Acá tenemos que crear un punto de montaje donde luego se puede acceder a ese **recurso compartido**, como normalmente se denomina en informática.

Para ello tenemos dos opciones. Una de ellas es editar un archivo de texto ubicado en **/etc** llamado **fstab**, donde debemos ingresar parámetros bastante complejos que quizás un nuevo usuario de

Linux no comprenda bien. La otra opción (bien gráfica) es mediante la acción incluida en la barra de herramientas central bajo el nombre de **Lugares -> Conectar con el servidor**. Ahí podremos conectarnos con nuestro servidor Windows o Linux donde tenemos los archivos tan preciados ingresando datos muy simples: en primer lugar la **dirección IP** de nuestro servidor (el nombre podría funcionar en algunos casos), luego el **directorío** donde se encuentra el subdirectorío al que se desea ingresar seguido de la carpeta, el nombre de usuario, el grupo de trabajo y por último el nombre que deseamos que se vea reflejado en nuestro escritorio. ¡Listo! Aceptamos una vez ingresados los datos y el montaje estará activo: ya podemos trabajar sobre éste como si estuviéramos parados en un directorio local.

```
[root@defiant /mnt]# more /etc/fstab
LABEL=/ / ext2 defaults 1 1
LABEL=/boot /boot ext2 defaults 1 2
LABEL=/home1 /home ext2 defaults 1 2
/dev/cdrom /mnt/cdrom iso9660 noauto,owner,ro,intr 0 0
/dev/cdrom1 /mnt/cdrom1 iso9660 noauto,owner,ro 0 0
/dev/fd0 /mnt/floppy auto noauto,owner 0 0
LABEL=/usr /usr ext2 defaults 1 2
LABEL=/var /var ext2 defaults 1 2
none /proc proc defaults 0 0
none /dev/pts devpts gid=5,mode=620 0 0
/dev/hdb1 swap swap defaults 0 0
defiant /export/RPMS /mnt/nfs nfs bg,soft,intr,retry=5 0 0
```

Figura 6. La edición del archivo **/etc/fstab** es compleja y lleva más tiempo que utilizar la opción gráfica para acceder a un recurso compartido.

## III PROCESOS QUE CONVIENE EJECUTAR EN SEGUNDO PLANO

El segundo plano es un ámbito de ejecución ideal para las aplicaciones que llevan a cabo procesamientos independientes y no necesitan de la intervención del usuario. En cambio, aquellos programas que hacen preguntas o que requieren que el usuario esté constantemente tomando decisiones no son aptos para ejecución en segundo plano ya que cuando corren en ese plano no se los puede ver.



- 1 Podemos eliminar la posibilidad de ingresar al sistema en modo monousuario (**linux single**, en LILO) editando el archivo `/etc/lilo.conf`.
- 2 Si queremos registrar en un archivo los mensajes de error que brinda un comando, podemos utilizar el operador de redirección `2> [archivo]` al final del comando.
- 3 Es posible realizar búsquedas de una cadena en un archivo de texto con el comando `cat [archivo] | grep [cadena_a_buscar]`.
- 4 Todos los scripts controladores de la actividad de los servicios del sistema se encuentran en `/etc/init.d`. Allí podemos controlar su actividad con los parámetros **start/stop/restart** (por ejemplo: `/etc/init.d/samba start`).
- 5 Con sólo poner un script con permisos de ejecución en el directorio `/etc/cron.daily` (**weekly** o **monthly**) podremos hacer que ese script se ejecute automáticamente gracias al cronómetro del sistema.
- 6 Si vamos a compilar el núcleo, siempre es bueno mantener una copia de la versión anterior para evitar que algún problema de configuración en el núcleo nuevo nos deje el sistema inutilizable.
- 7 Los repositorios de paquetes en Debian y sus derivados (como Ubuntu) se encuentran en `/etc/apt/sources.list`.
- 8 Para ver si nuestro equipo tiene reglas de firewall definidas podemos usar el comando `iptables -L`.
- 9 Con el comando `ps -u` podemos ver un listado de procesos de usuario.
- 10 Es muy fácil instalar las herramientas de desarrollo en una distribución Debian o Ubuntu con el comando `apt-get install build-essential`.

# Trucos para la consola del administrador

¡Más trucos para la consola de Linux!

En este apartado incluimos aquellos que consideramos de suma utilidad para los administradores de sistemas.

Cómo hacer que todos los usuarios puedan montar la unidad de CDROM	68
Definir límites de uso de disco para los usuarios	68
Cambiar el intérprete de comandos de cada usuario	71
Cómo deshabilitar un usuario temporalmente	71
Aumentar la seguridad en NFS	72
Hacer copias de respaldo con <code>pdumpfs</code>	73
Cómo bloquear el ping	74
Screen: una herramienta para administradores	74
Cómo crear una cuenta de usuario sin contraseña	75
Cómo controlar la actividad de la impresora desde la consola	75
Usar Netcat como escaneador de puertos	75
Backup diferenciales con <code>rdiff</code>	76
Realizar un backup de todo nuestro sistema	76
Cómo evitar los rootkits	77
Cómo poner la placa de red en modo promiscuo	78
Enviar y recibir archivos entre equipos de una red con <code>nc</code>	78
Cómo verificar la integridad del sistema de archivos	79
Cómo configurar múltiples IPs en la misma placa de red	79
Transferencia encriptada de archivos vía SSH	80
Recuperar la password de administrador en MySQL	81

## Cómo hacer que todos los usuarios puedan montar la unidad de CDROM

Por lo general, éste es un problema clásico de todas las distribuciones Linux. Esto es porque `/dev/cdrom` es un enlace de `/dev/hdc` o de `/dev/hdb` y esos archivos son propiedad del administrador.

Si nosotros queremos que cualquier usuario pueda montar y desmontar el CDROM, primero tendremos que saber cómo tenemos configurado nuestro dispositivo CDROM. Esto lo podemos consultar de una forma muy simple: `vdircdrom` o `ls -l /dev/cdrom`.

Luego nos fijamos que en el archivo de configuración `/etc/fstab` esté configurado no con el enlace sino con su **ubicación real** (el dato que nos dio el `vdircdrom`). Ahora sólo resta asignarle permisos para que cualquier usuario pueda hacer uso del CDROM. Para eso ingresamos:

```
# chmod a+rwX /dev/cdrom
# chmod 666 /dev/cdrom
# chmod a+rwX /dev/hdb (/dev/hdc)
```

A partir de ahora, todos los usuarios del sistema podrán montar la unidad de CDROM sin problemas.

## Definir límites de uso de disco para los usuarios

El sistema de **quotas** de disco permite definir límites de uso de particiones por usuario. Esto significa que un determinado usuario puede tener la posibilidad de escribir una N cantidad de megabytes

## III ALIAS PARA NO TIPEAR

Podemos crear **alias** para comandos muy largos que tipeamos frecuentemente del siguiente modo: **alias nombre="comando completo"**, por ejemplo, **alias dir="ls -l"**. De esta manera, cada vez que se escriba la palabra **dir**, en realidad se ejecutará el comando **ls -l** y se ahorrará tiempo.

en una partición, mientras que otro usuario puede tener un número mayor de megabytes disponibles o menor, según lo hayamos definido. Además de definir límites por cantidad de espacio a utilizar, el sistema de quotas nos permite definir límites de **inodos**. Esto es limitar la cantidad de archivos que se podrá almacenar por usuario y por partición. Para instalar este sistema, es absolutamente necesario trabajar con la cuenta de usuario del administrador (**root**).

Antes de poder utilizar el sistema de quotas, es necesario llevar a cabo ciertos preparativos. En primer lugar, tenemos que asegurarnos de que las utilidades **quota** estén instaladas en nuestro sistema. Para esto, podemos ejecutar:

```
# quota
```

Si el sistema nos devuelve un error de comando no encontrado, entonces debemos buscar el paquete **quota** correspondiente a nuestra distribución en los CDs de ésta. Los usuarios de Debian (y sus derivadas como Ubuntu) se contentarán al saber que con sólo ejecutar **apt-get install quota**, tendrán todos los comandos necesarios para utilizar el sistema.

Luego debemos editar el archivo `/etc/fstab` que contiene un listado de todas las particiones y los sistemas de archivos que se pueden montar en nuestro sistema. Veamos un ejemplo:

```
/dev/hda2 / ext2 errors=remount-ro 0 1
/dev/hda3 none swap sw 0 0
proc /proc proc defaults 0 0
/dev/fd0 /floppy auto user,
noauto 0 0
/dev/cdrom /cdrom iso9660
ro,user,noauto 0 0
/dev/sda1 /mnt/sda1 vfat
rw,user,noauto 0 0
/dev/hda4 /mnt/datos vfat
user,noauto 0 0
```

Lo que debemos hacer en este archivo es habilitar el sistema de quotas en las particiones que deseemos. Para esto, en la columna de opciones de la partición (la cuarta columna), debemos agregar una coma y la opción **usrquota** (para habilitar límites por usuarios) o **grpquota** (para habilitar límites por grupos). Entonces, para el archivo de ejemplo, si suponemos que queremos limitar la utilización del espacio por usuario a la partición `/dev/hda4`, la línea de configuración debería quedar del siguiente modo:

```
/dev/hda4 /mnt/datos vfat
user,noauto,usrquota 0 0
```

Una vez realizadas estas modificaciones, procederemos a desmontar y vol-

ver a montar las particiones con el sistema de quotas habilitado para que los cambios tengan efecto en el sistema.

El próximo paso consistirá en crear una base de datos quota en la que se almacenará la información del directorio (en el cual está montada la partición) relativa al espacio utilizado, el espacio libre, el espacio utilizado por usuario, la cantidad de inodos, etcétera. Para crear esta base de datos, utilizamos el comando **quotacheck** del siguiente modo.

```
quotacheck -cug [directorio]
```

El parámetro **c** le indica a **quotacheck** que queremos crear una nueva base de datos. Los parámetros **u** y **g** le indican a **quotacheck** que vamos a limitar por usuario y por grupos. Finalmente, **[directorio]** es el punto de montaje de la partición a la que vamos a aplicar los límites.

Una vez que se encuentra creada la base de datos, le vamos a pedir al comando **quotacheck** que complete esa base con datos iniciales. Esto lo hacemos con el siguiente comando:

```
quotacheck -vug
```

Ahora **quotacheck** se tomará un tiempo en completar la base de datos con información del tipo cantidad de espacio ocupado, cantidad de espacio ocupado por usuario, cantidad de espacio ocupado por grupo, cantidad de inodos, etcétera. En las particiones grandes y con mucha información, esto puede llevar un rato largo. Cuando este proceso finalice, ya tendremos todo nuestro sistema configurado con el sistema de quo-

tas y estaremos listos para imponer los límites que nuestros usuarios se merezcan. Para definir límites por usuario, vamos a utilizar el comando **edquota** seguido del nombre de usuario al cual vamos a configurar límites. Veamos:

### edquota pedro

Se mostrará en pantalla un editor de textos (que generalmente es **VI**) con un archivo que tenemos que editar. El archivo está estructurado en columnas de la siguiente forma:

- **FS:** sistema de archivos en cuestión.
- **Blocks:** cantidad de bloques utilizados.
- **Soft:** éste es el límite blando de cantidad de bloques.
- **Hard:** éste es el límite duro de cantidad de bloques.
- **Inodes:** cantidad de nodos utilizados.
- **Soft:** límite blando de inodos.
- **Hard:** límite duro de inodos.

La primera columna es la **partición**. La segunda columna es la cantidad de **espacio utilizado** en bloques. Luego se definen los límites de espacio a utilizar. El límite blando es un límite que cuando se alcanza se le advierte al usuario en pantalla. El límite duro es un límite que cuando se alcanza, el usuario ya no puede escribir nada más

en esa partición. Luego viene la columna **inodes**, que es la cantidad de archivos actualmente almacenados por el usuario y justo después, los límites blandos y duros de cantidades de archivos. Sencillo, ¿no?

Veamos un ejemplo. Si en la partición **/dev/hda1** queremos definir que el usuario **pedro** sólo pueda escribir 10MB y 100 archivos, la línea quedaría como la siguiente:

```
/dev/hda1 5000 8000 10000 3
70 100
```

En este caso, sabemos que el usuario actualmente ocupa 5000 bloques de la partición **/dev/hda1** y que tiene un límite blando de 8000 bloques y un límite duro de 10000 bloques. En cuanto a la cantidad de inodos, actualmente tiene tres archivos y posee un límite blando de 70 archivos y un límite duro de 100 archivos. Bien, habiendo entendido esto, procederemos a configurar los límites según nuestras necesidades. Una vez que terminamos, podemos guardar el archivo y volver al shell. Ahora, para verificar que el archivo haya sido correctamente configurado, utilizamos el comando **quota** del siguiente modo:

### quota [usuario]

Y nos mostrará la configuración actual. Si está todo bien, procedemos a habilitar los límites para la partición:

### quotaon [particion]

Y ahora estará todo funcionando. En cuanto a grupos, el proceso es igual salvo que para editar las cuotas debemos usar el comando **edquota** con el parámetro **-g**. Luego podemos utilizar el mismo parámetro del comando **quota** para verificar que los límites hayan sido correctamente definidos.

## Cambiar el intérprete de comandos de cada usuario

Cuando el usuario ingresa al sistema, el gestor de registración (el programa **login**) se encarga de ejecutar el intérprete de comandos configurado para ese usuario. Esto se define en el archivo **/etc/passwd** y es el último parámetro de cada registro de usuario (**Figura 1**). En el siguiente ejemplo, el intérprete de comandos para el usuario **facundo** es el famoso **Bash**.

```
facundo:x:500:100:facundo:/home/
facundo:/bin/bash
```

```
root@GoboLinux: ~]cat /etc/passwd
root:x:0:0:SuperUser:/Users/root:/bin/zsh
nobody:x:12:12:nobody:/tmp:
mail:x:8:12:mail:/:
news:x:9:13:news:/usr/lib/news:
uucp:x:15:15:uucp:/var/spool/uucp:
ftp:x:14:50:/:/var/ftp/tmp:
fibo:x:21:21:/:tmp:/bin/zsh
sshd:x:22:104:/:var/empty:/bin/false
hfarena:x:1000:1000:Facundo Arena,,,:/home/users/hfarena:/bin/bash
root@GoboLinux: ~]_
```

**Figura 1.** El archivo **/etc/passwd** contiene la información del intérprete de comandos de cada usuario.



Existe otro archivo que almacena información de los usuarios: **/etc/shadow**. Incluye las claves encriptadas y su estructura es similar a la de **passwd**.

En resumen, el último parámetro de cada registro de usuario en el archivo **passwd** corresponde al primer programa que se cargará justo después de que el usuario ingrese nombre de usuario y clave válidos para iniciar sesión.

## Cómo deshabilitar un usuario temporariamente

Suele suceder que el operador del sistema necesita deshabilitar un usuario, ya sea por mal comportamiento en el sistema, porque se va de vacaciones o porque no estará registrado por un largo período de tiempo. Una de las maneras de realizar esta operación es cambiándole el password (utilizando el comando **passwd**), aunque cuando este usuario intente registrarse en el sistema y note que su clave de acceso ha sido alterada sin razón alguna, podría llegar a sentirse ofendido.

## ||| LIMPIAR LA CONSOLA

¿Alguien se acuerda del comando **CLS** de MS-DOS? Para aquellos que no lo conocen, se trata de un comando que **limpiaba** toda la información de la pantalla. En Linux hay dos formas de hacer esto: con el comando **clear** o mediante la combinación de teclas **CTRL + L**.

Otra manera de realizar esto es cambiar su shell por un script que sólo muestre un mensaje. Un ejemplo podría ser el siguiente:

```
# !/usr/bin/tail +2
Ud ha sido temporariamente
deshabilitado. Por favor
comuníquese por vía
telefónica con el
administrador.
```

En este script, la primera línea ejecuta el comando **tail** con el parámetro **+2**. Este comando imprime en pantalla las dos últimas líneas del archivo en cuestión, que sólo son una cadena con un mensaje informativo.

## Aumentar la seguridad en NFS

El sistema NFS provee un sistema de seguridad basado en hosts confiables. ¿Qué quiere decir esto? Que el único filtro que nosotros como administradores disponemos para definir quién ingresa y quién no es utilizar el número de IP de cada máquina. Ahora bien, esto puede presentar algunos problemas.

El más importante de ellos es que desde un cliente monten un directorio nuestro como usuario **root**. Seguramente no queremos que esto suceda porque si alguien monta un directorio exportado como usuario **root** en el servidor, posee acceso total a él. Para solucionar esto, podemos utilizar el parámetro **root\_squash** en la definición del directorio exportado (en el archivo **/etc/exports**).

Veamos un ejemplo:

```
/directorio_publico
*(ro,root_squash)
```

En este caso, si alguien quiere montar el recurso llamado **directorio\_publico** como usuario **root**, éste será mapeado hacia el usuario **nobody**. Otra opción que tenemos es definir que automáticamente se mapee hacia el usuario **nobody** para cualquier usuario, lo que evita grandes problemas de seguridad en los directorios de sólo lectura. Veamos un ejemplo:

```
/directorio_publico
*(ro,all_squash)
```

De esta forma, usando **all\_squash**, cualquier usuario que monte este directorio tendrá los mismos privilegios que el usuario **nobody**. También podemos mapear hacia **nobody** cualquier usuario o rango de usuarios o cualquier grupo o rango de grupos. Esto lo hacemos con los parámetros **squash\_uids** y **squash\_gids**. Veamos unos ejemplos:

```
/usr/pub *(ro, squash_uids=0
50, squash_gids=0-50)
```

En este ejemplo mapeamos hacia el usuario **nobody** a todos los usuarios que tengan **UID** entre 0 y 50, y a todos los grupos que tengan **UID** entre 0 y 50. Lógicamente, podemos cambiar estos valores para que se adecuen a nuestras necesidades.

Una posibilidad relacionada con el mapeo de usuarios es la de mapear hacia un usuario que no sea **nobody**. Para eso usamos el parámetro **anonuid** (o **anongid** si queremos mapear el grupo también). El siguiente es un ejemplo de ello:

```
/para_compartir *(all_squash,
anonuid=1000,anongid=1000)
```

Como podemos ver, se mapea a todos los usuarios hacia el **UID 1000, GID 1000**. Esto es fácil de administrar cuando gestionamos el acceso de pocos clientes. Pero, ¿qué sucede cuando tenemos muchos? Realmente se hace muy difícil administrar el mapeo de usuarios. Por esta razón, NFS nos ofrece un sistema de definición de archivos de mapas por clientes. Así podemos definir toda una tabla de mapeo específica para cada host de nuestra red. Para esto, en el recurso a exportar debemos utilizar el parámetro **map\_static** [archivo]. Veamos un ejemplo:

```
/para_compartir maquina1
(map_static=/etc/nfs/maquina1.map)
```

Aquí decimos que al directorio **para\_compartir** sólo se podrá acceder desde el host llamado **maquina1**, que tendrá su mapa de usuarios en el archivo **/etc/nfs/maquina1.map**. Ahora, veamos un ejemplo de ese mapa:

```
gid 0-50 -
uid 100-200 1000
```

## SE COLGÓ LA TERMINAL

A veces sucede que la terminal se llena de caracteres raros por el mal funcionamiento de algún programa y se vuelve inutilizable. Para revertir esta situación molesta, podemos usar el comando **reset**. Si no está en el sistema, podemos crear este programa hecho en C. Para eso, debemos crear un archivo de texto y escribir el siguiente código:

```
#include <stdio.h>
main(){
    printf("\033c");
    printf("\033[2j");
}
```

Luego debemos guardarlo y compilarlo con el comando **gcc [archivo]**. Ahora podremos usar este programa cada vez que se nos cuelgue la terminal.

```
gid 0-50 -
uid 1001 2001
```

Como vemos, el archivo es muy sencillo. Primero se indica qué es lo que se va a mapear (una ID de usuario o una ID de grupo), luego se especifica el valor remoto (o rango de valores remotos) y después se especifica el valor local.

## Hacer copias de respaldo con pdumpfs

**Pdumpfs** es un sistema simple para realizar copias de respaldo (backups diarios) de forma similar a **dumpfs**, que guarda

réplicas diarias del sistema. Cuenta con la posibilidad de acceder a réplicas antiguas en cualquier momento. Si conjugamos **Cron** con **pdumpfs** tendremos un potente sistema de backup. Veamos un ejemplo de su uso:

```
$ pdumpfs directorio_para_back
directorio_destino [nombre_backup]
```

**Pdumpfs** crea la réplica **AAAA/MM/DD** en el directorio de destino. Todos los archivos originales se copian al directorio réplica (destino) cuando **pdumpfs** se ejecuta por primera vez. A partir de la segunda vez y en adelante, **pdumpfs** sólo copia archivos nuevos o actualizados, y almacena los archivos sin modificar como **enlaces duros** a los archivos de la réplica del día previo para ahorrar espacio en el disco.

## Cómo bloquear el ping

Los **crackers**, al realizar un ataque informático de cualquier tipo a nuestro equipo, deben tener la certeza de que el equipo responde a una acción determinada. Para esto, necesitan conocer la dirección o el nombre con el cual se identifica en la red, luego proceder a

ejecutar un comando llamado **ping** (ver si obtiene respuesta (o sea, si el equipo está encendido). Podemos protegernos contra los **ping** ignorando cualquier comando **ICMP** (*Internet Control Message Protocol*). Esto se realiza de forma muy simple con un solo comando:

```
echo 1 /proc/sys/net/ipv4/icmp_echo_
ignore_all
```

Para volver al estado anterior, tenemos que cambiar el **1** por **0**.

## Screen: una herramienta para administradores

Esta aplicación se utiliza mucho para administrar servidores vía consola, ya que permite continuar con las tareas en ejecución aun cuando la ventana se cierra. Antes que nada tendremos que instalarlo, aunque la mayoría de las distribuciones lo incluye de forma nativa o lo podemos encontrar en sus repositorios originales. Su uso es muy simple se antepone el comando **screen** al comando que se quiere encapsular.

```
$ screen top
```

Cerramos la consola, abrimos otra y con el comando **screen -list** podemos ver las aplicaciones que están en **screens**.

```
ll screen -list
There are screens on:
    6719.pts-1.mandriva
    (Detached)
    6777.pts-3.mandriva
    (Detached)
    2 Sockets in /home/ariel/tmp.
```

**Screen** muestra que están corriendo dos procesos con los números **6719** y **6777**. Podemos llamarlos para continuar en ese momento o después para retomar la tarea.

```
$ screen -Dr 6719
```

Ahora la aplicación estará de nuevo frente a nuestros ojos.

## Cómo crear una cuenta de usuario sin contraseña

Crear una cuenta de usuario sin contraseña en nuestro sistema puede ser de utilidad en varios casos, aunque también puede ser causa de un alto grado de inseguridad. Esto quedará a criterio del administrador encargado del servidor. Para realizar esta tarea, nos valdremos del comando **passwd** del siguiente modo:

```
passwd -d nombre_usuario
```

Y cada vez que este usuario ingrese al sistema, no necesitará tipear clave alguna.

## Cómo controlar la actividad de la impresora desde la consola

Para manejar la impresora de forma simple y cómoda desde una consola, vamos a utilizar los comandos **lpq**, **lprm** y **lpc**. Una de las cosas que podemos hacer con estos comandos es verificar desde la terminal el estado de la cola de impresión. Para ello, vamos a tipear:

```
lpq -Printer -l
```

La opción **-l** se utiliza para ver toda la información de cada trabajo. Si queremos cancelar un trabajo, vamos a utilizar el comando **lprm** del siguiente modo:

```
lprm -Printer - user [numero-de-trabajo]
```

La opción **-** borra todos los trabajos de la cola especificada o la que haya por defecto, y con el número de trabajo se borra el trabajo al que el sistema haya otorgado esa identificación.

Vale agregar que el comando **lpc** nos permite comprobar el estado de la impresora y controlar algunos aspectos de su utilización.

## Usar Netcat como escaneador de puertos

Si no tenemos un escaneador de puertos a mano (como **Nmap**), podemos usar **Netcat**. Existe un parámetro llamado **-v** (**verbose**) que informa en pantalla un detalle del estado de la conexión. Si lo usamos dos veces, nos dará el doble de la información.

## III APROVECHAR LA MEMORIA AL MÁXIMO

El comando **exec** permite ejecutar un comando matando al proceso padre (generalmente el intérprete de comandos **bash**). Es bueno utilizarlo, por ejemplo, cuando se carga el modo gráfico (**exec startx**) o cuando se carga otra shell (**exec mc**) para liberar la memoria RAM ocupada por Bash (800K aproximadamente).

Ahora bien, para usar Netcat como escaneador de puertos, tenemos que utilizar el parámetro **-z** (**z**ero entrada y salida). Con este parámetro le indicamos que sólo queremos saber si se puede establecer la conexión pero no queremos enviar ni recibir ningún dato. Finalmente, sólo tenemos que indicarle a Netcat cuáles son los puertos que queremos verificar si están abiertos. Veamos un ejemplo:

```
nc -v -v -z 10.0.0.100 20 22 23 25 110
```

Con este comando le estamos pidiendo a Netcat que se conecte a 10.0.0.100, que nos informe todo lo que sucede durante el intento de conexión por la pantalla y que no envíe ningún dato. Finalmente, le estamos pidiendo que haga el intento con los puertos 20, 22, 23, 25 y 110 (**Figura 2**).

## Backup diferenciales con rdiff

La utilidad **rdiff-backup** ofrece la opción de realizar sencillos backups con historial diferencial para cualquier tipo de archivos, como enlaces simbólicos.

```
Macbook-de-Facu:~ hfarena$ nc -v -v -z 192.168.0.1 20 22 23 25 110
192.168.0.1: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.0.1] 20 (ftp-data) : Connection refused
(UNKNOWN) [192.168.0.1] 22 (ssh) : Connection refused
(UNKNOWN) [192.168.0.1] 23 (telnet) : Connection refused
(UNKNOWN) [192.168.0.1] 25 (smtp) : Connection refused
(UNKNOWN) [192.168.0.1] 110 (pop3) : Connection refused
sent 0, rcvd 0
Macbook-de-Facu:~ hfarena$
```

**Figura 2.** Salida del comando **nc** cuando intenta establecer comunicación con diferentes puertos.

Para hacer una copia de respaldo de mayor parte del directorio **~/** en **/mnt/backup** se usa el comando de la siguiente manera:

```
$ rdiff-backup --include ~/tmp
/guardar
--exclude ~/tmp ~/ /mnt/backup
```

Luego, para restaurar los datos de los siete días de este archivo, en la ubicación **~/antiguo** debemos escribir:

```
$ rdiff-backup -r 7D /mnt/respaldo
~/antiguo
```

## Realizar un backup de todo nuestro sistema

Ésta es una receta para llevar a cabo antes de borrar cualquier distro de un equipo, y ahorrará muchas horas de trabajo ya que hace una copia de seguridad de casi todos los aspectos del sistema. Creamos un archivo vacío llamado **back\_sistema.sh** y un directorio en la raíz llamado **/back**. Luego lo editamos con cualquier editor y escribimos las siguientes líneas:

```
#!/bin/sh
# Back de todo el sistema
rm -R /back/*
cp -R /etc/samba /back/samba
cp -R /etc/passwd /back/passwd
cp -R /etc/networks /back/
cp -R /etc/X11 /back
cp -R /etc/hosts /back
cp -R /etc/inittab /back
cp -R /etc/resolv.conf /back
cp -R /etc/lilo.conf
cp -R /boot/grub /back
cp -R /etc/rc2.d /back/rc2
cp -R /etc/rc3.d /back/rc3
cp -R /etc/rc3.d /back/rc5
cp -R /var/log /back/
cp -R /home /back
df >
    /back/config/espacio_disco.txt
chown -R
    nombre_usuario_administrador
    /back
```

Este script del sistema realiza la copia de todos los archivos de configuración importantes, además de los directorios clave como el que aloja a **Xorg.conf** (servidor de video), los **run level**, los archivos de configuración para el arranque, los logs, la configuración de nuestra red, la resolución de nombres mediante DNS y, por supuesto, todos los directorios personales de nuestros usuarios. Al final del script tenemos dos líneas interesantes: una de ellas es la que va generar un archivo llamado **espacio\_disco.txt**, donde veremos el avance de tamaño físico. Luego, la última línea con el comando **chown** nos brindará la posibilidad de cambiar los permisos para un



Con respecto a las copias de seguridad, siempre es recomendable almacenar los archivos comprimidos en un medio diferente de aquél en el que se está

creando la copia de seguridad. Si disponemos de dos unidades de disco rígido, es bueno copiar el archivo de copia de seguridad en ambas unidades. De todas maneras, lo mejor siempre es ubicar la copia fuera del equipo, como en un CD o en un DVD.

usuario en especial. Cabe aclarar que si alguna de estas líneas no nos es de utilidad, podemos agregar delante de ella el signo **#** para desactivarla.

## Cómo evitar los rootkits

Los **rootkits** son herramientas (si así se los pueden llamar) que se instalan en un equipo y, por medio de diferentes artilugios, logran proveer al hacker de datos importantes, como por ejemplo la password de **root** o de algún usuario del sistema. Generalmente, funcionan reemplazando utilidades típicas del sistema (como **login**, **adduser** o **passwd**) por **emulaciones** de estas aplicaciones que se ven y funcionan igual, pero almacenan la información que procesan en un archivo para que luego el hacker lo descargue. Es muy común que los rootkits reemplacen la pantalla de ingreso al sistema por un programa que muestra **login** y **password** pero que, además de permitirle el acceso al usuario que está siendo engañado, guardan esa información en un archivo sin encriptar para que el hacker luego pueda descargarlo.

Muchos rootkits se pueden instalar desde una cuenta de usuario común y como administradores debemos tener la precaución de mantener nuestro sistema libre de rootkits. Para esto existe una herramienta que se llama **chkrootkit** ([www.chkrootkit.org](http://www.chkrootkit.org)), que analiza la integridad de las herramientas del sistema en búsqueda de diferencias que denoten la existencia de un rootkit. Su uso es muy sencillo: una vez descargada, con sólo ingresar el comando **chkrootkit** como **root** desde una consola, estaremos verificando la integridad del sistema.

## Cómo poner la placa de red en modo promiscuo

Los **sniffers** son utilidades que permiten escuchar todo el tráfico de la red. Son muy utilizados para llevar a cabo técnicas de **hacking**, robo de claves, etcétera. Para eso, desde un sistema se pone la placa de red en modo **promiscua**, lo que significa que comienza a escuchar todo el tráfico de la red en lugar de prestar atención solamente a los datos que le envían. Bien, con **ifconfig** podemos saber si nuestra placa de red está en modo promiscua (no debería estarlo, pero lo está si hay un sniffer instalado). ¿Cómo detectarlo? Sencillo: debemos buscar una línea como la siguiente en la información del dispositivo:

```
UP BROADCAST NOTRAILERS RUNNING
MULTICAST MTU:1500 Metric:1
```

Cuando un dispositivo está en modo promiscuo se ve del siguiente modo:

```
UP BROADCAST NOTRAILERS RUNNING
PROMISC MULTICAST MTU:1500 Metric:1
```

O sea, ahora aparece la palabra **PROMISC** en el estado de la red. Podemos poner la placa de red en modo promiscuo con el comando **ifconfig** del siguiente modo:

```
ifconfig [dispositivo] promisc
```

## Enviar y recibir archivos entre equipos de una red con nc

Una funcionalidad muy interesante de Netcat es que nos permite transferir archivos de una máquina a otra sin necesidad de que haya un servicio como **FTP**, **NFS** o **SSH** montado. Simplemente usamos el comando Netcat en las dos computadoras, de las cuales una actuará de **servidor** y la otra de **cliente**. Para enviar un archivo de una máquina a otra, en la máquina que va a **recibir** el archivo escribimos el siguiente comando:

```
nc -v -l -p 3000 -w 3 > [nombre_de_archivo]
```

Analicemos: estamos levantando un servidor en el puerto 3000 que redireccionará todo lo que entre hacia un archivo llamado **[nombre\_de\_archivo]**. Desde el **emisor** ahora tipearemos el siguiente comando:

```
nc -v [IP_receptor] 3000 < [nombre_de_archivo]
```

Como vemos, en el emisor lo único que hacemos es conectarnos al receptor con su correspondiente **puerto** (en este caso

3000), pero se puede elegir cualquier otro siempre y cuando no esté ocupado por algún servicio en funcionamiento) y le redireccionamos como flujo de entrada el contenido del archivo que queremos enviar.

## Cómo verificar la integridad del sistema de archivos

Bien GNU/Linux hace un chequeo general del sistema en cada inicialización, es bueno que hagamos una verificación un poco más detallada por medio de la herramienta **fsck**. El formato básico de este comando es:

```
fsck.ext3 opcion dispositivo
```

Pero si disponemos de una partición Extended-2 en lugar de tener una Extended-3, debemos utilizar el comando **fsck.ext2**. Las opciones que permite utilizar son las siguientes:

- p**: reparación automática (sin preguntas).
- n**: no hacer cambios al sistema de archivos.
- y**: asumir siempre la respuesta **y**.
- e**: chequeo de bloques defectuosos.
- f**: chequeo forzoso aunque el sistema esté marcado como limpio.
- v**: modo **verbose**.

Un ejemplo de utilización de esta aplicación puede ser:

```
fsck.ext3 -p /dev/hda1
```

Este comando buscará errores en la primera partición del disco rígido

principal y no hará preguntas sobre qué hacer con ellos. Si la partición que va a chequear está montada, el programa devolverá el siguiente mensaje:

```
dev/hda1 is mounted.
```

```
WARNING!!! Running e2fsck on a
mounted filesystem may cause
SEVERE filesystem damage.
```

```
Do you really want to continue
(y/n)?
```

Cuando ocurre esto, es recomendable no continuar, desmontar la partición que se quiere chequear y tipear el comando de nuevo.

## Cómo configurar múltiples IPs en la misma placa de red

Si hay cosas interesantes para hacer, eso es definir dos direcciones IP simultáneas para el mismo dispositivo. Esto es muy útil para armar **firewalls** más seguros con una sola placa de red, entre otras cosas. Para llevar a cabo esto, usamos el comando **ifconfig** del siguiente modo:

```
ifconfig [dispositivo]:[valor] [ip]
```

Veamos un ejemplo:

```
ifconfig eth0 up 10.0.0.1
ifconfig eth0:0 10.0.0.2
```

El valor es un número que se le asigna al dispositivo virtual, en orden ascendente.

## Transferencia encriptada de archivos vía SSH

Cuando trabajamos de forma remota, los archivos que generamos se almacenan en el servidor. Esto puede presentar una dificultad ya que si necesitamos almacenar algo de este proceso remoto de forma local, deberíamos usar algún servicio de transferencia de archivos como NFS o FTP.

Si usamos el sistema **OpenSSH** ([www.openssh.org](http://www.openssh.org)), disponemos de un set de herramientas que nos permiten enviar y recibir archivos desde el servidor hacia nuestro cliente. Es importante destacar que para usar estas herramientas, debe estar el servidor **sshd** corriendo en la máquina remota y debemos tener los comandos necesarios en el sistema local. OpenSSH nos ofrece dos herramientas para satisfacer nuestra necesidad de enviar y recibir archivos. Una de ellas es **scp** (o **Secure cp**), que es una herramienta de **copia segura** con una sintaxis muy similar a la del comando **cp** de UNIX. Antes de usarla, verifiquemos que esté instalado el comando en nuestro sistema:

```
facundo@Servidor:~> scp
usage: scp [-pqrvcBC1246] [-F
config] [-S program] [-P
port]
        [-c cipher] [-i
identity] [-l
limit] [-o
option]
[[user@]host1:]file1
[...] [[user@]
host2:]file2
facundo@Servidor:~>
```

Bien, todo lo que necesitamos en el cliente está instalado. En caso de no estarlo, nuevamente se pueden tomar los CDs de la distribución que estemos usando e instalar las herramientas de cliente SSH. La sintaxis de este comando es muy sencilla. Si queremos descargar archivos del servidor a nuestro host local debemos tipear:

```
scp usuario@host_remoto:/[ruta_al_archivo_remoto] .
```

Entonces **scp** nos preguntará la clave correspondiente al usuario y luego procederá a copiar el archivo a nuestro sistema local. Podemos notar que el tercer parámetro de este comando es un simple punto (.) que indica a **scp** que queremos descargar el archivo al directorio en el cual estamos parados actualmente. En caso de querer almacenarlo en otro directorio, sólo debemos ingresar la ruta completa a él. En caso de querer enviar archivos locales hacia el servidor, debemos usar el comando **scp** del siguiente modo:

```
scp [ruta_al_archivo_local] usuario@host:/[ruta_destino]
```

Aquí la situación es a la inversa ya que se indica primero la ubicación local del archivo que queremos enviar y luego el destino en el servidor. De más está decir que sólo podremos escribir archivos en el servidor en los directorios en los que el usuario que estamos tenga **permisos de escritura**. Si estamos accediendo como administrador (usuario **root**), entonces podremos escribir archivos en cualquier directorio del servidor.

## Recuperar la password de administrador en MySQL

Es normal olvidarse las claves de un administrador en un servidor de base de datos, y ni hablar si tenemos que trabajar detrás de otro admin que abandonó su función. En cualquier caso, existe una forma de resetear las claves, que además se puede utilizar para cambiar los permisos de las tablas sin problemas. Por supuesto, el primer paso es apagar el servidor de base de datos con la siguiente línea:

```
/etc/init.d/mysql stop
```

Ahora vamos a volver a iniciar el servidor MySQL indicándole no tener en cuenta ninguna gestión de permisos:

```
/usr/bin/mysqld_safe --skip-grant-tables &
```

Luego, vamos a conectar con MySQL, sin password, a la base de datos principal de gestión (que lleva el nombre **mysql**):

```
mysql -h nuestro_hostname mysql
```

Vamos a brindarle permisos al administrador para todo (hay que indicar la nueva password):

```
mysql
update user
set Select_priv='Y',
        Insert_priv='Y',
        Update_priv='Y',
        Delete_priv='Y',
        Create_priv='Y',
```

```
Drop_priv='Y',
Reload_priv='Y',
Shutdown_priv='Y',
Process_priv='Y',
File_priv='Y',
Grant_priv='Y',
References_priv='Y',
Index_priv='Y',
Alter_priv='Y',
Show_db_priv='Y',
Super_priv='Y',
Create_tmp_table_priv='Y',
Lock_tables_priv='Y',
Execute_priv='Y',
Repl_slave_priv='Y',
Repl_client_priv='Y',
Create_view_priv='Y',
Show_view_priv='Y',
Create_routine_priv='Y',
Alter_routine_priv='Y',
Create_user_priv='Y',
Password =
PASSWORD('contraseña')
WHERE User='root';
```

El paso siguiente es recargar los nuevos privilegios:

### FLUSH PRIVILEGES;

Salir del cliente MySQL, apagar MySQL y volverlo a iniciar normalmente con la siguiente línea:

```
/etc/init.d/mysql start
```

¡Y listo! De esta forma conseguimos acceso a **root** en el servidor donde se aloja MySQL y ya podemos resetear todo.

No todas las distribuciones incluyen los paquetes de gestión de cuotas de disco. Podemos buscar en el repositorio por el nombre **quota** o **quota-utils**.

Es posible limitar a un usuario a usar un solo programa de la consola cambiando el último parámetro de su línea en el archivo **/etc/passwd**.

Las claves de los usuarios se almacenan de manera encriptada en el archivo **/etc/shadow**. No es conveniente editar este archivo a mano.

Con el comando **showmount -exports [ip]** es posible obtener un listado de los directorios exportados en un equipo con NFS.

Con **smbclient -L //equipo/** es posible ver un listado de todos los recursos compartidos en un equipo Windows o Linux con Samba.

**6** Se puede usar iptables para bloquear los pings con el comando **iptables -A INPUT -p icmp -j DROP**.

**7** Los archivos de registro de actividad del sistema se almacenan en **/var/log**. Allí los más importantes son **/var/log/syslog** y **/var/log/messages**.

**8** Todos los scripts de ayuda para la administración del equipo deben estar almacenados en **/usr/sbin** con permisos de ejecución exclusivos para el usuario administrador.

**9** Nunca es recomendable dejar una terminal abierta (local o remota) con el usuario **root** en actividad.

**10** Si presionamos la tecla **k** en la aplicación **top** es posible eliminar un proceso rápidamente.

# Trucos para KDE y GNOME

En este capítulo aprenderemos una completa selección de trucos para aprovechar al máximo las infinitas funcionalidades que ofrecen los dos gestores de escritorio más populares: KDE y GNOME.

Conexión automática a la red en KDE	84
Montaje automático de disquetes bajo KDE	84
KDE con transparencias	85
Trucos para KWrite	87
Trucos para KWord	88
Montar un servidor de acceso remoto en modo gráfico con KDM	89
Configuración de GDM para montar un servidor de acceso remoto en modo gráfico	92
Cómo crear combinaciones de teclas personalizadas en KDE	97
Protocolos soportados por Konqueror	99
Protocolos soportados en Nautilus	99
Controlar las aplicaciones de KDE desde mensajes DCOP	100
Hacer magia con DCOP	101
Cómo hacer que el papel tapiz de GNOME se cambie aleatoriamente	104
Cómo eliminar el menú de usados recientemente en GNOME	105
Script para cambiar el fondo de GNOME-Terminal de forma aleatoria	105
Cómo cambiar el splash screen en GNOME	107
Dominar la terminal Konsole	107
Secretos de KWIN	109
Bonus tips	110

## Conexión automática a la red en KDE

Un problema clásico del entorno gráfico KDE es que cuando tenemos que acceder a un determinado servidor, aparece la necesidad de ingresar en cada conexión un determinado nombre de usuario y su contraseña. Si estamos acostumbrados a acceder a recursos remotos en nuestra LAN, esta tarea es tediosa. Pero por suerte hay una solución: vamos a realizar una **conexión predeterminada** desde el panel de control provisto por KDE denominado **kcontrol**. Para ello, sólo debemos dirigirnos al menú principal, seleccionar **Ejecutar** y escribir la palabra **kcontrol**. De forma automática vamos a ingresar al panel, y deberemos ir a la solapa **Internet y Red** y dirigirnos a **Navegación por la red local**. Allí ingresamos en la solapa derecha el usuario y la contraseña. Aplicamos los cambios y listo: la conexión se encuentra

## KPPP PARA TODOS

Un problema muy común en las distribuciones de Linux es que la aplicación **KPPP** sólo puede ser ejecutada por el usuario **root**, lo que entonces permite que sólo ese usuario sea el que puede iniciar la conexión a Internet. Para solucionar esto, hay que hacer clic con el botón derecho del mouse sobre el icono de **Kppp**, seleccionar la solapa **Permisos** y habilitar la opción **SUID**.

ya definida para que se realice de manera automática (**Figura 1**).

## Montaje automático de disquetes bajo KDE

Para montar de forma automática un disquete, sólo debemos abrir una consola y ejecutar el siguiente comando:

`mcedit Desktop/floppy.kdelnk`

De esta manera, se abrirá un editor basado en el muy conocido **Midnight Commander** con el texto de acceso al disquete en el escritorio. En él debemos modificar el valor contenido dentro de la línea **FStype=Default** por **FStype=v-fat**.

## KDE con transparencias

Desde la versión 3.4, KDE viene con soporte de transparencias, y la configuración es muy simple. La totalidad de la configuración se realiza, como primera medida, desde el panel de control **kcontrol** en la sección **Aspecto/Aspecto de las ventanas** y, en la solapa derecha, encontraremos **Translucidez**, que activamos con tan sólo un clic. Al iniciar KDE de nuevo, seguramente veremos un error que nos informa la

necesidad de añadir una línea dentro del archivo del servidor gráfico X.org. Para que los efectos de translucidez y opacidad funcionen en nuestro KDE, necesitamos contar con el servidor gráfico X.org superior a la versión 6.8. A continuación vamos a editar el archivo gráfico **xorg.conf** ubicado en **/etc/X11/xorg.conf**, donde añadiremos al final las siguientes líneas:

```
mcedit /etc/X11/xorg.conf

Section "Extensions"
Option "Composite" "Enable"
EndSection
```

Reiniciamos el servidor gráfico con la combinación de las teclas **CTRL + ALT + RETROCESO**, ingresamos nuevamente con nuestro usuario y ya tendremos ventanas transparentes (**Figura 2**).

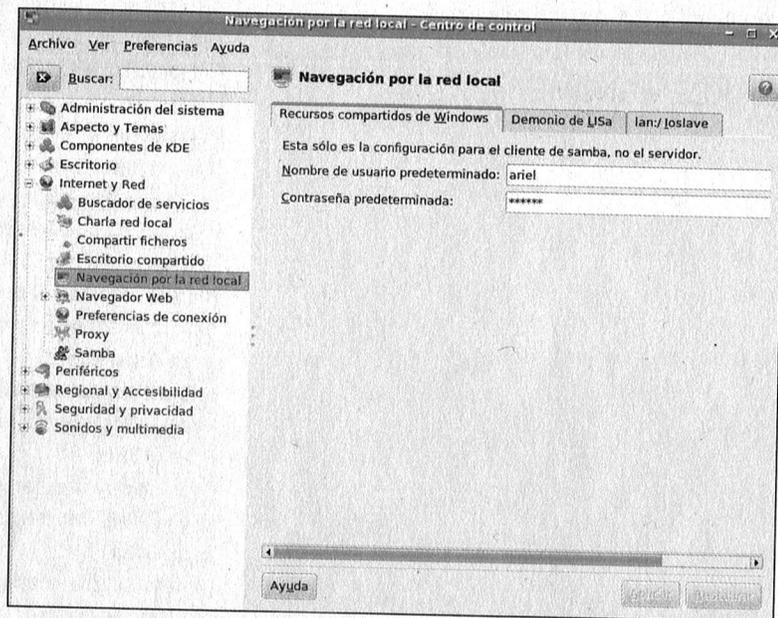


Figura 1. Ventana de definición de las opciones de acceso automático a la red en KDE.

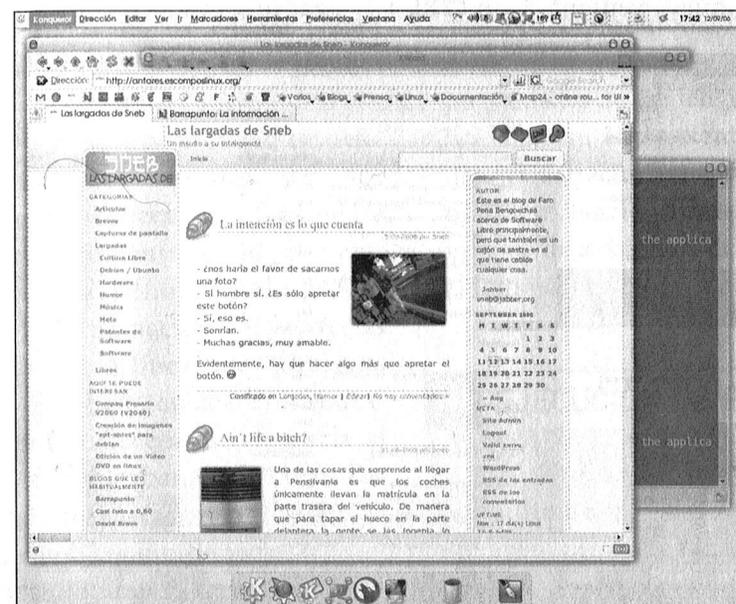


Figura 2. Habilitar las transparencias en KDE no es una tarea demasiado complicada.

## Navegar con palabras clave

En **Konqueror** contamos con la posibilidad de definir palabras que nos servirán como atajos para los sitios más visitados en la red de redes. Para configurarlos, sólo debemos ir a **Propiedades/Configurar Konqueror/Configuración de la navegación mejorada**, tal como muestra la **Figura 3**. Por ejemplo, al escribir **gg:gnu** en la barra de dirección, se abrirá el buscador **Google** con palabra **gnu**.

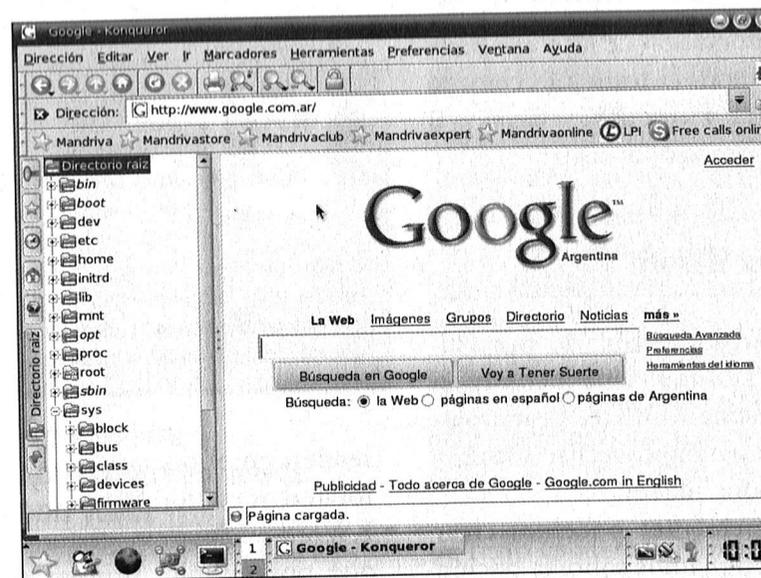
## Particularidades de Konqueror

Konqueror es mucho más que un simple manejador de archivos. Está lleno de funcionalidades que quizá nunca lleguemos a descubrir por completo. Aquí hay algunas características que lo hacen sumamente útil y práctico.

- Si presionamos **CTRL + N** podemos abrir una nueva ventana. Con **CTRL +**

**SHIFT + N** podremos abrir una nueva pestaña de visualización.

- En el menú **Edición/Crear nuevo** podremos acceder a toda una serie de opciones relacionadas con la creación de contenido. Desde una nueva carpeta hasta una página web.
- En el menú **Ver** podremos personalizar la visualización de nuestro panel principal, entre otras cosas (**Figura 4**). Además, también podemos cambiar la forma de vista (árbol, lista, iconos, etcétera), poner una imagen de fondo y agrandar o achicar los iconos, entre otras opciones.
- Con la tecla **F5** actualizamos la vista actual si es que hicimos cambios fuera de Konqueror y éstos no se reflejan en la ventana.
- En el menú **Herramientas** tenemos accesos rápidos hacia utilidades, como abrir una terminal, ejecutar un comando sin abrir una terminal o realizar una búsqueda de archivos.



**Figura 4.** Konqueror ofrece muchas formas diferentes de visualización de los archivos.

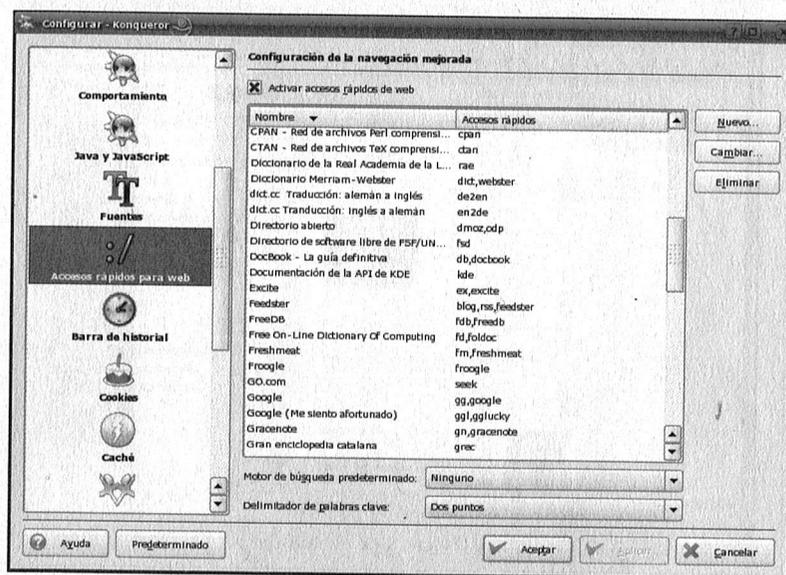
## Trucos para KWrite

Este editor de texto no difiere mucho de los que ya conocemos, con las típicas funciones de **Abrir**, **Guardar**, e **Imprimir** en el menú **Archivo** y las funciones de búsqueda en el menú **Edición**. Pero **KWrite** tiene algunas particularidades que veremos a continuación:

- **KWrite** nos da la posibilidad de definir **marcadores visuales** que nos permitirán encontrar más fácilmente la información dentro de los archivos de texto. Para definirlos, sólo presionamos la combinación de las teclas **CTRL + B** en la línea que queremos marcar. Entonces, en el menú **Marcadores**, nos aparecerá la referencia al marcador que acabamos de crear. Para borrar un marcador, utilizamos la misma combinación de teclas.
- Si vamos a utilizar **KWrite** para programar, es bueno saber que con la tecla **F6** podemos habilitar la alineación au-

tomática del texto, y con la tecla **F11** podemos habilitar una columna que nos indica el número de línea en la que estamos trabajando. En **Herramientas/Resaltado** podemos definir en qué lenguaje de programación estamos trabajando para que automáticamente el código se coloree según las funciones y propiedades de ese lenguaje (**Figura 5**).

- En el menú **Herramientas** encontraremos el **Corrector ortográfico**. Antes de usarlo, es bueno chequear en la parte inferior de la ventana del corrector que el idioma del texto esté bien definido.



**Figura 3.** Forma de configuración de palabras clave en Konqueror.

## EJECUCIÓN DE PROGRAMAS

KDE y GNOME ofrecen una combinación de teclas para la ejecución rápida de comandos. Si presionamos **ALT + F2** se abrirá una ventana de diálogo que nos permitirá tipear el comando que sea necesario ejecutar.

Con la combinación de las teclas **CTRL + U** convertimos el texto seleccionado a mayúsculas. Y con **CTRL + SHIFT + U** hacemos el proceso inverso.

## Trucos para KWord

KWord es la aplicación de procesamiento de texto que viene incluida en a suite de oficina KOffice. Veamos algunos trucos para aprovechar al máximo sus funcionalidades:

- Desde el menú **Ver** podemos crear vistas si presionamos sobre la opción **Nueva vista**. Esto abrirá una nueva ventana con el documento que estamos editando.
- Es posible insertar imágenes, tablas y otros objetos. Si tenemos un escáner conectado y configurado, entonces podremos utilizar la opción **Escanear imagen** del menú **Insertar**, que se encargará del trabajo de escaneo e in-



Kword es un procesador de texto que resulta ideal para los sistemas de bajos recursos, ya que ofrece muchas funcionalidades a bajo costo porque no consume tanta memoria RAM como OpenOffice Writer, por ejemplo. Otra opción muy interesante para este tipo de equipos es Abiword, que posee la misma filosofía de ofrecer mucho a cambio de poco.

serción en el documento de manera totalmente automática.

- KWord incluye un gestor de estilos que nos permite brindar diferentes looks a nuestro documento con tan sólo un clic. Para acceder a él, sólo debemos presionar la combinación de las teclas **CTRL + ALT + S**.
- En el menú **Herramientas**, encontraremos el **Corrector Ortográfico** con el diccionario en español habilitado.
- Otra de las características de KWord es que sirve como herramienta de de-

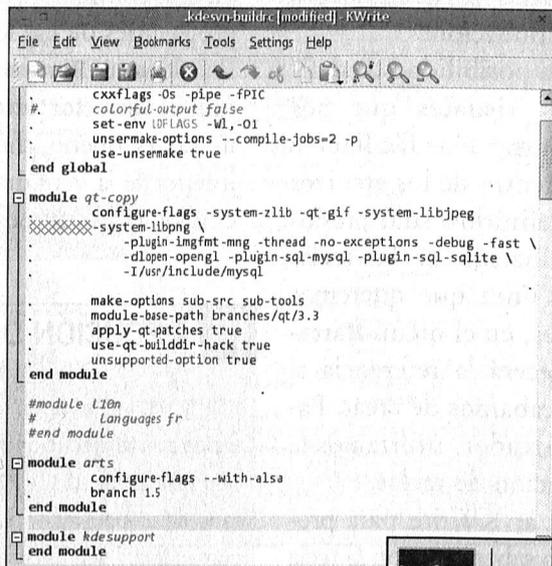


Figura 5. Entre otras cosas, KWrite ofrece coloreado de código fuente de programación.

desarrollo de páginas web. Para esto, simplemente hay que armar la página tal como cuando trabajamos editando un texto y luego elegir la opción **Documento HTML** del menú **Guardar Como**. Sólo hay que tener la precaución de guardar la página web en el mismo directorio en el que se encuentran las imágenes que se le insertaron.

## Montar un servidor de acceso remoto en modo gráfico con KDM

Un servidor de acceso remoto en modo gráfico nos permite usar una computadora de baja potencia como cliente de un servidor de aplicaciones para usar las utilidades allí alojadas de manera remota. Para configurar KDM como servidor de acceso gráfico, debemos ubicar el archivo **kdmrc** y editarlo con un editor

## CALCULADORA DE GOOGLE

El buscador más famoso del mundo posee una calculadora integrada. Podemos poner formulas matemáticas en la barra de búsqueda y nos ofrecerá el resultado al instante. Desde Konqueror, si usamos el prefijo **gg:**, podemos realizar búsquedas directamente en Google desde la barra de dirección. Este prefijo también funciona con las fórmulas matemáticas, por lo que por ejemplo, **gg:2\*sqrt(2)** funciona sin problemas.

de texto (podemos buscarlo con la herramienta de búsqueda de la distribución que usemos). El archivo tiene la siguiente estructura:

```
[General]
AuthDir=/var/lib/xdm/authdir/
r/authfiles/
```

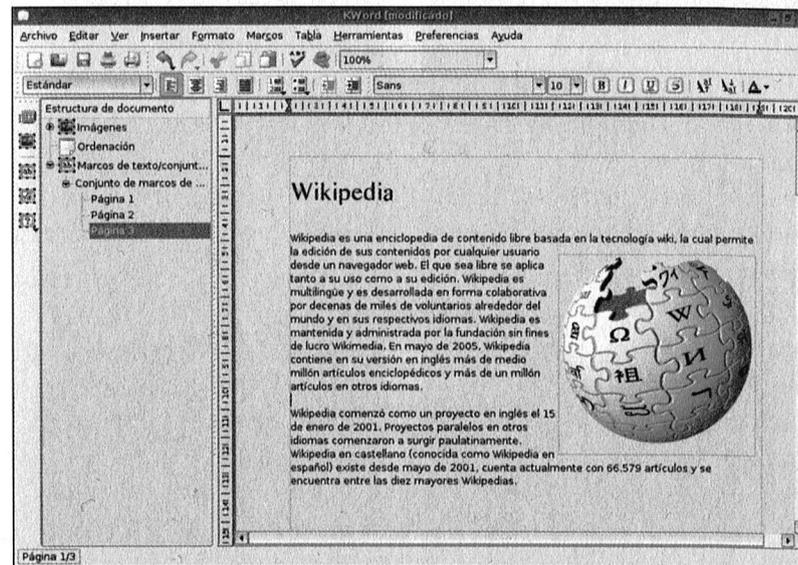


Figura 6. Podemos editar un archivo .PDF en KWord, el procesador de texto de la suite KOffice.

```

ConfigVersion=2.0
DaemonMode=true
PidFile=/var/run/kdm.pid
Xservers=/etc/opt/kde3/share/
    config/kdm/Xservers

[Shutdown]
HaltCmd=/sbin/halt
LiloCmd=/sbin/lilo
LiloMap=/boot/map
RebootCmd=/sbin/reboot
UseLilo=false

[X-* -Core]
AllowNullPasswd=true
AllowRootLogin=true
AllowShutdown=All
AutoReLogin=true
Reset=/etc/X11/xdm/Xreset
Resources=/etc/X11/xdm/Xresources
Session=/etc/X11/xdm/Xsession
Setup=/etc/X11/xdm/Xsetup
Startup=/etc/X11/xdm/Xstartup
TerminateServer=true

[X-* -Greeter]
AdminSession=YaSTadminSession
AntiAliasing=true
BackgroundCfg=/etc/opt/kde3/shar
    /config/kdm/backgroundrc
ColorScheme=SuSE
DefaultUser=facundo
EchoMode=OneStar
FaceSource=AdminOnly
FailFont=Nimbus Sans 1,14,
    1,5,74,0,0,0,0,0
FocusPasswd=false
GUIStyle=Keramik
GreetFont=Nimbus Sans 1,20,
    1,5,48,0,0,0,0,0

```

```

GreetString=SuSE Linux 9.0 (%h)
GreeterPosFixed=false
GreeterPosX=100
GreeterPosY=100
HiddenUsers=root,nobody,bigsister
Language=es
LogoArea=Clock
LogoPixmap=
MaxShowUID=65000
MinShowUID=500
PreselectUser=None
SelectedUsers=
SessionTypes=kde,gnome>windowmak
    r,blackbox,cdesim,fvwm2,icewm,
    twm,failsafe,
ShowUsers=NotHidden
SortUsers=true
StdFont=Nimbus Sans 1,14,
    1,5,48,0,0,0,0,0
UseBackground=true

[X-:* -Core]
AllowNullPasswd=true
AllowRootLogin=true
AllowShutdown=All
NoPassEnable=false
NoPassUsers=

[X-:* -Greeter]
AdminSession=YaSTadminSession

[X-:0 -Core]
AutoLoginEnable=true
AutoLoginUser=facundo
NoPassEnable=false
NoPassUsers=

[X-:0 -Greeter]
AdminSession=
LogSource=/dev/xconsole

```

```

showLog=false

[Xdmcp]
Enable=false
Willing=/etc/X11/xdm/Xwilling
Xaccess=/etc/X11/xdm/Xaccess

```

Lo que debemos hacer es ir a la sección **[Xdmcp]** y poner la opción **Enable=true**. Esto habilitará el protocolo **XDMCP** que es el utilizado para gestionar sesiones remotas en modo gráfico. Este archivo tiene algunas opciones más que quizá sean de nuestro interés. Veamos:

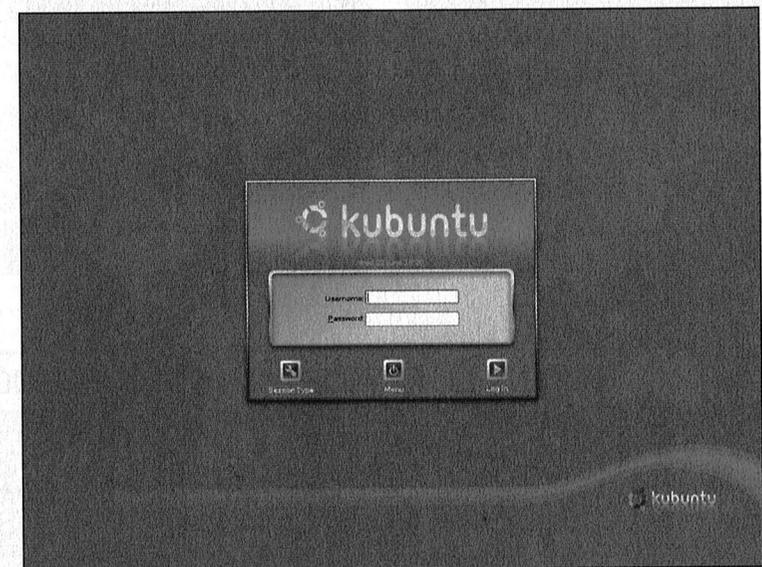
- **DaemonMode=true:** habilita a KDM para que funcione como demonio en segundo plano. Esto hará que el servidor no se cierre cada vez que alguien se desconecta.
- **HaltCmd=/sbin/halt:** con este comando definiremos con qué comando se apagará el sistema si remotamente elegimos la opción **Cerrar el sistema**.

- **AllowRootLogin=true:** permite el ingreso del usuario administrador o no.
- **AllowShutdown=root:** con esta opción podemos definir qué usuario es el único que puede apagar el equipo.

Cuando ya realizamos las modificaciones necesarias, guardamos el archivo y reiniciamos el sistema. Desde un equipo cliente, con el modo gráfico cerrado (con el comando **init 3** nos aseguramos de pasar a modo texto total en el cliente) tipeamos el siguiente comando para conectarnos:

#### X -query 10.0.0.100

Esto levantará el entorno gráfico y hará que se conecte con la dirección **10.0.0.100**, que debemos reemplazar por la IP que corresponda a nuestro servidor X. Ahora deberíamos ver la pantalla de bienvenida al sistema tal como la vemos cuando estamos sentados frente al equipo servidor (**Figura 7**).



**Figura 7.** Una típica pantalla de inicio de sesión en KDM.

## Configuración de GDM para montar un servidor de acceso remoto en modo gráfico

GDM es mucho más que un simple gestor de sesiones. Podemos usarlo para montar un servidor de acceso remoto en modo gráfico o para usar las aplicaciones de nuestro equipo desde cualquier otra terminal de la red (Figura 8). El proceso es muy sencillo, ya que simplemente tenemos que buscar el archivo `gdm.conf` (en general se encuentra ubicado en `/etc/opt/gdm/gdm.conf`) y editar algunos parámetros.

El archivo es bastante más largo que el de KDM, pero debemos identificar en él la siguiente sección:

```
[xdmcp]
# Distributions: Ship with this
# off. It is never a safe
# thing to leave
# out on the net. Alternatively
# you can set up
# /etc/hosts.allow and
# /etc/hosts.deny to only allow
# say local access.
Enable=false
# Honour indirect queries, we
# run a chooser for these, and
```

```
then redirect
# the user to the chosen host.
Otherwise we just log the
user in locally.
HonorIndirect=true
# Maximum pending requests
MaxPending=4
MaxPendingIndirect=4
# Maximum open XDMCP sessions at
any point in time
MaxSessions=16
# Maximum wait times
MaxWait=15
MaxWaitIndirect=15
# How many times can a person
log in from a single host.
Usually better to
# keep low to fend off DoS
attacks by running many
logins from a single
# host. This is now set at 2
since if the server crashes
then gdm doesn't
# know for some time and
wouldn't allow another
session.
DisplaysPerHost=2
# The number of minutes after
which a non-responsive
session is logged off.
# Better keep this low. This
```

```
is the last version where
this will be minutes.
# In the current development
versions this is replaced
with PingIntervalSeconds
# which defaults to 15 which is
much nicer.
PingInterval=1
# The port. 177 is the standard
port so better keep it that
way
Port=177
# Willing script, none is
shipped X11's one is used by
default. If
# none is present we'll send
hostname system id. But if
you supply
# something here, the output of
this script will be sent as
status of
# this host so that the chooser
can display it. You could
for example
# send load, or mail details for
```

```
some user, or some such.
Willing=/etc/X11/xdm/Xwilling
```

Nuevamente, debemos ubicar la sección **Enable=false** y cambiarla por **Enable=True**. Algunos parámetros de configuración adicionales de este archivo son:

- **MaxSessions=16**: número máximo de clientes que el servidor soportará.
- **DisplaysPerHost=2**: número máximo de sesiones por equipo que soportará.

Cuando terminamos de editar, guardamos y reiniciamos el equipo (o reiniciamos el servicio `gdm`). El proceso para conectarnos desde los clientes es exactamente el mismo en los dos sistemas. Sólo debemos ejecutar el comando `X-query [IP]` desde el cliente.

## Tips para KSpread

KSpread es la planilla de cálculo de proyecto KOffice (Figura 9). Es muy

## III ENVIAR UNA VENTANA AL FONDO CON UN CLIC EN GNOME

Si hacemos clic con el botón del medio del mouse en la barra de título de una ventana, la podremos enviar directamente al fondo del escritorio (atrás del resto de las ventanas). Si tenemos un mouse que sólo cuenta con dos botones, podemos presionar ambos al mismo tiempo para lograr el mismo resultado.

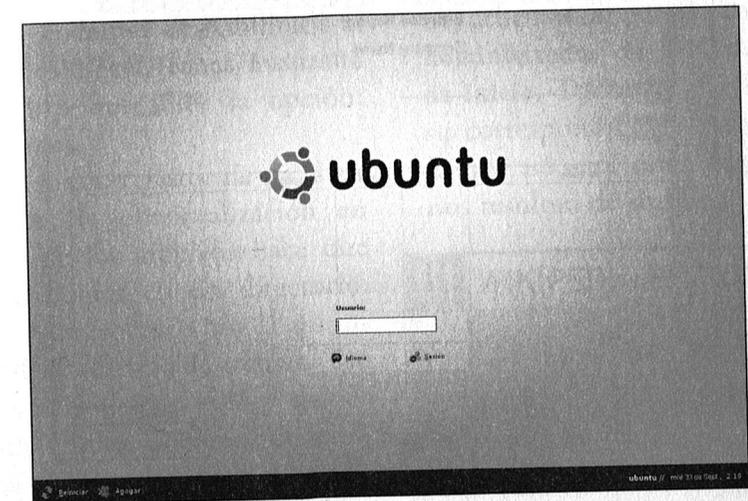


Figura 8. Ejemplo de pantalla de inicio de sesión en GDM.

completa y muchas personas ignoran el potencial que ofrece. Veamos algunas de sus particularidades:

- KSpread tiene varias **plantillas** prearmadas que se presentan ante nosotros cuando iniciamos la aplicación. Podemos escoger entre modelos de plantillas para el hogar o para la oficina.
- Al igual que con KWord, podemos crear múltiples vistas del mismo documento desde el menú **Ver**.
- En el menú **Herramientas**, tenemos una utilidad para generar calendarios de manera fácil y rápida.
- Dentro de las celdas podemos incluir varios objetos, como imágenes y enlaces a sitios de Internet.

## Quitar artilugios visuales a KDE

El gestor de escritorios KDE incluye muchos **artilugios visuales** para hacer

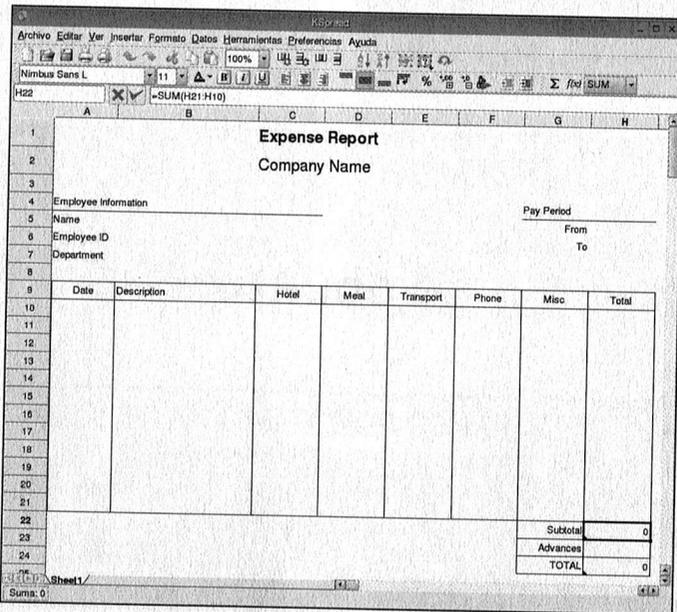


Figura 9. KSpread es la aplicación de hoja de cálculo incluida en el proyecto KOffice.

que nuestro escritorio se vea más lindo. Sin embargo, en algunos casos sucede que esos adornos hacen que nuestro equipo sea mucho más lento. Veamos qué podemos afinar para optimizar un poco la velocidad en el modo gráfico. Todo lo que sigue, lo podemos llevar a cabo desde el **Centro de control de KDE (Figura 10)**, ubicado en el menú **Sistema/Configuración/Configurar su escritorio**.

- Hay varios adornos aplicados a las ventanas que consumen recursos y ayudan poco, aunque se vean lindos. Para deshabilitarlos, podemos ir a **LookNFeel/Decoración de ventanas** y probar con quitar las opciones **Usar texto sombreado** y **Botones Animados**.
- Algunas distribuciones configuran múltiples escritorios virtuales (más que los dos que incluye **Mandriva Linux**). Casi nadie usa tantos, por lo que podemos deshabilitarlos para ganar algunos bytes de RAM. Para eso,

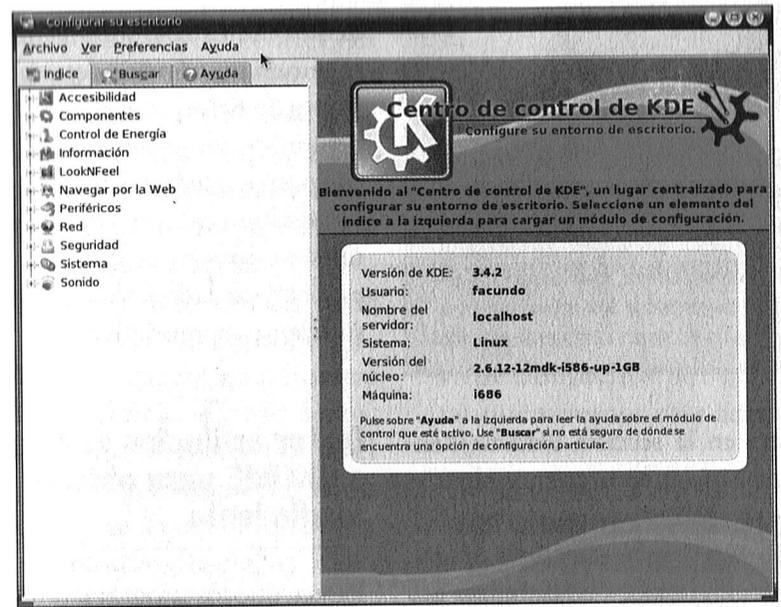


Figura 10. Panel de configuración de los parámetros visuales de KDE.

tenemos que ir a **LookNFeel/Escritorios Múltiples** y configurar el número de escritorios en **1**.

- El **anti-aliasing** (suavizado de bordes) de las fuentes también consume recursos. Podemos deshabilitarlo en la opción **LookNFeel/Fuentes/Anti-Aliasing**.
- La animación de los iconos consume mucho y aporta poco también. Si queremos, podemos deshabilitarla si vamos a **LookNFeel/Iconos/Avanzado** y allí dentro buscamos la opción: **Animar iconos**.
- En el navegador y gestor de archivos **Konqueror**, la previsualización en miniatura de los archivos hace que cuando entremos en un directorio, tengamos que esperar a que la aplicación analice cada tipo de archivo. Eso toma bastante tiempo y si podemos prescindir de ello, podemos deshabilitar esta funcionalidad en **Administrador de Archivos/Previsualización** y

**meta-datos**. Particularmente, la opción que buscamos se llama **Usar miniaturas empotradas en archivos**.

- Si KDE se toma mucho tiempo para iniciar, es posible que se deba a que demasiados servicios de KDE están configurados para iniciar de forma automática. Para ver un listado y poder decidir cuáles queremos y cuáles no, debemos dirigirnos a **Componentes/Administrador de Servicios/Servicios de Inicio**. Todos los servicios tienen su correspondiente descripción, por lo que no será difícil definir cuáles nos resultan de utilidad.

## WIKIPEDIA EN KONQUEROR

Podemos usar el prefijo **wp:** en la barra de direcciones de Konqueror para realizar búsquedas directamente en Wikipedia. Por ejemplo: **wp:GNUPedia**.



Muchas distribuciones, como Ubuntu por ejemplo, presentan un diseño de interfaz de usuario minimalista. Esto tiene como principal finalidad la de lograr que el sistema operativo

funcione en una mayor cantidad de configuraciones de equipos. Luego, si el usuario lo desea, puede agregar todos los artilugios visuales que quiera de manera sumamente sencilla.

Finalmente, en la sección **Sistema/Rendimiento de KDE**, encontraremos algunas opciones interesantes para optimizar el desempeño de nuestro entorno visual. Por ejemplo, en la sección **Minimizar uso de memoria** podemos decirle al sistema que sólo guarde en memoria una sesión de Konqueror, por más que tengamos muchas ventanas abiertas. Esto hace que el sistema tenga más memoria RAM libre y, por consiguiente, todo funcione de manera más fluida. De forma predefinida, esta opción viene definida **Solo para el administrador de archivos**, pero podemos aplicarla al navegador también si marcamos la opción **Siempre**. Lógicamente, esta mejora del rendimiento no la notaremos si utilizamos Firefox o cualquier otro navegador web. Otro recurso: en la zona **Precargar**, po-

demos indicarle al sistema que guarde en memoria un número determinado de sesiones de **Konqueror**. Entonces, cuando cerramos una ventana, en realidad el programa queda en memoria y cuando volvemos a abrirlo aparecerá frente a nuestros ojos más rápidamente. Usemos esto con cuidado y sólo si disponemos de suficiente memoria RAM.

### Quitar artilugios visuales a GNOME para obtener mejor rendimiento

Parece que los programadores de GNOME priorizaron el rendimiento del sistema a la cantidad de funcionalidades. Y esto lo vemos cuando vamos al Centro de Control GNOME (Figura 11). No hay mucho que modificar para mejorar el rendimiento, pero hay algunas opciones medio escondidas que quizá nos ayuden. He aquí las mejores:

- En **Preferencias/Administrador de archivos** podemos deshabilitar todas las previsualizaciones si cada vez que entramos a un directorio tenemos que esperar que se dibujen en pantalla. Simplemente debemos ir a la solapa **Vista Previa** e inhabilitar lo que no necesitamos.

Algunas distribuciones configuran programas para que se carguen automáticamente al inicio de GNOME. Para ver si tenemos alguno y así analizar si es necesario deshabilitarlo o no, debemos ir a **Preferencias/Avanzadas/ Sesiones/Programas al inicio**.

El panel de GNOME a veces tiene **applets** que consumen mucha memoria RAM y hacen que nuestra computadora funcione más lento. Podemos fijarnos si tenemos el panel muy cargado y sacar applets que no utilicemos si presionamos el botón derecho del mouse sobre ellos y elegimos **Quitar del panel**.

Finalmente, en **Preferencias/Avanzadas/Aplicaciones Preferidas**, podemos definir cuáles serán nuestras herramientas preferidas para chequear e-mails y navegar en la Web. A veces, Evolution y Mozilla son

## ORGANIZAR SOLAPAS

¿Sabía que puede utilizar el botón del medio del mouse para mover solapas en aplicaciones como Konqueror y Konsole? ¡Es una excelente forma de organizarlas!

muy pesados para el sistema y conviene cambiarlos por las herramientas que veremos más adelante.

## Cómo crear combinaciones de teclas personalizadas en KDE

No cabe duda de que operar el sistema desde el teclado es mucho más rápido que usar el querido mouse. KDE ofrece un sinnúmero de combinaciones de teclas para llevar a cabo tareas de forma mucho más rápida, como por ejemplo **ALT + F2**

## SIEMPRE ES MEJOR LO DE ADENTRO QUE LO DE AFUERA

Si tenemos poca memoria RAM y usamos **KDE**, siempre tengamos especial preferencia por las herramientas incluidas en este sistema. Usemos **KOffice** en lugar de **OpenOffice**. Usemos **Konqueror** en lugar de **Firefox**. Esto se debe a que, generalmente, estas aplicaciones funcionan mucho más rápido porque ya hay una gran cantidad de componentes y librerías cargados en memoria RAM por el mismo entorno **KDE**.



Figura 11. Desde el Centro de Control GNOME podemos deshabilitar varias opciones para hacer que el sistema funcione más rápido.

para ejecutar un comando. Lo más interesante de todo esto es que KDE nos permite crear nuestras propias combinaciones de teclas personalizadas.

Para esto, presionamos **ALT + F2** y ejecutamos el comando **kcontrol**. Luego vamos a la sección **Regional y Accesibilidad (Regional and Accessibility)** y seleccionamos la opción **Accesos rápidos de teclado (Keyboard Shortcuts)**. Aparecerá un listado con muchísimas acciones que podemos automatizar con las combinaciones de teclas que más nos gusten.

## Protocolos soportados por Konqueror

Muchas personas no saben que Konqueror es un excelente cliente de múltiples protocolos de red.

Generalmente, utilizamos los dos más populares como **http://** (para acceder a un sitio web) y **file://** (para navegar por

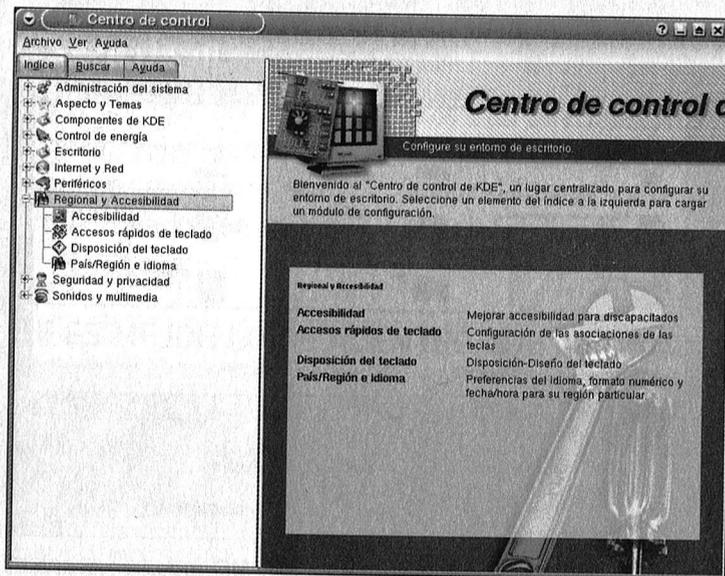


Figura 12. Panel de configuración de accesibilidad y opciones regionales en KDE.

## RENOMBRAR SESIONES

Para cambiar el nombre de una sesión en la terminal **Konsole**, simplemente debemos hacer doble clic con el botón izquierdo del mouse sobre la pestaña correspondiente y presionar la combinación de las teclas **ALT + CTRL + S**.

el sistema de archivos, como vemos en la Figura 13). Pero hay varios adicionales, como los siguientes:

- **ftp://**: para acceder a un servidor FTP remoto.
- **nfs://**: para acceder a un servidor NFS remoto.
- **samba://**: para acceder a un servidor Windows remoto.
- **audiocd://**: para visualizar el contenido de un CD de audio como archivos y además permite **ripear** archivos.
- **camera://**: para visualizar el contenido de una cámara de fotos.

## Protocolos soportados en Nautilus

Al igual que Konqueror, Nautilus nos permite acceder a diferentes unidades y espacios de trabajo con tan sólo indi-

car el protocolo correspondiente en la barra de dirección (Figura 14). Algunos protocolos útiles son:

- **computer://**: son todas las unidades locales.
- **file://**: archivos locales.

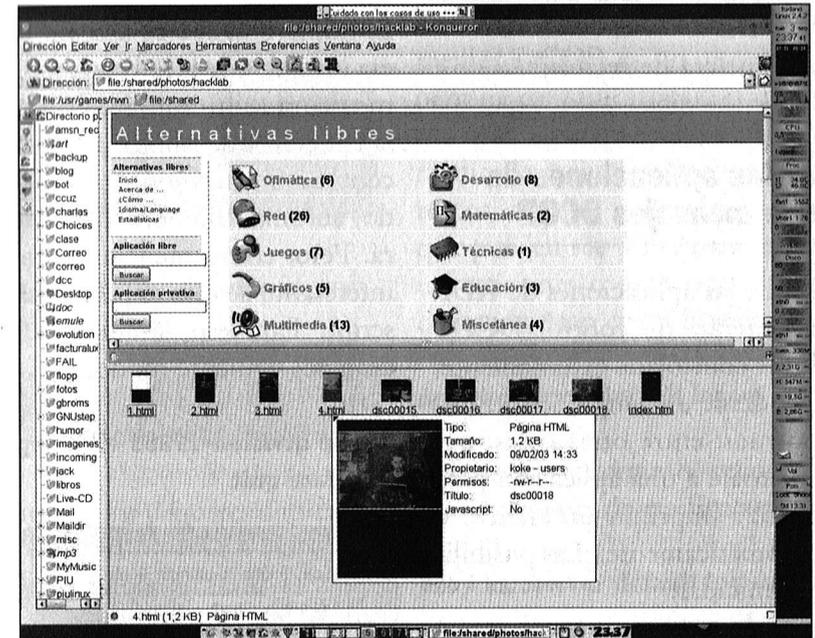


Figura 13. Konqueror puede acceder a diferentes medios y sistemas de archivos de red como si se trataran de directorios locales.

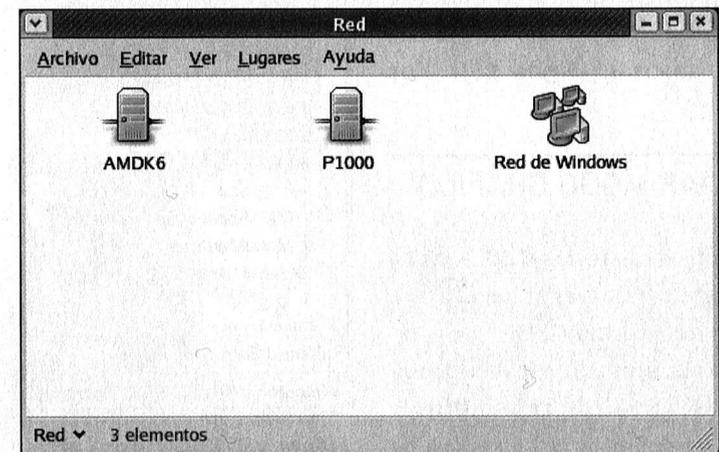


Figura 14. Nautilus, al igual que Konqueror, puede acceder a unidades de red remotas de manera totalmente transparente.

- **ssh://**: transferencia de archivos encriptada por SSH.
- **smb://**: acceso a equipos y redes Windows.
- **network://**: todas las unidades de red accesibles desde Nautilus.
- **burn://**: espacio virtual para grabar CDs y DVDs.
- **trash://**: Papelera de reciclaje.

## Controlar las aplicaciones de KDE desde mensajes DCOP

La mayoría de las aplicaciones de KDE posee una interfaz de control DCOP que puede ser controlada externamente, por ejemplo desde un simple script. Entonces podemos, entre otras cosas, enviarle un mensaje a una aplicación para que comience a imprimir un archivo o se cierre automáticamente. ¡Las posibilidades son ilimitadas!

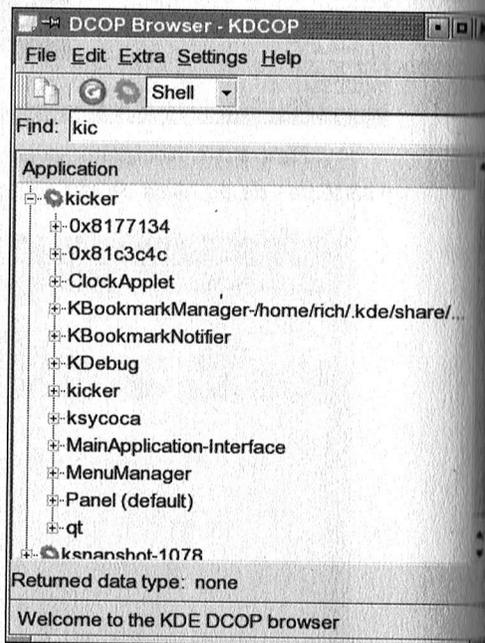
El primer problema es cómo saber qué mensajes acepta cada aplicación, y para esto podemos utilizar una herramienta llamada **kdcop**, que muestra en pantalla información DCOP de todas las aplicaciones KDE abiertas (Figura 15). Por

ejemplo, si estamos corriendo la aplicación **QTVision**, veríamos entre sus entradas una similar a la siguiente:

```
qtvision-19283 => MainApplication-Interface => void quit()
```

El número **19283** es una ID única para esa instancia de la aplicación. Lo que vemos como último parámetro es **quit()**, la función que obliga a la aplicación a cerrarse. Si hacemos clic sobre ese método, automáticamente **QTVision** se cerrará. Pero, como mencionamos antes, lo interesante de esto es poder usarlo en un script. Para eso, usamos el comando **dccop** del siguiente modo:

```
> dccop qtvision-19283 MainApplication-Interface quit
```



**Figura 15.** KDCOP nos permite develar los misterios ocultos en las entrañas de las aplicaciones de KDE.

## Utilizar transparencias en GNOME en distribuciones basadas en Debian

La primera medida es editar el archivo de configuración **Xorg.conf**, disponible en **/etc/X11/xorg.conf**. Allí buscamos la siguiente sección:

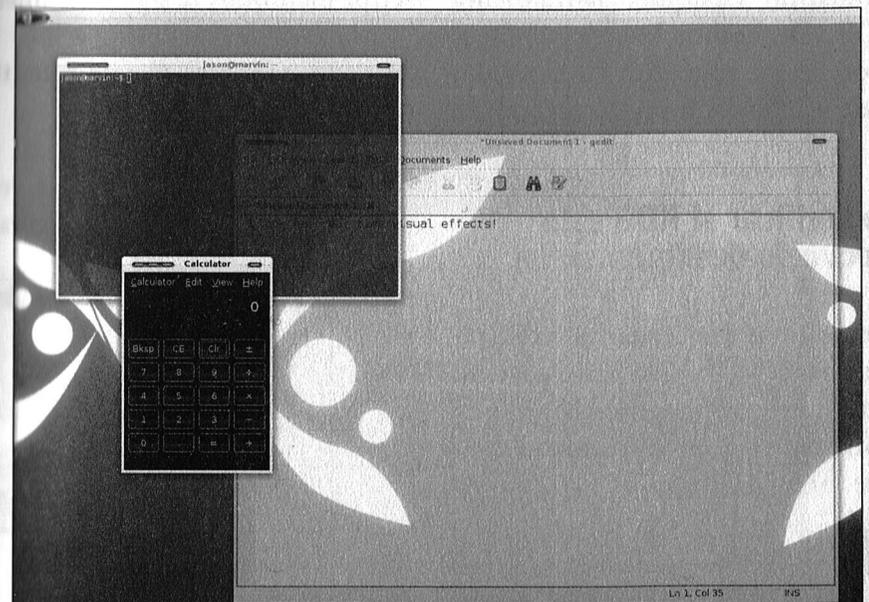
```
Section "Extensions"
Option "Composite" "Enable"
EndSection
```

Y luego en la sección **drivers** ingresamos la siguiente línea:

```
Option "RenderAccel" "true"
```

Además, para instalar los paquetes siguientes:

```
apt-get install xcompmgr transset
```



**Figura 16.** Un escritorio GNOME con las transparencias habilitadas.

Una vez terminado, reiniciamos el servidor gráfico con la combinación de teclas **CTRL + ALT + RETROCESO** y listo (Figura 16).

## Hacer magia con DCOP

Como ya hemos mencionado anteriormente, DCOP es un sistema que nos permite comunicarnos con las aplicaciones que se encuentran en ejecución por medio de mensajes. A continuación veremos algunos ejemplos de mensajes que pueden ser útiles para usar en ciertas ocasiones o en scripts. El siguiente comando **dccop** cierra la sesión de KDE:

```
dccop kdesktop default logout
```

Si queremos cerrar la sesión sin pasar por la ventana de confirmación:

```
dccop ksmserver default logout 0 -1 -1
```

## REINICIAR MODO GRÁFICO

La combinación de las teclas **CTRL + ALT + SUPR** nos permite reiniciar el modo gráfico de modo no convencional.

Al utilizarla, lo que se hace es matar todos los procesos, volver a cargar **X11.Org** y volver a cargar el gestor de escritorios predeterminado del sistema.

## III AJUSTAR A LA VENTANA

Podemos hacer que el contenido de una terminal se ajuste automáticamente al tamaño de la ventana (algunas distribuciones ya incluyen esta funcionalidad de manera predeterminada). Para esto, debemos editar el archivo `/etc/bashrc` y al final de éste incluir la siguiente línea:

```
shopt -s checkwinsize
```

Para maximizar una ventana:

```
dcop konsole-7266 konsole-mainwindow#1 maximize
```

Para bloquear la pantalla (**lock screen**):

```
dcop kdesktop KScreensaverInterface lock
```

Si combinamos DCOP con scripts de Perl, por ejemplo, podremos controlar y automatizar cualquier herramienta de KDE. De hecho, las posibilidades que nos brinda son ilimitadas y sólo dependen de nuestra imaginación.

## Cómo verificar si KDE está en ejecución desde un script

Hay veces en que necesitamos verificar si el entorno KDE está en ejecución desde un script, y podemos hacerlo de dos formas. Una es con `dcop` del siguiente modo desde bash:

```
if dcop kdesktop >/dev/null 2>&1  
; then
```

```
echo KDE esta corriendo;  
else  
echo no esta corriendo  
fi
```

La otra forma consiste en verificar directamente el contenido de la variable de entorno `$KDE_FULL_SESSION`, que se crea cuando se inicia el sistema KDE.

## Utilizar diálogos de KDE desde nuestros scripts

Si nuestros scripts se ejecutan bajo el modo gráfico, podemos condimentarlos visualmente con diálogos de KDE para que el usuario ingrese la información. Esto lo podemos llevar a cabo sencillamente con `kdiallog`, un comando incluido en KDE para este propósito. Veamos un ejemplo. Si en nuestro script incluimos la siguiente línea:

```
kdiallog --password "Please enter the  
server access code:"
```

Al usuario se le presentará una ventana como vemos en la siguiente figura:

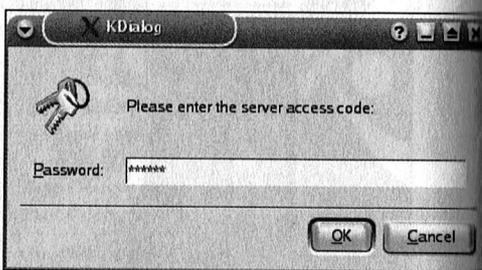


Figura 17. Ventana de ingreso de información generada por la herramienta `kdiallog`.

El parámetro `--password` hace que cada carácter que el usuario tipea sea representado por un símbolo `*` para poder ingresar información confidencial como una clave de acceso. Podemos sacarlo si queremos que el usuario vea en la ventana de diálogo exactamente lo que está tipeando. `KDialog` devolverá el dato tipeado por el usuario, que podemos guardar, por ejemplo, en una variable de un script:

```
clave=`kdiallog --password "Enter  
the password"`  
echo $clave
```

Por último, `KDialog` le ofrece dos botones al usuario. Uno llamado **OK** y otro llamado **Cancel**. Podemos controlar cuál de los dos botones fue presionado simplemente controlando la variable de estado de salida del comando `kdiallog` (llamada  `$?` ). Si una variable posee como dato un `0` (salida

sin error), quiere decir que el usuario presionó **OK**. En caso contrario (cualquier valor diferente a cero), habrá presionado **Cancel**. Veamos un ejemplo:

```
kdiallog --password "Por favor  
ingrese el codigo de acceso"  
if [ $? = 0 ]; then  
echo "usted presiono: OK"  
else  
echo "usted presiono:  
Cancel"  
fi
```

## Cómo cambiar el gestor de ventanas en KDE

Es posible utilizar el escritorio KDE con otro gestor de ventanas de nuestro agrado, como pueden ser **ICEWm** (Figura 18) o **Enlightenment**.

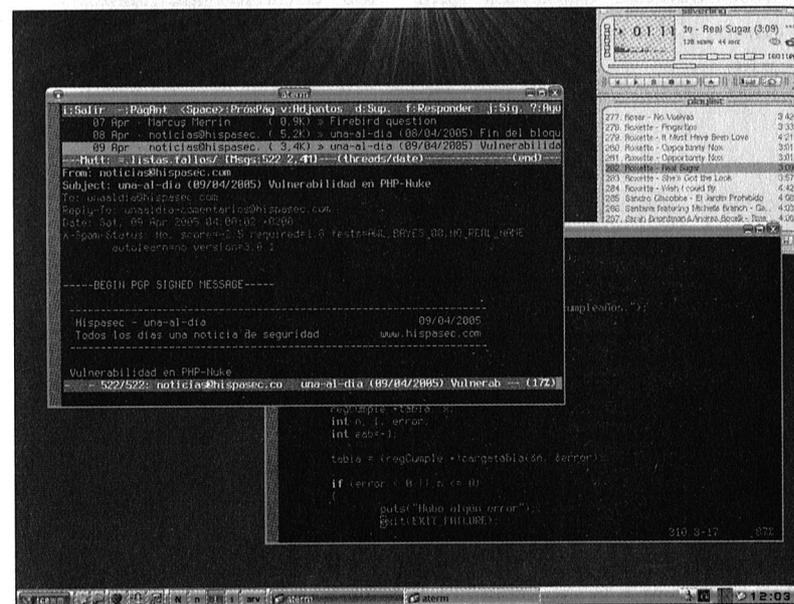


Figura 18. ICEWm es un excelente gestor de ventanas que consume muy pocos recursos.

Para lograr esto tenemos que definir la variable de entorno **KDEWM** con el nombre del gestor de ventanas que queremos que se inicie cuando se carga KDE. Por ejemplo:

```
export KDEWM="icewm"
```

Es importante recordar que las variables de entorno se eliminan cuando se cierra la sesión, por lo que si queremos que esta variable se mantenga cada vez que ingresamos al sistema, debemos editar el archivo **\$HOME/.bashrc** y agregar la línea al final de éste.

## Controlar el cursor del mouse con el teclado en KDE

Esta funcionalidad tuvo sus orígenes en la década de los 80, de la mano de las computadoras **Atari**, que permitían utilizar una combinación de teclas para mover el cursor del mouse en caso de

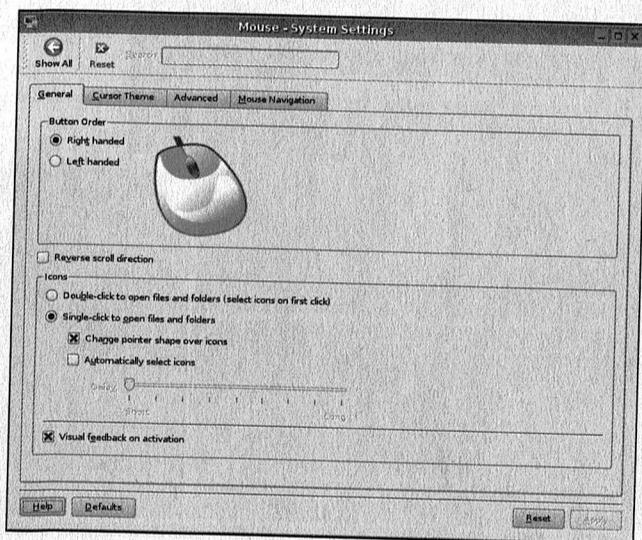


Figura 19. Opciones del mouse en el panel KControl de KDE.

que éste estuviera averiado o no conectado al equipo. En KDE se puede hacer exactamente lo mismo si presionamos la combinación de las teclas **ALT + F12** y usamos las teclas de dirección del teclado para controlar el puntero. Para hacer clic sólo debemos presionar la tecla **ENTER**. Una vez presionado **ENTER**, se deshabilita automáticamente el modo de control del puntero con el teclado. Si queremos efectos permanentes, debemos ir al **Centro de Control KDE/Periféricos/Mouse** (Figura 19) y en la sección de navegación del mouse habilitar la opción **Mover mouse con el teclado**.

## Cómo hacer que el papel tapiz de GNOME se cambie aleatoriamente

El papel tapiz es una de las características visuales más utilizadas por todos los usuarios de sistemas operativos. Luego de algunos días, suele suceder

que nos aburrimos de ver la misma imagen y ya no nos parece tan interesante como antes. Para evitar esto, podemos usar un simple comando que cambiará el fondo de pantalla por uno elegido de forma aleatoria desde un directorio de imágenes del usuario. El comando es el siguiente:

```
gnomeftool -t str -s /desktop/  
gnome/background/picture  
filename `find  
"$BACKGROUND_DIR" -name \*.jpg |  
shuf | tail -n 1`
```

**BACKGROUND\_DIR** es el directorio que contiene las imágenes que se van a utilizar como fondo. Cabe aclarar que éstos deben tener extensión **.JPG**.

## Cómo eliminar el menú de usados recientemente en GNOME

A veces, por cuestiones de privacidad, nos urge la necesidad de eliminar los ítems que aparecen en los menús de **recientemente usados**. Esto lo podemos hacer con el siguiente script:

```
#!/bin/sh  
#  
# Gnome-Terminal Random  
# Background  
# _____  
# Greg Harris  
# <gharris@minuteman.org>  
# 6.13.05  
#  
# Run this script every time you  
# want to update  
# your terminals background  
# image.  
#  
# This is really all you need to  
# setup. Point it to a  
# directory  
# with all of your background  
# images.  
BGPATH=/home/angrylogic/Document
```

```
cat > $HOME/.recently-used <<EOF  
<?xml version="1.0"?>
```

```
<RecentFiles>  
</RecentFiles>  
EOF  
rm -f .gnome2/totem.pls
```

## Script para cambiar el fondo de GNOME-Terminal de forma aleatoria

El siguiente script permite cambiar la imagen de fondo de la terminal de GNOME de manera totalmente aleatoria (Figura 20). Podemos incluirlo en el **cron**, por ejemplo, para que se ejecute en forma periódica. El directorio de imágenes está definido en la variable **BGPATH**.

```
#!/bin/sh  
#  
# Gnome-Terminal Random  
# Background  
# _____  
# Greg Harris  
# <gharris@minuteman.org>  
# 6.13.05  
#  
# Run this script every time you  
# want to update  
# your terminals background  
# image.  
#  
# This is really all you need to  
# setup. Point it to a  
# directory  
# with all of your background  
# images.  
BGPATH=/home/angrylogic/Document
```

```

/Backgrounds
FILTERS="*.jpg *.JPG *.png *.PNG
*.tiff *.TIFF *.gif *.GIF"
PROFILE=Default

# Set counters and lists to
nothing
NUM=0
TOTAL_PICS=0
PIC_LIST=

# Generate the list of pictures
depending on wither or not
# we are doing extension
filtering. If your directory
has
# non-image files, you want to
use filtering.
if [ -n FILTERS ]
then
PIC_LIST=`ls -l $BGPATH`
else

```

```

for FITLER in FILTERS
do
PIC_LIST=$PIC_LIST `ls -l
$BGPATH/$FILTER`
done
fi

# Calculate the total amount of
pictures, and then choose
# a random picture from the
total amount of pictures
let TOTAL_PICS=`echo $PIC_LIST |
wc -w`
let RAND_PIC=$RANDOM%$TOTAL_PICS

# FIXME:
# If anyone knows a better way
of extracting a certain el
ement
# from a list in bash please

```

```

email me. I couldn't find
anything in
# my poking around and I
couldn't get the cut utility
to take
# space as a delimiter
for PICTURE in $PIC_LIST
do
let NUM=$NUM+1
# Loop until we find
our lucky picture in
the list,
and set
# that up as the
background picture.
if [ $NUM -eq $RAND_
PIC ]
then
# This updates
the gconf
key pointing
to your
profile
# and sets up
your
background
image
accordingly
gconftool-2

```

```

type string
set
/apps/gnome-terminal/profiles/
$P ROFILE/background_image
$BGPATH/$PICTURE
fi
done

```

## Cómo cambiar el splash screen en GNOME

Con el siguiente comando podemos cambiar el **splash screen** (Figura 21) de GNOME (la imagen que se muestra mientras se carga el sistema):

```

gconftool-2 --set --type string
/apps/gnome-session/options/s
plash_image
/ruta/a/la/imagen

```

La imagen puede tener cualquier tamaño, y podemos hacer un splash screen que ocupe toda la pantalla si queremos.

## Dominar la terminal Konsole

Konsole es la aplicación terminal incluida en la suite KDE (Figura 22). Veamos algunas funcionalidades que están a nuestra disposición:

- En la esquina inferior derecha de la ventana hay un pequeño botón con un sol amarillo. Si lo presionamos, se creará una nueva pestaña en **Konsole** para que podamos utilizar otra terminal virtual en simultáneo con las que tenemos.

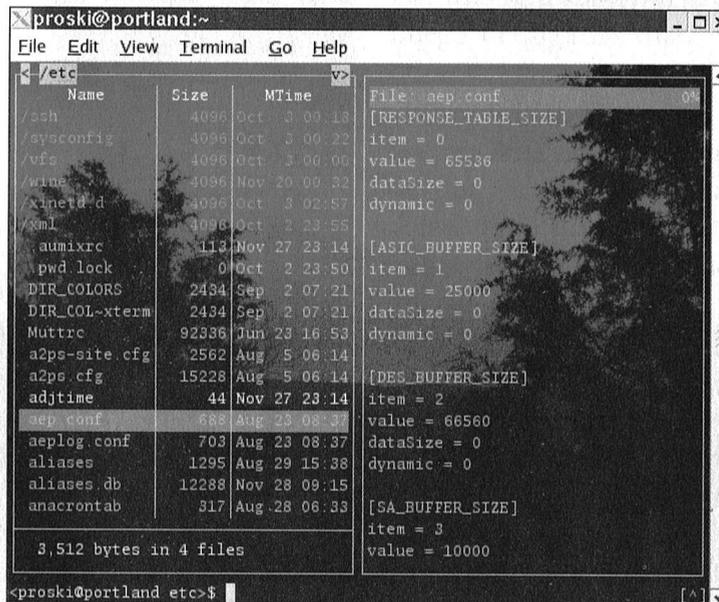


Figura 20. La terminal de GNOME puede mostrar una imagen de fondo.

## ABRIR CON UN COMANDO

Desde la línea de comandos, podemos hacer que KDE abra un documento con la aplicación que le corresponda de forma automática. Para esto, usamos el comando **kfmclient** del siguiente modo:

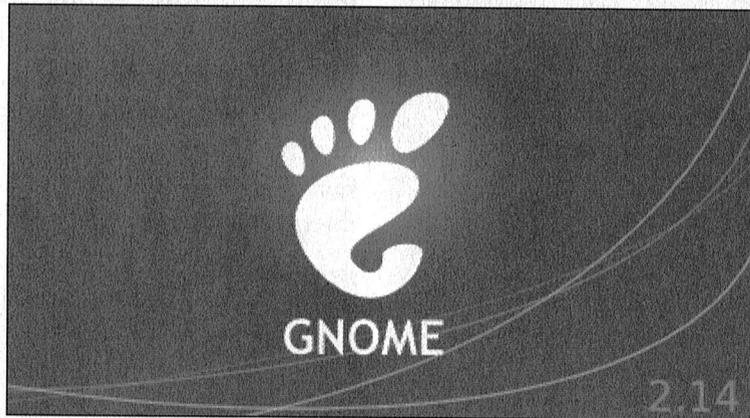
```

kfmclient exec /ruta/al/documento

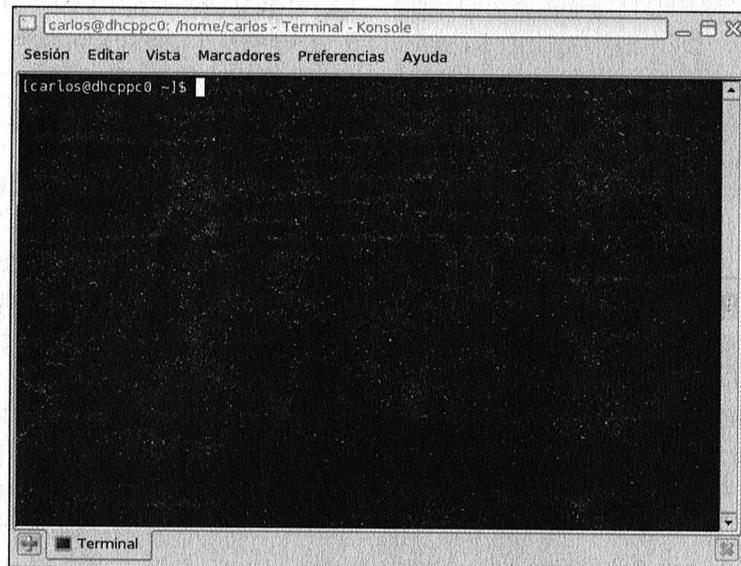
```

- En el menú **Editar**, encontraremos funciones típicas de cualquier editor de texto, como por ejemplo **Copiar**, **Cortar**, **Pegar**, y las funciones para realizar búsquedas.
- En el menú **Marcadores** podremos crear accesos rápidos a los directorios que frecuentamos más comúnmente.
- Por último, en el menú **Preferencias**, encontraremos todas las funciones rela-

cionadas con la configuración de los colores, las fuentes, los tamaños, etcétera. Si trabajar dentro de una ventana no resulta incómodo, podemos presionar la combinación de teclas **CTRL + SHIFT + F**, que automáticamente pasará a modo de **pantalla completa**. Si volvemos a presionar la combinación de estas teclas, automáticamente volveremos al modo ventana.



**Figura 21.** ¿Cansados de la pantalla de bienvenida de GNOME? Con este truco podremos cambiarla a gusto.



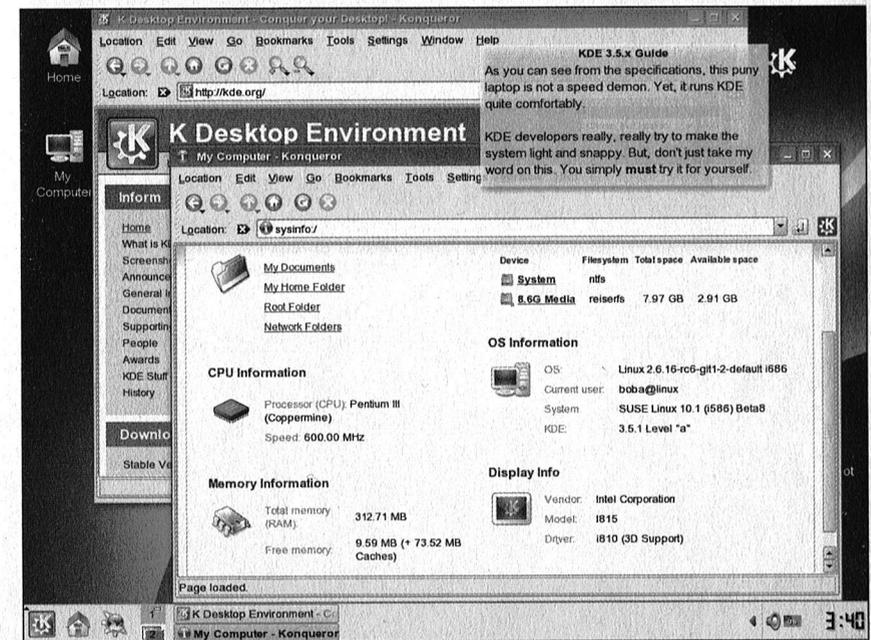
**Figura 22.** La Terminal del proyecto KDE tiene muchas funcionalidades.

## Secretos de KWIN

KWin es el gestor de ventanas del proyecto KDE e incluye algunas funcionalidades muy interesantes para gestionar mejor las decenas de ventanas que pueden estar abiertas mientras navegamos, chateamos, escribimos e-mails, escuchamos música y codificamos un archivo de video. A continuación veremos un resumen de las funcionalidades más interesantes:

- Es posible **enrollar** las ventanas para que no ocupen demasiado espacio en el escritorio. Para esto debemos hacer click con el botón derecho del mouse sobre el borde de una ventana y selec-

- cionar la opción **Shade (Enrollar)** del menú contextual. Luego, cada vez que la ventana no esté activa, se enrollará y cuando el cursor se posicione en uno de sus bordes, la ventana se desplegará.
- Es posible mantener una ventana siempre en primer plano sobre el resto de las ventanas del escritorio si presionamos el botón derecho del mouse sobre uno de sus bordes y vamos al menú **Advanced/Keep Above Others (Avanzado/Mantener sobre las demás)**.
- También es posible esconder los bordes de una ventana si presionamos el botón derecho del mouse sobre alguno de sus bordes y elegimos la opción **Advanced/No Border (Avanzado/Sin borde)**.



**Figura 23.** KWin es el gestor de ventanas oficial del proyecto KDE.

- 1 Es posible modificar la cantidad de escritorios virtuales de KDE desde el Centro de Control KDE.

---

- 2 En sistemas con bajos recursos, GNOME se desempeña mejor que KDE ya que éste último requiere de más memoria RAM.

---

- 3 Aquellos usuarios de Ubuntu que quieran usar KDE, pueden instalarlo desde Synaptic o probar la distribución Kubuntu.

---

- 4 GDesklets es una aplicación para GNOME que nos permite instalar widgets en el escritorio.

---

- 5 El monitor de actividad de GDesklets para GNOME nos permitirá monitorear cuánta memoria RAM se está consumiendo en el sistema en todo momento.

---

- 6 Tilda es una terminal de texto para el entorno visual muy similar a la terminal de comandos del popular juego Quake. Podemos instalarla en Debian y Ubuntu con el comando **apt-get install tilda**.

---

- 7 El proyecto GNOME desarrolla un navegador alternativo llamado Epiphany que es muy liviano e ideal para utilizar en sistemas con bajos recursos.

---

- 8 KOffice es una suite de oficina desarrollada por la gente del proyecto KDE que posee gran compatibilidad con los archivos de Microsoft Office.

---

- 9 Si GNOME y KDE son demasiado pesados para el equipo, siempre se puede optar por un gestor de escritorios como XFCE, que viene incluido en la distribución Xubuntu.

---

- 10 Si los gestores de escritorios son demasiado pesados para el equipo, siempre se puede optar por IceWM, Blackbox o Fluxbox, que son gestores de ventanas que ofrecen muchas funcionalidades para la interfaz visual de Linux y consumen pocos recursos del sistema.

# Trucos para las aplicaciones

El mundo de Linux está repleto de aplicaciones colmadas de funcionalidades. Muchas veces, los usuarios no llegan a descubrir todas las posibilidades de los programas. Por eso, en este capítulo, veremos los trucos para sacarles el máximo provecho a las aplicaciones más populares para Linux.

Evitar correo no deseado con SpamAssassin en Evolution	1
Usar perfiles de visualización en Konqueror	1
Galería de fotos web	1
Gestión de canciones en Konqueror	1
Migrar de Evolution a Thunderbird	
Crear nuestra propia distribución con Garfio	1
Huevo de pascua en OpenOffice	
Sacarle jugo a Firefox	
Mini-trucos para Sylpheed	
Optimizar Firefox	
Cómo usar VMware Player con nuestras propias imágenes de sistemas operativos	
Auto guardado en Thunderbird	
Corrección automática de ortografía en Sylpheed	
Filtrado de correo no deseado con Sylpheed	
Visualizar e-mails HTML en Sylpheed sin usar Claws	
Optimizar el funcionamiento de Firefox y OpenOffice	
Algunos trucos para OpenOffice	
Cómo hacer sesiones de VNC colaborativas	
Control de navegación en Firefox	
Guía de extensiones Firefox	

## Evitar correo no deseado con SpamAssassin en Evolution

Se considera que un mensaje es **spam** si y sólo si el mensaje es no solicitado y, a su vez, es duplicativo. **No solicitado** significa que el receptor no dio un permiso verificable al remitente o emisor para que se le envíe el mensaje. **Duplicativo** significa que el mensaje se envía como parte de un paquete o una colección mayor de mensajes, donde todos tienen el mismo contenido y queda expuesta nuestra dirección a los demás receptores. Un mensaje electrónico es spam si:

- La identidad personal del receptor y el contexto son irrelevantes porque el mensaje es igualmente aplicable a muchos otros receptores potenciales.
- No se puede verificar que el receptor haya dado un permiso deliberado, explícito y aun revocable, para que el mensaje le sea enviado.

Una alternativa posible para protegernos de este flagelo es **SpamAssassin**. La con-

figuración cuenta con varios pasos y tiene algunos requisitos, el primero de los cuales es tener instalado **SpamAssassin**. Éste se encuentra disponible en la mayoría de las distribuciones Linux, y de no ser así, podemos descargar la última versión de su sitio oficial, en <http://spamassassin.apache.org> (Figura 1).

La configuración está seguida por la apertura de nuestro cliente de correo favorito (en este ejemplo usaremos **Evolution**). En él creamos la carpeta llamada **Spam** y dentro de ésta las subcarpetas **FilteredSpam**, **MissedSpam** y **NonSpam**. Luego nos dirigimos a la configuración de nuestro cliente y abrimos la sección de reglas, donde creamos una nueva que llamaremos **SpamAssassin**. Ahora debemos seleccionar (según nuestro cliente de correo) el espacio asignado al correo basura en kilobytes. Le asignamos un valor de **250000**. Dejamos todas las opciones como están y le indicamos que use el binario de **SpamAssassin** (`/usr/bin/spamassassin`). Ahora creamos un segundo filtro de la misma

forma que el anterior y lo llamamos **SpamFilter**. Definimos la regla: cualquier cabecera con el texto **X-Spam-Status: YES** en el cuadro de cabecera, moverla a la carpeta **FilteredSpam**.

Siempre hay que tratar que en el listado de reglas éstas sean las primeras que se activen y luego las demás. De esta forma, una vez que ejecutemos **Evolution**, veremos cómo el Spam se va a las carpetas seleccionadas.

Ahora viene la parte más interesante del filtrado y de la que ninguna PC se escapa: la de nuestra intervención. En muchas ocasiones veremos que hay correo basura que se escapa del filtro y en otras (no tantas) hay correos que no son spam y se mueven a la carpeta de spam.

Para esto tenemos dos funciones: cuando se escapa spam del filtro tenemos que moverlo a la carpeta **MissedSpam**, y los correos que no son spam y que se fueron a la carpeta de spam por error los pasamos a la carpeta **NonSpam**. Ahora sí, podemos ir a la carpeta **SpamAssassin** y borrar sin problemas todos los correos no solicitados.

Esto no termina acá. Para que nuestro filtro adquiera el aprendizaje para filtrar los correos, cerramos el programa cliente de correo y vamos a una consola con el usuario **root**. Allí debemos ejecutar la siguiente orden:

```
#sa-learn --spam --dir  
/home/ariel/.evolution/mail/local.spam.  
directory/MissedSpam/*
```

De esta forma, el programa aprenderá a determinar mejor si un mensaje se trata de correo basura o no. Mientras más

## III FONDO PROPIO EN FIREFOX

Para utilizar una imagen de fondo propia en las cajas de herramientas de Firefox, debemos guardarla con el nombre de archivo **background.gif** y ubicarla en el directorio **chrome** del directorio de preferencias del usuario de Firefox.

Luego, en el archivo **userChrome.css** que se encuentra en el mismo directorio (o si no existe debemos crearlo), tenemos que agregar el siguiente texto:

```
menubar, toolbox, toolbar, .tabbrowser-  
tabs {  
    background-image: url("background-  
-gif") !important;  
    background-color: none !important;  
}
```

correos tengan las carpetas de aprendizaje, más útil será el filtro, por lo que es conveniente juntar bastantes mensajes y luego ejecutar el comando. Cabe aclarar que es importante **no utilizar el comando de aprendizaje** cuando el cliente de correo está abierto.

## Usar perfiles de visualización en Konqueror

Como seguramente sabemos, en Konqueror existen distintos modos de visualización. A veces necesitamos ver simplemente el nombre del archivo y su tamaño, y otras veces es más útil tener una previsualización automática del contenido de todos los archivos. En el menú **Previsualizar** (dentro de **Ver**) podemos tildar

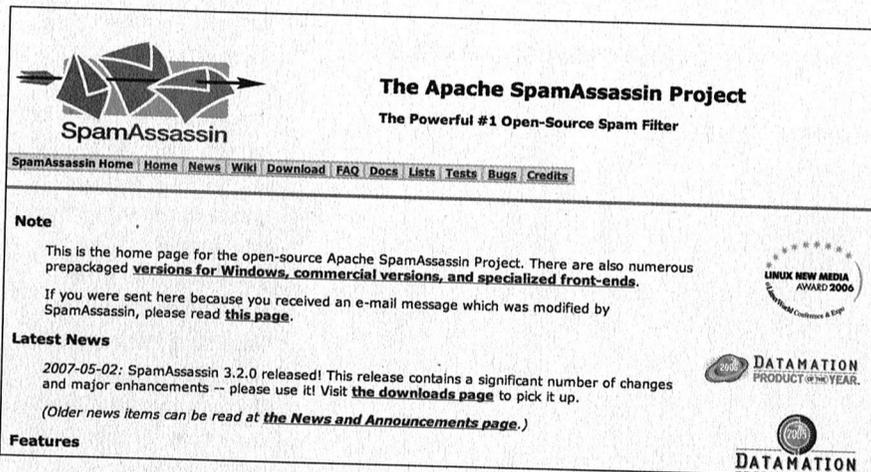
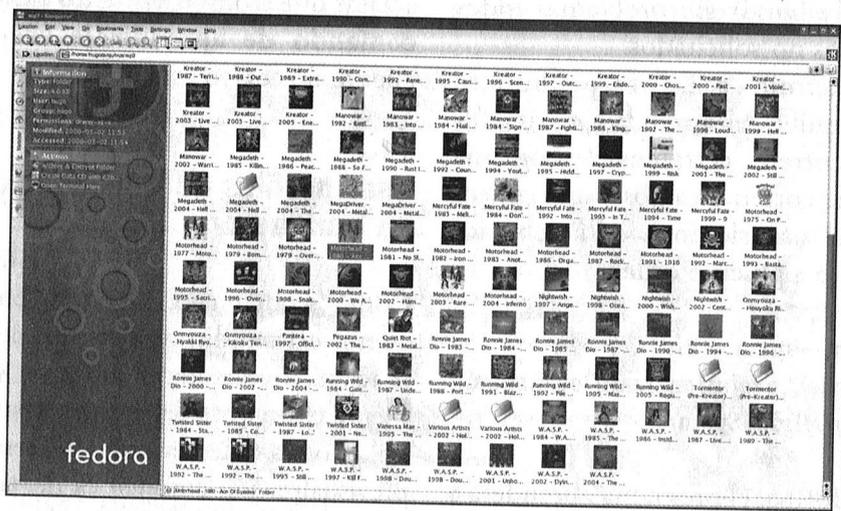


Figura 1. SpamAssassin es un script en Perl que nos ayudará a evitar todo el correo no deseado.

qué elementos serán previsualizados. Si tenemos una computadora rápida podemos seleccionar todas las opciones sin una pérdida apreciable de rendimiento. Especial atención merece la opción de previsualizar archivos **HTML**. No siempre conviene tener activada esa opción ya que algunos documentos HTML tienen referencias a elementos que no tenemos en nuestro disco y se provoca un cartel de error por cada archivo HTML con estas características. Cada directorio conserva automáticamente las preferencias de visualización. Por ejemplo, si en un directorio con fotos establecemos que se previsualicen las imágenes, cada vez que volvamos a abrir ese directorio se verán los contenidos de las fotos en lugar de sólo los nombres de los archivos (**Figura 2**). La información relacionada con la forma en la que tiene que verse cada directorio se almacena en un archivo de texto llamado **.directory**. Por este motivo, esta opción de guardado automá-

tico sólo funciona en los directorios donde tenemos permiso de escritura. Tampoco tenemos que conformarnos con ver un solo recurso a la vez, sino que podemos dividir horizontal o verticalmente el área de visualización en dos ventanas y ver distintos recursos en cada sección. Quienes hayan usado el **Norton Commander** (o **Midnight Commander**, el clon de Linux), podrán apreciar inmediatamente las ventajas de esta configuración. Las operaciones de copiado o movimiento de archivos se ven facilitadas al tener a la vista el origen y el destino de éstos. Cada ventana retiene su configuración de manera independiente. Esta visualización se obtiene desde el menú **Ventana** en la opción **Horizontal**. Ya que estamos agregando componentes al Konqueror, hay que mencionar también la opción de acoplar una ventana de terminal para poder ejecutar comandos manualmente. Si estamos conformes con el aspecto y la funcio-



**Figura 2.** Konqueror se puede configurar para que cada directorio respete sus propias opciones de visualización.

### QUITAR EL FORMATO BOLD

Para quitar el formato **bold** (negrita) de los nombres de las solapas activas, podemos editar el archivo **userChrome.css** ubicado en el directorio **chrome** del directorio de preferencias del usuario. En él debemos agregar el siguiente código:

```
tab[selected="true"] {
    font-weight: normal !important;
}
```

nalidad que logramos, podemos guardar esa configuración bajo un perfil para poder invocarla más adelante o simplemente dejarla como predeterminada. Los perfiles se administran desde **Preferencias/Configurar perfil de vista**. Para usar un perfil ya grabado, lo seleccionamos en **Preferencias/Cargar perfil de vista/Nombre del perfil**. Algunas distribuciones de GNU/Linux vienen con algunos perfiles preconfigurados, que podemos usar o adaptarlos a nuestros gustos y necesidades.

### Galería de fotos web

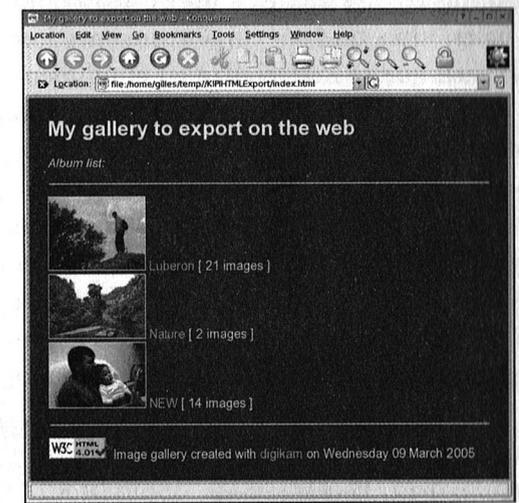
Si bien Konqueror nos permite previsualizar nuestras fotos, es bueno saber que es muy fácil hacer una página web a partir de nuestras fotos. Simplemente tenemos que elegir la opción **Galería de imágenes** dentro del menú **Herramientas**. Esto nos conduce a una ventana de diálogo donde podemos personalizar la apariencia de la página web que vamos a generar. Podemos seleccionar el color de fondo, de las

letras, el título de la página, la cantidad de fotos que incluimos por línea y otras opciones. También podemos agregar **comentarios** a las fotos, con un archivo de texto con el siguiente formato:

**nombre\_foto1:**  
**comentario.**

**nombre\_foto2:**  
**otro comentario.**

Luego, en el menú de creación de la página web, tendremos que indicar el nombre de este archivo de texto. Konqueror nos generará las previsualizaciones y la página html correspondiente (**Figura 3**).



**Figura 3.** Ejemplo de galería de fotos creada en Konqueror.

### Gestión de canciones en Konqueror

Una característica única de Konqueror es la posibilidad de administrar el contenido de CDs de música como si fueran

archivos comunes. Cuando tenemos un CD de música en la lectora de CD, sólo hay que escribir **audiocd:/** en la barra de direcciones. En la ventana principal veremos los archivos de música en el formato original y unas carpetas nuevas. De estas carpetas, la más significativa es la llamada **Ogg Vorbis**. Adentro encontraremos los mismos archivos que tenemos en el directorio raíz del CD pero en formato Ogg Vorbis, similar a MP3 pero con mejor calidad y sin restricciones legales (Figura 4).

Si queremos copiar estos archivos a nuestro disco rígido, lo podemos hacer como si fueran cualquier archivo corriente. Hay que tener en cuenta que en realidad, esos archivos no existen como tales, sino que son una representación visual que se **materializa** en el momento de copiarlos a un lugar físico concreto (como un disco rígido, una llave USB, etcétera). Cuando se los copia, en realidad Konqueror convierte el archivo de audio en un archivo Ogg Vorbis en el directorio de destino. Por esta razón, la copia tarda más que lo habitual, entre tres y cinco minutos, según el tamaño

del archivo y la velocidad de nuestra máquina. La alternativa a esto es usar un programa especial para convertir archivos de audio en archivos Ogg Vorbis, pero el proceso de **encoding** (transformación a Ogg Vorbis) tardaría lo mismo.

## Migrar de Evolution a Thunderbird

Muchos usuarios de Evolution suelen experimentar problemas de diversa índole, como pérdidas de e-mails y cierres inesperados de la aplicación, entre otras cosas. Una opción posible es usar Thunderbird como cliente de correo, y para importar los e-mails de Evolution sólo tendremos que hacer unos pocos procedimientos. Primero vamos a crear en nuestra instalación nueva de Thunderbird las mismas carpetas que teníamos en Evolution (es muy importante que escribamos sus nombres respetando mayúsculas y minúsculas). Luego entramos en el directorio **.evolution** de nuestro directorio personal y buscamos el archivo **.mbox** correspondiente a la carpeta en Evolution.

Veamos un ejemplo concreto: hay una carpeta en Evolution 2.0 con el nombre **ariel** que se quiere pasar a Thunderbird. El comando es muy simple:

```
cp /home/ariel/.evolution/mail/
local/ARIEL
/home/ariel/.thunderbird/
664dfuau.default/Mail/Local
Folders/ARIEL
```

De esta manera, copiamos la carpeta completa, abrimos el programa Thunderbird y listo, nos vamos a encontrar con todos nuestros e-mails.

## Crear nuestra propia distribución con Garfio

Garfio es un sistema que posee una infraestructura muy simple (desarrollada por Mauro Torres) para armar distribuciones a medida, orientada a grupos concretos de usuarios como universidades, colegios, cursos, empresas, etcétera. Gracias a esta potente herramienta (GNU/GPL) podremos desarrollar distribuciones de GNU/Linux con las siguientes características:

## MÁS ARCHIVOS RECIENTES

En general, OpenOffice sólo muestra cuatro ítems en la lista de archivos recientemente usados. Para modificar esto, debemos cerrar todas las ventanas de OpenOffice y editar el archivo **user\registry\data\org\openoffice\Office\Common.xcu**. Allí tenemos que buscar la línea **>node oor:name="History"<** y justo debajo agregar el siguiente texto:

```
>prop oor:name="PickListSize" oor:typ
e="xs:int" < >value<10>/value< >/prop<
```

En esta línea, debemos cambiar el **10** por la cantidad de ítems que queremos que aparezcan en ese menú. Luego guardamos el archivo y volvemos a abrir OpenOffice.

- Compatibilidad con todas las distribuciones más conocidas (Debian, Suse, RedHat, Mandriva, etcétera).
- Configuración del hardware y software en forma automática.
- Posibilidad de tener una distribución LIVECD (e instalable) en 15 minutos.
- 2 GB de aplicaciones en un CD de 650 MB.
- Rapidez al iniciar la distribución desde el CD.
- Múltiples opciones de arranque.
- Posibilidad de escribir en toda la distribución corriendo en modo Live CD (desde el CD).
- Es muy personalizable.

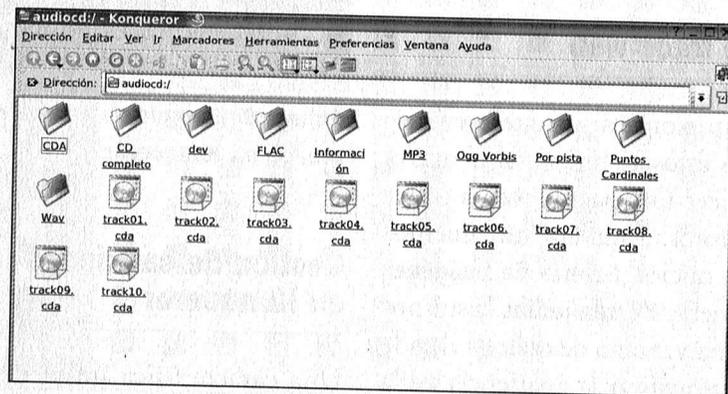
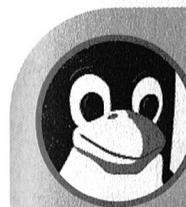


Figura 4. En Konqueror podemos navegar un CD de audio y convertir sus tracks en archivos Ogg Vorbis de manera sumamente práctica.



Otra forma de crear nuestra propia distribución es con una herramienta web en línea llamada NibleX (<http://custom.niblex.net/>). En este sitio podemos definir muchas opciones sobre las características del sistema (incluso el fondo de pantalla), y luego descargar una imagen ISO con la distribución personalizada según lo que hayamos elegido.

Ahora vamos a crear nuestra propia **Live CD**. Para poder comenzar la creación tendremos que obtener todos los paquete-

tes de aplicaciones necesarias para su correcto funcionamiento. Éstos se pueden descargar del sitio oficial de Garfio en [www.garfio.org.ar](http://www.garfio.org.ar) (Figura 5).

Los paquetes necesarios son **Garfio Versión 0.2 Testing**, un núcleo con soporte para **squashfs** y **supermount** (podemos utilizar el de Garfio, que ya viene configurado para usarse en la mayoría de las computadoras), las utilidades de **squashfs** para comprimir la distribución, una distribución instalada llamada **A** desde este momento y, en otra partición o directorio (con **chroot**), la distribución que queremos transformar en Live CD (que llamaremos **B**). Ésta es la que debemos personalizar. Una vez terminado todo este proceso, recién se podrá generar el Live CD. Recordemos que todos los pasos siguientes se realizan desde la distribución **A**.

El primer paso es crear una estructura de directorios donde vamos a trabajar, que se puede crear como quiera cada

usuario. En este caso, nos centraremos en una de ejemplo. Creamos el siguiente directorio en **/mnt**:

```
/mnt/
-> sources
-> master
-> iso
```

```
# mkdir -p /mnt/{sources, master, iso}
```

Descomprimos las fuentes de Garfio en el directorio **master**:

```
# tar -xvzf GARFIO-0.2-Testing.tgz -C
/mnt/master/
```

Y obtendremos la siguiente estructura de directorios en **/mnt/master**:

```
/mnt/master/
-> boot
-> garfio
-> Distro
```

Ahora montamos la distribución **B** en **/mnt/sources**:

```
# mount /dev/hdxx /mnt/sources
```

Una vez que tenemos montada la distribución **B**, comprimimos para que pueda ser ingresada en el tamaño de un CD-Rom. Siempre debemos recordar que la relación es de 2 GB en 650 MB. Para lograr esta función necesitamos el binario **mksquashfs**.

```
#mksquashfs /mnt/sources/ /mnt/
master/Distro/MIDISTRO.squashfs
```

Debemos agregar la extensión **squashfs** ya que es el tipo de compresión que utiliza Garfio. También soporta otro tipo de compresión (**CLOOP**) pero no es recomendable. Ahora vamos a configurar el archivo **/mnt/master/garfio/custom/custom.conf** de la siguiente forma:

```
#Nombre de la distribución B
comprimida, sin la extensión
.squashfs.
NombreDistro="MIDISTRO"

#Tamaño de la distribución B,
en bytes. du -ac /mnt/sources
TOTAL=1938544

#Nombre de la maquina, puede
ser el nombre de la
Distribución B.
HOSTNAME=MIDISTRO

#Lenguaje por defecto
LANGUAGE=es
```

## III EL ICONO DE FIREFOX

Para cambiar el icono de la ventana de Firefox, sólo tenemos que ir al directorio **chrome** del directorio de preferencias del usuario y crear un nuevo directorio llamado **icons**. Dentro de él hay que crear otro directorio llamado **default**, y allí dentro podemos guardar el icono en formato **XPM** con el nombre **main-window.xpm**.

Si usamos el núcleo y los módulos de Garfio, deberemos colocar el archivo **modules.squashfs** en **/mnt/master/Distro/**:

```
# cp modules.squashfs /mnt/master/
Distro/
```

Y la imagen del núcleo en **/mnt/master/boot/grub/**:

```
# cp vmlinuz /mnt/master/
boot/grub/
```

Podemos editar el menú de arranque del LiveCD editando el fichero **/mnt/master/boot/grub/menu.lst**.

Si queremos colocar un núcleo propio, deberemos colocar su imagen con el nombre **vmlinuz** en **/mnt/master/boot/grub/**, comprimir los módulos utilizando **mksquashfs** y colocarlos en **/mnt/master/Distro/**.

```
# mksquashfs /lib/modules/
modules.squashfs && cp
modules.squashfs /mnt/master/Distro/
```

En cuanto a la instalación y detección de hardware, Garfio las realiza en for-

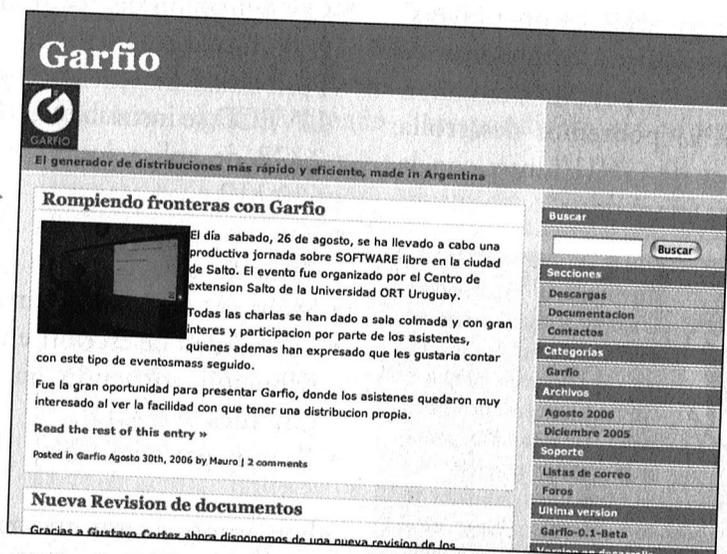


Figura 5. Garfio posee su propio sitio web en [www.garfio.org.ar](http://www.garfio.org.ar). Allí encontraremos scripts adicionales para crear distribuciones personalizadas.

ma automática, por lo que nosotros sólo debemos concentrarnos en la personalización de la distribución. Ahora estamos en condiciones de decir que nuestro Live CD está listo para generar la imagen y luego copiarla a CD. Como último paso, sólo nos resta generar la imagen ISO para grabarla en el CD y a disfrutar.

```
#mkisofs -R -b boot/grub/stage2
-no-emul-boot -input-charset ISO-8859-1
-V NOMBREDETUDISTRO \
-boot-load-size 4 -boot-info-table -o
/mnt/iso/NOMBREDETUDISTRO.iso
/mnt/master/
```

En esta explicación fuimos detallando todos los pasos y comandos a realizar, pero Mauro creó dos aplicaciones que agilizan y automatizan estas tareas que, tal vez a un usuario nuevo en el mundo de GNU/Linux, puedan resultarle dificultosas. Para realizar estas tareas de forma auto-

mática hay que descargar el **Generador Automático** del mismo sitio de Garfio. Éste es un didáctico script, muy intuitivo, que necesitará la aplicación **dialog** en la distribución A. Luego está el **Generador Manual**, que es un script de texto plano donde podremos editar la configuración comentando y descomentando las opciones que vamos a utilizar.

## Huevo de pascua en OpenOffice

En la suite ofimática OpenOffice se encuentran varias características bastante interesantes. Una de ellas es la de poder recrearnos un poco con un juego simple desde nuestra planilla de cálculo (Figura 6). Éstas son características ocultas por los programadores y no hace mucho se había encontrado un denominado **huevo de pascua** en esta misma suite, donde los mismos programadores se daban a conocer ingresando sus nombres y foto.

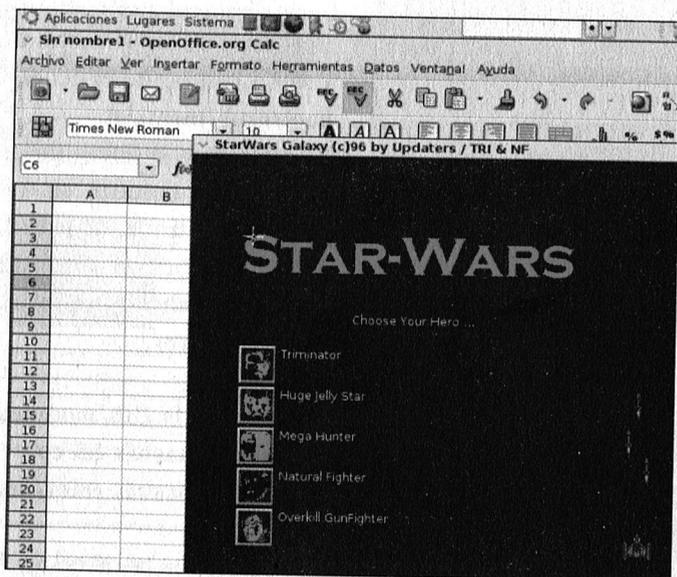


Figura 6. StarWars es un juego escondido en la planilla de cálculo de OpenOffice.



Figura 7. Si tenemos muchas solapas abiertas, acordarse las combinaciones de teclas para controlarlas puede ser de gran ayuda.

Bueno vamos al truco, para lo que tendremos que ingresar en una planilla de cálculo nueva y escribir, en la celda A1, el siguiente texto:

```
=GAME("StarWars")
```

Ahora, a disfrutar de un momento de ocio en nuestra oficina (sin que nos vea nuestro jefe, por supuesto).

## Sacarle jugo a Firefox

En muchas ocasiones encontramos que los usuarios de Firefox no utilizan muchas de las virtudes que tiene incluido este navegador. Quizá sea por costumbre en el manejo de navegadores más simples (que no vale la pena nombrar). Por eso, aquí veremos algunas tips para poder sacarle el máximo provecho y trabajar de forma más rápida (Figura 7).

Comenzaremos por aprender a cerrar las **pestañas** de forma simple y rápida, algo que se puede lograr de tres maneras. Una de ellas es hacer un clic sobre la pestaña con el botón central del mouse. La segunda opción es utilizar la combinación de las teclas **CTRL + F4**, y la tercera alternativa es mediante **CTRL + W**. Siguiendo con las pestañas, una buena opción a la hora de usar una por cada acceso a la Web es mantenerlas

## III ¿QUÉ REPRODUCE AMAROK?

Podemos conectarnos remotamente a nuestro equipo por SSH y, si aMaroK está funcionando, preguntarle qué canción está tocando con el siguiente comando:

```
ssh -i ~/.ssh/id_rsa YOURIP dcoy -user
USUARIO amarok player nowPlaying
```



Thunderbird 2 incluye la posibilidad de aplicarles etiquetas (tags) a los usuarios. Con esta utilidad, podemos organizar de forma sencilla los e-mails en grupos como trabajo, amigos, listas de correo, etcétera. Además, podemos crear nuestras propias etiquetas para personalizar aún más la organización de nuestros mensajes.

ordenadas. Esto se realiza de forma simple con tan sólo arrastrar la indicada al lugar en el que nosotros deseamos dejarla. Y si por el contrario queremos abrir una página nueva sin cerrar la actual, sólo debemos usar **CTRL + T**.

Si deseamos aumentar o disminuir el tamaño de la letra que tenemos en Firefox sólo debemos presionar la tecla **CTRL** y desplazar el botón central del mouse hacia arriba o hacia abajo según corresponda. También podremos utilizar las teclas **CTRL + o** **CTRL -** si es que no contamos con el scroll en el mouse.

A continuación veremos un resumen de combinaciones de teclas:

- Para cambiar de pestaña: **CTRL + TAB**.
- Para ir a la siguiente pestaña: **CTRL + AVPAG**.
- Para regresar una pestaña: **CTRL + REPAG**, **CTRL + Shift + TAB**.

## COPIA DE SEGURIDAD COMPLETA DE THUNDERBIRD

Para hacer una copia de seguridad completa de toda la información relevante de nuestro Thunderbird (e-mails, configuraciones, etcétera) sólo debemos duplicar el directorio **.thunderbird** ubicado en nuestro directorio personal a un directorio llamado, por ejemplo, **thunderbird-duplicado**. Luego debemos comprimirlo con el comando **tar -cvf backup-thunderbird.tar thunderbird-duplicado ; gzip thunderbird.tar**.

- Por medio del teclado numérico: **CTRL + Número de pestaña**.
- Para realizar una búsqueda: **F3**, **CTRL + F**.
- Para posicionarse en la barra de direcciones: **F6**.
- Para ahorrarnos escribir **www.dominio.com** escribimos el nombre del dominio y oprimimos **CTRL + ENTER**. En caso de un dominio **.ORG** debemos presionar **CTRL + SHIFT + ENTER** y si fuera un dominio **.NET** la combinación es **SHIFT + ENTER**.

## Mini-trucos para Sylpheed

Veamos una lista de trucos para este cliente de correo:

- Si tenemos varias cuentas de e-mail, podemos configurarlas en **Configuration/ Edit Accounts**. Luego podemos verificarlas todas juntas presionando el botón **Get All**.
- Un detalle de interfaz: la primera vez que lo usa, todo el mundo se confunde el botón **Send** (enviar) con **Compose** (componer). El primero envía los mensajes a la cola de envío (aquellos que ya fueron escritos pero puestos en espera). El segundo crea un mensaje de correo electrónico nuevo.

- Si tenemos varias cuentas, es recomendable crear algunas carpetas desde el menú **File/Folder/Create New Folder** para organizar mejor los e-mails. Luego, debemos crear una regla para redireccionar los mensajes a esa carpeta. Una forma fácil de hacerlo es en **Tools/Create Filter Rule/By To**. Con esto aparecerá un panel nuevo en donde podemos definir la dirección de correo electrónico que servirá de filtro y la carpeta a la que irán a parar los mensajes filtrados.
- Muchas personas suelen tener problemas para recordar sus nombres de usuario y contraseñas de los servidores de e-mail. Para hacer la depuración y ver en qué nos estamos equivocando, podemos utilizar la ventana de **Log** que incluye Sylpheed (algo que se extraña en otros clientes de correo electrónico populares) presio-

## UBICAR LA BARRA LATERAL

Si no nos gusta la sidebar a la izquierda en Firefox, podemos ubicarla a la derecha. Para esto, debemos editar el archivo **userChrome.css** ubicado en el directorio **chrome** del directorio de preferencias de Firefox de usuario, y allí agregar el siguiente código:

```
hbox#browser { direction: rtl; }
hbox#browser > vbox { direction: ltr; }
```

nando la combinación de tecla **SHIFT + CTRL + L**. Esta terminal también nos permitirá visualizar el progreso de la descarga de correo electrónico con mucho detalle.

- En el menú **Tools** podemos encontrar una agenda (**Address Book**) muy simple de utilizar, compatible con **Jpilot** **LDAP** y tecnología **vCard**.

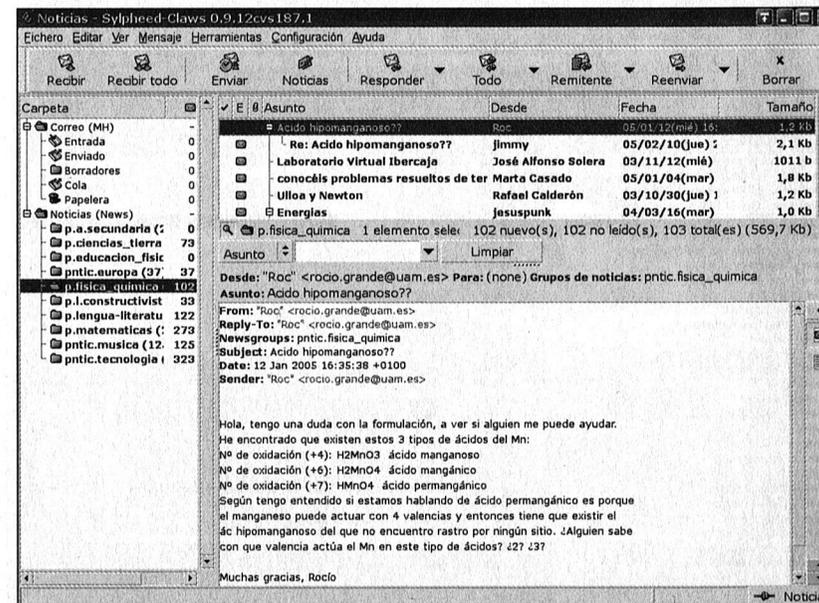


Figura 8. Sylpheed es un excelente cliente de correo electrónico, sumamente liviano e ideal para los sistemas con bajos recursos.

## Optimizar Firefox

Uno de los motivos por los que Firefox consume mucha memoria es porque no libera ciertos recursos al ser minimizado. Esto se puede cambiar de forma fácil. Primero abrimos una pestaña y escribimos la dirección **about:config** (Figura 9). Luego, en la lista enorme que aparece, hacemos clic con el botón derecho del mouse y elegimos **Nuevo/Lógico (New/Boolean en inglés)**. En la cajita que nos aparece escribimos **config.trim\_on\_minimize** y le asignamos el valor **True**.

Reiniciamos Firefox para que el cambio tenga efecto y utilizamos nuestra herramienta preferida para ver que, al ser minimizado, Firefox utiliza menos memoria. Es muy útil para los que siempre lo tenemos que tener abierto, y es un truco que también funciona en Windows.

## Cómo usar VMware Player con nuestras propias imágenes de sistemas operativos

VMware tiene una versión gratuita lista para usar en la mayoría de las distribuciones, aunque con ciertas limitaciones. Una de ellas es que no nos permite crear una instalación propia de un sistema operativo y utilizarlo como máquina virtual (Figura 10). La única opción que tendremos es bajar una determinada imagen de Internet con extensión **.VMX** y así poder usar esa imagen.

Bueno, para todos nuestros lectores, estuvimos investigando y les traemos la solución. No es simple... pero tampoco imposible.

La solución es técnicamente posible y pasa por recurrir a **QEMU**, con la que vamos a generar la imagen del disco

**sistema.vmdk**. Lógicamente, para poder utilizar VMware Player y crear una imagen propia debemos tener instalado, en nuestro GNU/Linux, QEMU y VMware player (éstos son paquetes clásicos en cualquier distribución conocida). Para crear una imagen de 6 GB tendremos que ingresar la siguiente orden desde la consola:

```
qemu-img create -f vmdk
```

```
sistema.vmdk 6G
```

Y luego utilizar nuestro editor de texto preferido para crear el archivo **sistema.vmx**. Para eso podemos ingresar en la consola la siguiente línea:

```
touch sistema.vmx
```

Después tenemos que editarlo para ingresar todo el texto siguiente:

```
config.version = "7"
memsize = "128"
ide0:0.present = "TRUE"
ide0:0.fileName = "sistema.vmdk"
ide0:1.present = "TRUE"
ide0:1.fileName = "/dev/cdrom"
ide0:1.deviceType = "atapi-cdrom"
floppy0.fileName = "/dev/fd0"
Ethernet0.present = "TRUE"
sound.present = "TRUE"
sound.virtualDev = "es1371"
displayName = "sistema"
guestOS = "other26xlinux"
```

Debemos introducir las líneas tal como se indica. Lo único que quizá necesitamos modificar es la línea de configuración del CDROM y de la unidad floppy. Una vez que creamos estos dos archivos, contamos con una máquina virtual que podemos arrancar con **wmplayer**

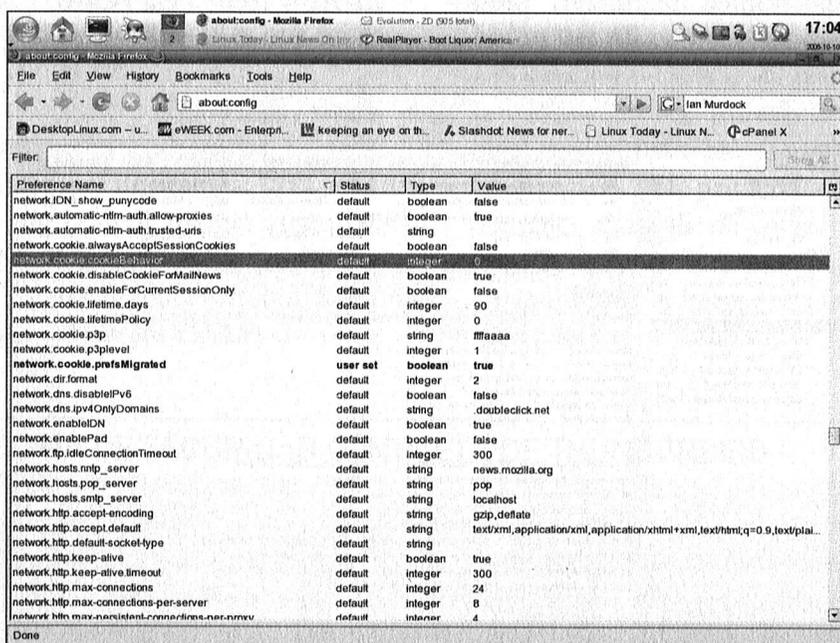


Figura 9. Desde el sistema de configuración de Firefox podemos modificar ciertos parámetros para ajustar su rendimiento.

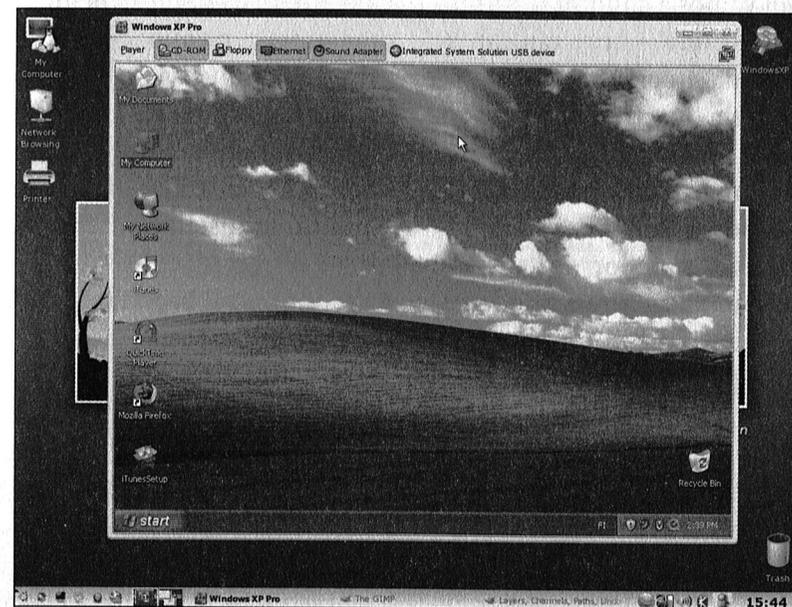


Figura 10. VMware Player es un sistema que nos permite ejecutar un sistema operativo contenido en una ventana gracias a la tecnología de virtualización.

**sistema.vmx.** Con **F2** entramos en la BIOS de la máquina virtual para habilitar el arranque del CD y así poder instalar cualquier sistema operativo utilizando su CD de instalación. Asimismo, podemos utilizar **QEMU-IMG** (del paquete QEMU) para convertir de QEMU (formato **QCOW**) a VMware (formato **VMDK**), como también a otros formatos como VPC. Hay que recordar que para VMware, además hay que generar de forma manual el archivo **.VMX**.

## Auto guardado en Thunderbird

En muchas oportunidades nos encontramos trabajando en la redacción de un e-mail en Thunderbird y, por alguna razón inexplicable, se nos presenta la mala experiencia de cerrarse y todo lo redactado se borra de buenas a primeras. Por eso, aquí vamos a conocer la fórmula de **auto guardado**, que es tan simple como activar una función que, quién sabe por qué, está inactiva por defecto. Para activarla nos dirigimos a **Editar/Preferencias/Avanzadas** y ahí hacemos un clic en el botón **Editor de configuración**. En la zona superior de la nueva ventana escribimos **autosave**, modificamos el valor por **true** y listo.

## EVITAR LA VENTANA DE PROGRESO EN THUNDERBIRD

Para evitar que aparezca automáticamente la ventana de progreso cuando se envía un mensaje de correo electrónico en Thunderbird, debemos abrir el editor de configuración desde **Edición/Preferencias**, seleccionar **Opciones Avanzadas/Solapa General** y hacer clic en el **Editor de Configuración**. Allí hay que cambiar el valor de la llave **mailnews.show\_send\_progress** a **false**.

## Corrección automática de ortografía en Sylpheed

Para que al editar el correo se nos subrayen las palabras que no figuren en el diccionario que tengamos instalado, debemos seleccionar en la barra de menú la opción **Configuración** y luego **Preferencias comunes**.

En **Componer** elegimos la pestaña **Comprobación ortográfica** y allí seleccionamos **Activar comprobación ortográfica**. Y si hablamos castellano, en la casilla de **Idioma por omisión** debemos colocar **es**. Listo, la próxima vez que editemos los e-mails tendremos menos posibilidades de cometer errores ortográficos.

## Filtrado de correo no deseado con Sylpheed

Para filtrar el spam debemos tener un programa de reconocimiento del correo basura. El que viene preconfigurado (pero no preinstalado) con Sylpheed es **Bogofilter**, que es muy bueno y sencillo. En Debian y derivados como Ubuntu, la instalación es trivial:

```
# apt-get install bogofilter bogofilter
-bdb bogofilter-common
```

La configuración se realiza desde la barra de menú de Sylpheed seleccionando **Configuración y Preferencias comunes**. Dentro de la pestaña **Correo basura** habilitaremos la opción **Activar el control del correo basura**.

Ahora debemos seleccionar una carpeta donde colocar el spam (si no tenemos una la podemos crear ahí mismo). Es importante que coloquemos el spam en una carpeta y no lo borremos automáticamente porque aunque los falsos positivos son raros, pueden ocurrir y nadie quiere perder un e-mail por una confusión de Bogofilter.

Antes de que Bogofilter pueda empezar a funcionar hay que marcar una buena cantidad, que algunos recomiendan que no sean menos de 100 e-mails spam y otros tantos que no lo sean. Para marcar correo como spam o como no spam debemos usar las op-

ciones correspondientes en **Mensaje** en la barra de menú de Sylpheed.

Una vez que entrenamos a Bogofilter podemos disfrutar del filtrado automático del correo basura, aunque no está de más echarle una mirada rápida a la carpeta de spam cada tanto, especialmente al principio o si nuestros amigos suelen enviar e-mails que parecen spam.

## Visualizar e-mails HTML en Sylpheed sin usar Claws

Este script debe ejecutarse desde Sylpheed, para lo cual debe estar incluido en su menú **Acciones**. Para ello debemos ir al menú **Configuración** y seleccionar **Acciones**. En **Nombre de menú** debemos poner el nombre con el que se identificará el script (por ejemplo, **Ver mail como HTML**) y en **Línea de orden**

**BOGOFILTER v1.1.5** sourceforge.net

Bogofilter is a mail filter that classifies mail as spam or ham (non-spam) by a statistical analysis of the message's header and content (body). The program is able to learn from the user's classifications and corrections.

The statistical technique is known as the Bayesian technique and its use for spam was described by Paul Graham in his article [A Plan For Spam](#) in August 2002. Gary Robinson, in his weblog [Rants](#) (September 2002), suggested some refinements for improved discrimination between spam and ham. Bogofilter's primary algorithm uses the  $f(w)$  parameter and the Fisher inverse chi-square technique that he describes. Paul Graham's new article [Better Bayesian Filtering](#) (January 2003) suggests some useful parsing improvements.

Bogofilter is run by an MDA script to classify an incoming message as spam or ham (using wordlists stored by BerkeleyDB). Bogofilter provides processing for plain text and html. It supports multi-part mime message with decoding of base64, quoted-printable, and uuencoded text and ignores attachments, such as images.

Bogofilter is written in C. Supported platforms: Linux, FreeBSD, Solaris, OS X, HP-UX, AIX, RISC OS, SunOS, OS/2 ...

Download bogofilter	From <a href="#">Sourceforge Mirrors</a> . Some <a href="#">distributors</a> ship bogofilter.
Project Page	Please see our <a href="#">Sourceforge project page</a> .
Security Announcements	Check out the bogofilter <a href="#">security announcements</a> .

**Figura 11.** En <http://bogofilter.sourceforge.net> podemos encontrar la última versión de Bogofilter.

tenemos que escribir `~/sylpheed-2.0/sylpheed2html %f`. Finalmente, hacemos clic en **Añadir**. También se puede especificar el directorio de Sylpheed como segundo parámetro. Por ejemplo: `~/sylpheed-claws/sylpheed2html %f ~/sylpheed-claws`.

Para ejecutar el script hacemos clic en **Herramientas** opción **Acciones** y seleccionamos el script por el nombre ingresado en el paso anterior, por ejemplo: **Ver mail como HTML**.

Para un correcto funcionamiento se requiere que esté configurado el navegador web. Para ello debemos ir al menú **Configuración** y seleccionar **Preferencias comunes**, y elegir la pestaña **Otras**. En el campo **Navegador Web** completamos con el nombre de nuestro navegador favorito seguido de `%s`. Ejemplos:

```
galeon -n %s
firefox %s
opera -remote 'openURL(%s,new-page)'
mozilla %s
```

## Optimizar el funcionamiento de Firefox y OpenOffice

Firefox y OpenOffice son dos de las herramientas que más enorgullecen a toda la comunidad del código abierto.

Ahora bien, las dos son consideradas grandes **monstruos come-recursos**. Ambas piden bastante memoria RAM y bastante CPU. Pero no hay que preocuparse, porque hay algunas cosas que podemos hacer para que Firefox y OpenOffice funcionen un poquito más rápido. Comencemos por el navegador mimado del proyecto Mozilla. Normalmente, el navegador **Firefox** hará una petición HTTP por vez. Lo que vamos a hacer es habilitar el **pipelining**, que hará que el navegador funcione mucho más rápido porque realizará varias peticiones en forma simultánea. Para hacer esto, abrimos Firefox y en la barra URL tipeamos **about:config**. Una vez allí debemos buscar las siguientes secciones y definir los valores correspondientes, a saber:

```
network.http.pipelining = True
network.http.proxy.pipelining = True
network.http.pipelining.
maxrequests = 30
```

Con esto, el navegador hará aproximadamente treinta peticiones simultáneas. Por último, presionamos el botón derecho del mouse en un área vacía y seleccionamos **Nuevo/Enter**. Le ponemos el nombre **nlayout.initialpaint.delay** y de valor, cero. Esto define la cantidad de



Aunque es la suite más conocida, en muchos casos OpenOffice no es la mejor alternativa cuando hay que implementar una suite de oficina en un equipo que cuenta con pocos recursos. ¡Existen otras opciones! Entre ellas podemos mencionar **Koffice**, **Siag Office**, **GNUMeric** y **Abiword** como algunas de las que conviene tener en cuenta para este tipo de equipos.

segundos que transcurren entre que el navegador baja una página y la procesa. Con esto, notaremos que navegamos mucho más rápido por Internet. En cuanto a **OpenOffice**, todos notamos que, si bien el programa tarda bastante en arrancar, una vez que está en funcionamiento todo anda bastante bien. Por eso, con el siguiente consejo podremos acelerar bastante el inicio de OpenOffice, aun en sistemas que tienen poca memoria RAM. Dentro de OpenOffice, debemos ir a **Herramientas/Opciones**. Allí, en la sección **OpenOffice.org** vamos a la sección **Memoria de trabajo** y, en la sección **Antememoria de Imagen**, definimos el valor **Uso de OpenOffice.org** en 30 MB. Luego, el valor **Memoria por objeto** hay que definirlo en 2 MB. Si cerramos OpenOffice y lo volvemos a abrir, notaremos un importante crecimiento en la memoria RAM.

Tanto OpenOffice como Firefox (y Thunderbird) son herramientas que consumen muchos recursos, y no son recomendables para ser usadas en computadoras que cuenten con menos de 256 MB de memoria RAM.

## Algunos trucos para OpenOffice

He aquí un pequeño listado de trucos que hemos seleccionado especialmente:

- En **Writer**, para poder hacer un conteo de **caracteres totales** del documento, hay que ir a **Archivo/Propiedades/Estadísticas**.
- En **Writer**, podemos cambiar el color de **resaltado** si hacemos clic con el botón derecho del mouse en el botón de resaltado. Si lo mantenemos presionado se desplegará en pantalla la lista de colores.
- En **Calc**, si queremos que el texto se adapte al tamaño de una celda, debemos presionar con el botón derecho sobre la celda e ir a **Propiedades** y luego en la solapa **Alineación**, seleccionar la opción **Corte de línea automático**.
- Generalmente, las **fórmulas de Excel** no funcionan en **Calc** porque éste utiliza comillas para separar los argumentos en lugar de comas.
- Entre **Impress** y **PowerPoint** hay varias diferencias. Lo mejor es hacer un docu-

## REPOSITORIO DE EXTENSIONES FIREFOX

Existe un sitio que nuclea la comunidad de desarrolladores alrededor de Firefox llamado **Mozdev** ([www.mozdev.org](http://www.mozdev.org)). Allí no sólo encontraremos **parches** y **actualizaciones** para Firefox, sino un completo índice de **extensiones**.

## CAMBIAR EL IDIOMA POR DEFECTO DE LOS SITIOS EN FIREFOX

Muchos sitios web ofrecen contenido en diferentes idiomas según la configuración del navegador. Podemos definir múltiples idiomas y la prioridad de uso de cada uno en el menú **Herramientas/Opciones/Avanzado**, en la sección **General**.

mento en Impress, pasarlo a PowerPoint para ajustar las diferencias y volver a abrirlo en Impress hasta que se vea exactamente igual en los dos. Lo más fácil, es guardar la presentación en .PDF.

- En **Writer**, si queremos que nuestros documentos puedan ser accesibles desde casi cualquier procesador de textos, debemos utilizar el formato **RTF** para almacenar los archivos.
- OpenOffice no funciona con las **macros** de MS Office.
- Cuando trabajemos en cualquiera de las aplicaciones de OpenOffice, es conveniente usar fuentes que sepamos que están disponibles en MS Office, así todo el mundo podrá ver nuestras creaciones tal como las hicimos. Por ejemplo, podemos utilizar fuentes como Times new Roman, Arial y Courier, entre otras.

## Cómo hacer sesiones de VNC colaborativas

Para aquellos que no lo conocen, VNC es un sistema por el cual uno se puede

conectar a un **equipo remoto** y controlarlo de manera visual tal como si estuviera en la máquina real (**Figura 12**). Una sesión VNC colaborativa consiste en que muchas personas se pueden conectar al escritorio del mismo equipo y cada una tiene un puntero del mouse (es muy divertido). Ahora bien, el concepto es muy interesante pero al mismo tiempo puede ser un terrible caos (imagínense cinco punteros del mouse yendo de aquí para allá). Para solucionar eso, el sistema recomendado es **Collaborative VNC** ([www.benjie.org/software/linux/collaborative-vnc](http://www.benjie.org/software/linux/collaborative-vnc)), que ofrece un sistema muy versátil para **controlar quién está al mando**. Generalmente, el primero que se conecta a la sesión de VNC es el que tiene el control. Luego, si se conectan otras personas, hay varias formas de manejar por turnos quién tiene el control del escritorio. El método más simple es **delegar el derecho** a la persona que queremos habilitar para trabajar. Otra forma es un modo por el cual si alguien hace clic sobre el escritorio (y la persona que tiene el derecho de control está inac-

## III FUNCIONALIDADES DEL ICONO DE LA TRAYBAR DE AMAROK

El iconito de aMarok ofrece mucha información y varias funcionalidades mientras el reproductor está funcionando. Veamos algunas características:

- Si posicionamos el cursor del mouse sobre él, veremos un recuadro con toda la **información** relacionada con la canción que se está reproduciendo.
- Si presionamos el botón del medio del mouse, se **pausará** la canción actual.
- Si el mouse tiene una ruedita, usándola sobre el icono se cambia el **volumen** de la canción.
- Si presionamos la tecla **SHIFT** mientras se usa la ruedita del mouse, se cambia la ubicación de reproducción de la canción.
- Si presionamos **CTRL** mientras se usa la ruedita del mouse, se cambiará de canción.

tiva por más de quince segundos) se le otorga automáticamente el control a esa persona.

Como vemos, Collaborative VNC ofrece varias formas de hacer que el trabajo de varias personas sobre el mismo escritorio no sea un terrible caos. Es importante mencionar que ésta no es una aplicación per-se, sino que es un **patch** para el popular **TightVNC** (que podemos descargar de [www.tightvnc.com/download.html](http://www.tightvnc.com/download.html)). El patch en cuestión lo podemos descargar de [www.benjie.org/software/linux/collaborative-vnc/vnc-collaborate5.1.diff](http://www.benjie.org/software/linux/collaborative-vnc/vnc-collaborate5.1.diff) y para aplicarlo al código fuente de TightVNC debemos utilizar el siguiente comando:

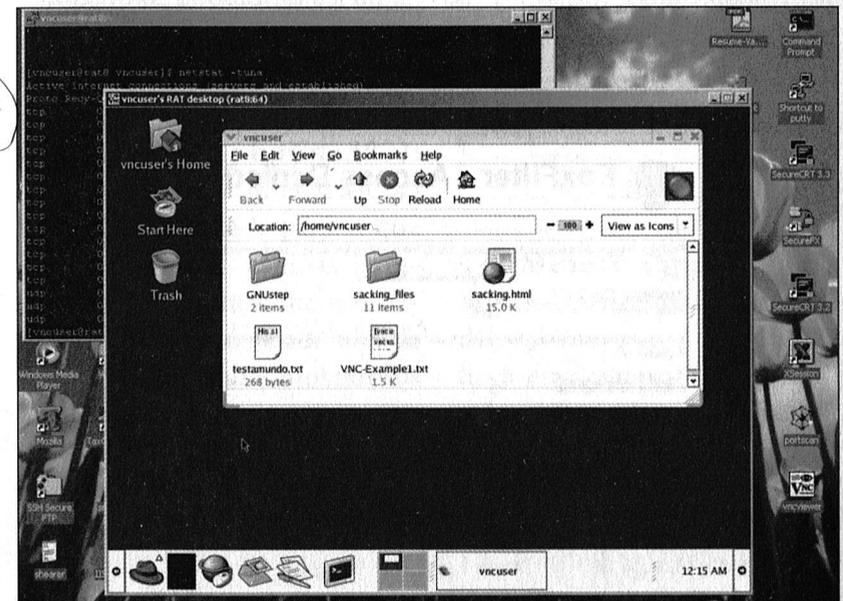
```
patch -p1 < ../vnc-collaborate5.1.diff
```

Este comando debe ejecutarse desde el directorio **vnc\_unixsrc** del paquete des-

comprimido de TightVNC. El directorio con el contenido del patch debe estar en la misma ubicación que el directorio **vnc\_unixsrc** (como podemos ver por la ruta utilizada en el comando patch). Una vez finalizado el proceso de modificación, se procede a compilar TightVNC normalmente.

Por último, para ejecutar el servidor VNC en modo colaborativo debemos tipear el comando **Xvnc -collaborate**. De todas maneras, vale la pena leer la documentación del patch para informarse sobre los diferentes parámetros que permiten modificar el modo de trabajo.

Una última recomendación: sería bueno que la sesión VNC colaborativa fuese ejecutada en una cuenta especial de usuario en el servidor, en donde tanto el escritorio como las aplicaciones estén especialmente dispuestas para el trabajo colaborativo. Por ejemplo, podemos ar-



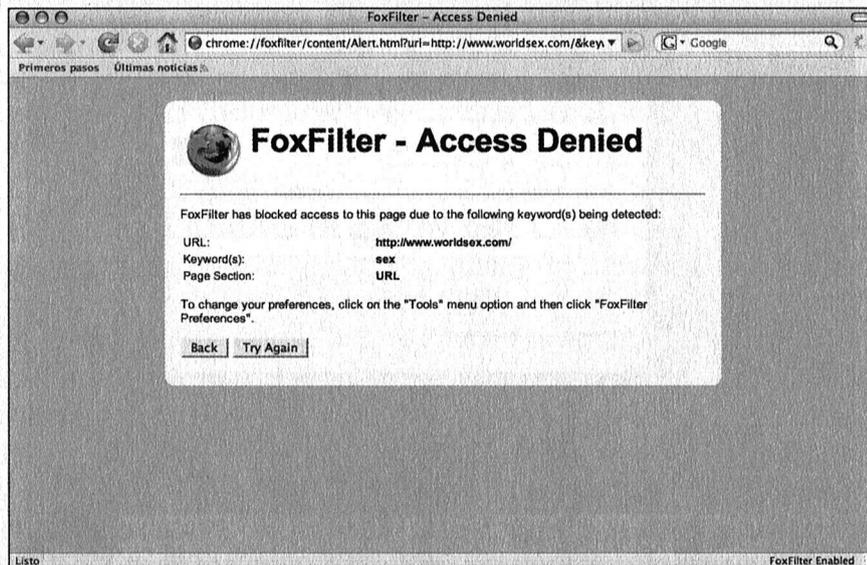
**Figura 12.** Con el sistema VNC podemos conectarnos a un equipo remoto y controlarlo como si estuviéramos allí.

mar un escritorio en donde los únicos iconos que haya correspondan a los documentos en los que se va a trabajar de manera colaborativa. Al mismo tiempo, nunca está de más poner algún tipo de aplicación de chat o un simple procesador de texto para que las personas conectadas puedan hablar entre ellas. De todas formas, esto puede complementarse perfectamente con una sesión de Skype para que, mientras trabajan, las personas puedan charlar sobre el trabajo, la vida, la familia, etcétera.

## Control de navegación en Firefox

A la hora de controlar a qué sitios web entran nuestros hijos o hermanos pequeños, las opciones son bastante limitadas en el mundo de Linux. Hay soluciones muy buenas pero propietarias (pagas), y las

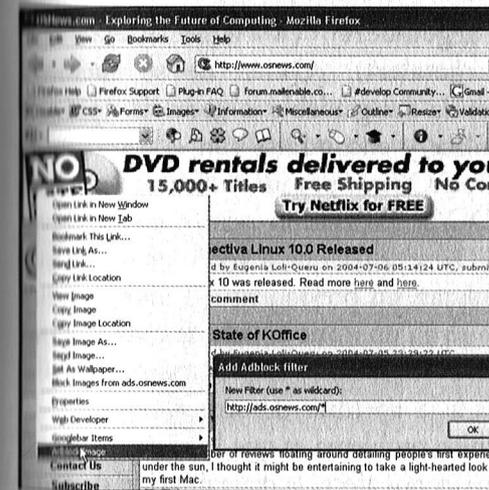
gratuitas son bastante difíciles de configurar. De todas formas, hay una opción intermedia que, si bien no es tan completa como **SurfSafe**, es gratuita y fácil de usar. Se trata de un plug-in para Firefox llamado **FoxFilter** (<http://addons.mozilla.org/firefox/4351>) que nos permite controlar qué sitios web serán accesibles (**Figura 13**). Una vez instalado, ingresamos al panel de configuración (**Herramientas/FoxFilter preferences**) e ingresamos la password por defecto que es **foxfilter** (debemos cambiarla inmediatamente, claro). Una vez dentro del panel de configuración, podemos definir un listado de palabras clave a buscar dentro de los sitios para detectar el contenido no deseado. También podemos definir en dónde buscará esas palabras (en la URL, en el contenido del sitio, en los meta tags, etcétera). Además, podemos definir algunos sitios como **excepción** (podemos poner aquí los buscadores como Google).



**Figura 13.** FoxFilter nos permitirá controlar cuáles sitios web son accesibles (y cuáles no) desde el popular navegador.

## Guía de extensiones Firefox

**Adblock** (<http://adblockplus.mozdev.org>): bloquea las publicidades de todo tipo. Basta con señalar una publicidad, apretar el botón secundario del mouse y definir que ésta no vuelva a aparecer. Permite ingresar palabras clave para excluir dominios enteros desde los cuales aparece publicidad.

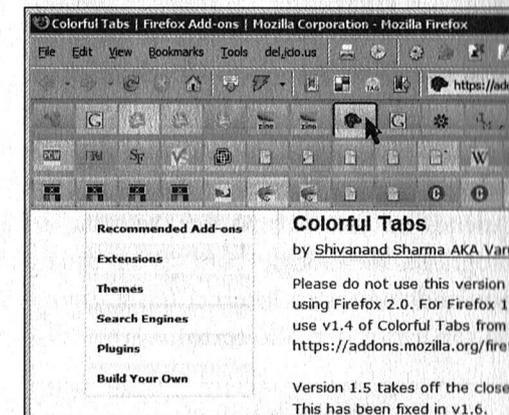


**Cache** (<http://addons.mozilla.org/extensions/moreinfo.php?id=301>): esta utilidad nos brinda un fácil acceso a la versión cacheada de la página que estamos mirando. Las páginas cambian el contenido, algunas incluso desaparecen. Esta extensión hace uso de la cache de Google para mostrarnos una versión anterior de la página en cuestión. Para eso, en el menú contextual nos habilita una opción llamada **Cache This Page**.

**Flashgot** ([www.flashgot.net](http://www.flashgot.net)): permite conectar Firefox con los gestores de descarga más populares, tanto para Li-

nux como para Windows. Se usa normalmente en las descargas grandes o múltiples en las que no queremos arriesgarnos a que un apagado involuntario del navegador nos haga perder lo que llevamos horas bajando. Incluso podemos enviar a nuestro gestor de descargas múltiples links de una sola vez.

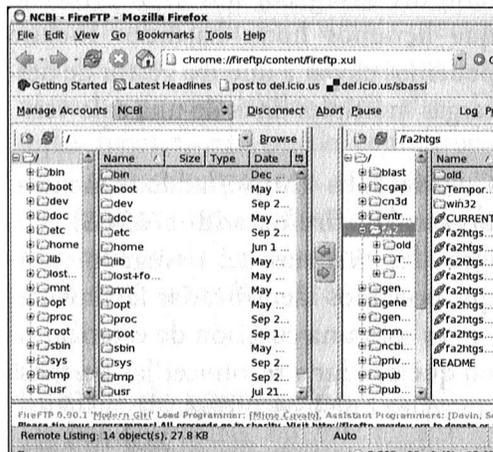
**ColorfulTabs** (<https://addons.mozilla.org/en-US/firefox/addon/1368>): colorea las pestañas del navegador para que podamos identificarlas fácilmente. No es sólo una cuestión de estética, sino que ayudar a reconocer las pestañas sin tener que leer el título.



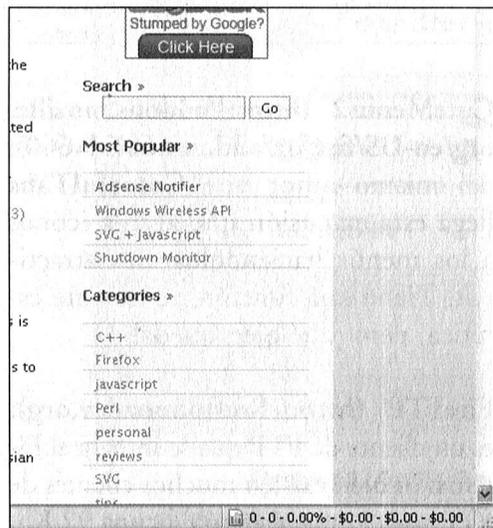
**CuteMenus2** (<https://addons.mozilla.org/en-US/firefox/addon/660?id=660>): del mismo autor que **ColorfulTabs** llega esta extensión que agrega iconos a los menús haciéndolos más atractivos. Tiene una función puramente estética, pero ¡qué bien queda!

**FireFTP** (<http://fireftp.mozdev.org>): es un cliente de FTP que se integra al Firefox. Si bien existen muchos clientes de FTP para Linux, éste sólo ocupa 92 Kb,

es bastante intuitivo y soporta continuar la transferencia de archivos descargados parcialmente. No funciona de manera individual, es decir, necesita Firefox.



**Adsense Notifier** (<http://code.mincus.com/?p=3>): muestra en la barra de estado de Firefox el monto que tenemos acumulado en nuestra cuenta de **Adsense**. Para quienes no son webmasters, se trata del programa de Google para ganar dinero poniendo avisos en Internet (en su mayoría son avisos con solo texto).



**Web Developer** (<http://chrispederick.com/work/webdeveloper>): extensión obligatoria para quienes hacen páginas web. El listado de utilitarios agrupados en esta extensión es muy grande para nombrar en este espacio. Permite editar estilos CSS de la página que estamos viendo (y aplicarlo en el momento), validar código, ver fuentes de imágenes, ver detalle de los formularios y un largo etcétera.

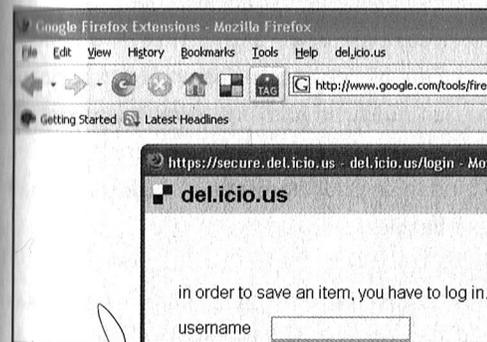
**TargetAlert** ([www.bolinfest.com/targetalert](http://www.bolinfest.com/targetalert)): ¿Quién no se encontró con la desagradable sorpresa de seguir un enlace y encontrarse con que se abre el OpenOffice porque había un archivo .DOC? Lo mismo ocurre con archivos .PDF y .ZIP, que a veces los webmaster enlazan sin previo aviso. TargetAlert nos muestra un icono al lado de cada link que lleva a algún otro tipo de archivo (o si lleva a la apertura de otro navegador). El único problema es que la aparición de estos iconos afecta la diagramación de aquellos sitios que tienen calculada la posición de sus elementos pixel por pixel. En la última versión, el icono se muestra sólo cuando pasamos la flecha por encima, aunque este comportamiento es regulable.

**Greasemonkey** (<http://greasemonkey.mozdev.org>): permite hacer modificaciones a nivel local por si no nos gusta el comportamiento de la página tal como fue programada por el diseñador del sitio. Incluso permite arreglar los bugs o agregar nuevas opciones. Mayor flexibilidad, imposible.

**Gmail File Space** ([www.rjonna.com/ext/gspace.php](http://www.rjonna.com/ext/gspace.php)): los usuarios de Linux

podrán dejar de envidiar el Gmail Drive de Windows, ya que **Gspace** permite usar los 2 GB de espacio de Gmail como si fuera un disco remoto.

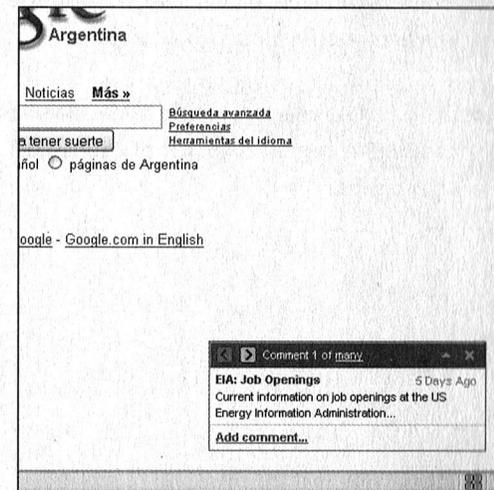
**Del.icio.us buttons** (<http://addons.mozilla.org/extensions/moreinfo.php?id=1532>): nos permite guardar en línea las direcciones de nuestros sitios favoritos y compartirlos con otros usuarios. No importa desde qué computadora nos conectemos, nuestras direcciones favoritas estarán siempre disponibles. Esta extensión agrega un menú en el navegador para publicar en del.icio.us de manera más cómoda.



**Tab Preview** (<http://ted.mielczarek.org/code/mozilla/tabpreview>): cuando tenemos muchas solapas abiertas, a veces se hace difícil recordar qué hay dentro de cada una de ellas. Para ayudarnos, **Tab Preview** nos muestra una pequeña imagen de la página sin tener que abrir la solapa. Es muy útil para ver si una página ya terminó de cargar. Permite configurar el tiempo de aparición y el tamaño de la imagen.

**Google Toolbar** (<http://toolbar.google.com/firefox>): a pesar de que Firefox tiene su ventana de Google incorporada, esta extensión no es redundante. Viene con corrector ortográfico en varios idiomas y para cualquier formulario que usemos, acceso para publicar directamente en Blogger, traductor, autocompletado de formularios y búsqueda en los distintos servicios de Google. Otra diferencia con respecto a la ventana Google de Firefox es que a medida que vamos escribiendo, nos va dando sugerencias de búsqueda.

**Blogger Web Comments** ([www.google.com/tools/firefox/webcomments](http://www.google.com/tools/firefox/webcomments)): nos muestra qué es lo que opina la gente con respecto a cualquier página que estamos visitando con un mensaje en la esquina inferior derecha. También permite dejar nuestra opinión sobre una página si es que tenemos una cuenta en Blogger ([www.blogger.com](http://www.blogger.com)).





- 1 Midnight Commander posee un menú de acciones rápidas al que se puede acceder con la tecla **F2**. Allí, por ejemplo, podemos elegir una opción para generar paquetes .tar.gz según una selección de archivos.
- 2 Muchas distribuciones ofrecen previsualización automática de archivos en Konqueror. Por ejemplo, con sólo posicionar el mouse sobre un icono de un archivo MP3 y esperar unos segundos, éste comenzará a reproducirse automáticamente.
- 3 Crear un backup de la base de datos de e-mails de Thunderbird es tan simple como resguardar el directorio **.thunderbird** ubicado en **\$HOME**.
- 4 Es posible escuchar archivos MP3 desde la consola con el comando **mpg123 [archivo.mp3]**. Esta herramienta viene incluida en prácticamente todos los repositorios de las distribuciones más populares.
- 5 Si buscamos una solución de e-mail para la consola, la combinación **pine + fetchmail** es excelente.
- 6 Prácticamente todos los comandos del sistema poseen un parámetro **-v (verbose)** que los vuelve más **comunicativos** con el usuario.
- 7 Muchos comandos del sistema poseen un parámetro **-H** que convierte su salida en formato **legible por humanos**.
- 8 NimbleX es una aplicación en línea que permite crear una distribución propia de GNU/Linux en **www.nimblex.net**.
- 9 Es posible descargar un sistema de búsqueda mejor para Linux llamado **Beagle**. Éste ofrece resultados en el acto y busca en nuestros archivos, e-mails, conversaciones, etcétera. Podemos instalarlo en Debian y sus derivados con **apt-get install beagle**.
- 10 Los usuarios de Debian que quieran instalar grupos de aplicaciones multimedia, de escritorio y juegos pueden utilizar **tasksel** para hacerlo de manera rápida y sencilla.

# Trucos para Vi y Emacs

Vi y Emacs son los dos editores más populares del mundo de UNIX y de Linux. Su flexibilidad y su cantidad de funcionalidades hacen de ellos los preferidos por los programadores, administradores y usuarios. En este capítulo veremos una selección de trucos para sacar el máximo provecho de estas herramientas y un tutorial para los usuarios que nunca las usaron.

Tutorial de Vi en 10 minutos	138
Cómo visualizar caracteres ocultos en Vi	143
Acceso rápido a la shell desde Vi	143
¿En qué línea estoy?	143
Insertar el texto de un archivo en la posición del cursor	144
Cómo insertar la salida de un comando como texto en Vi	144
No tengo permisos de root para guardar este archivo	145
Cómo realizar búsquedas sin respetar la capitalización de las letras	145
Guardar la configuración en un archivo de texto	145
Cómo usar ventanas en Vi	146
Cómo usar solapas en la versión 7 de Vim	146
Tutorial de Emacs en 10 minutos	147
Cómo abrir una terminal de comandos en Emacs	151
Usar emacs como gestor de archivos	152
Asignar teclas de función a operaciones típicas de edición	153
Hacer a emacs un poco menos irritable	154
Los juegos ocultos de Emacs	154
Cómo acceder a una sesión con un psicoterapeuta	155
Algo sobre nuestro futuro, todos los días	157

## Tutorial de Vi en 10 minutos

Este tutorial tiene como objetivo aprender a dominar las funciones básicas (y algunas avanzadas) del editor de texto Vi en la consola de Linux (y podríamos decir del resto de los sistemas operativos UNIX en general). Lo primero que haremos es asegurarnos de que Vi esté instalado con el siguiente comando:

### vi —version

Se devolverá algo como lo siguiente:

```
VIM - Vi IMproved 6.2 (2003 Jun
 1, compiled Mar  1 2006
 20:29:11)
Compiled by root@b21.apple.com
Normal version without GUI.
  Features included (+) or not
  (-):
-arabic +autocmd -balloon_eval
  browse +builtin_terms
  +byte_offset +cindent
-clientserver -clipboard
  +cmdline_compl +cmdline_hist
  +cmdline_info +comments
+cryptv +cscope +dialog_con +diff
  +digraphs -dnd -ebcdic
  emacs_tags +eval
+ex_extra +extra_search -farsi
  +file_in_path +find_in_path
  +folding -footer
+fork() -gettext -hangul_input
  +iconv +insert_expand
  +jumplist -keymap -langmap
+libcall +linebreak +lispindent
  +listcmds +localmap +menu
  +mksession
```

```
+modify_fname +mouse -mousethape
  -mouse_dec -mouse_gpm
  mouse_jsbterm
-mouse_netterm +mouse_xterm
  +multi_byte +multi_lang
  netbeans_intg -osfiletype
+path_extra -perl +postscript
  +printer -python +quickfix
  rightleft -ruby
+scrollbind -signs +smartindent
  sniff +statusline
  sun_workshop +syntax
+tag_binary +tag_old_static
  tag_any_white -tcl +terminfo
  +termresponse
+textobjects +title -toolbar
  +user_commands +vertsplitt
  +virtualedit +visual
+visualextra +viminfo +vreplace
  +wildignore +wildmenu +windows
  +writebackup
-X11 -xfontset -xim -xsmpt
  xterm_clipboard -xterm_save
system vimrc file: "$VIM/
  vimrc"
  user vimrc file: "$HOME/.
  vimrc"
  user exrc file: "$HOME/
  .exrc"
fall-back for $VIM: "/usr/
  share/vim"
Compilation: gcc -c -I. -Iproto
  DHAVE_CONFIG_H -arch i386
  -arch ppc -g -Os -pipe -no
  cpp-precomp -arch i386 -arch
  ppc -pipe
Linking:
gcc -arch i386 -arch ppc
  -o vim -lnurses
  liconv
```

Si no entendemos de qué se trata todo este texto, no tenemos que hacernos problema, ya que lo importante es que Vi se encuentra instalado. Todo el resto se trata de un listado de librerías y otros componentes de soporte que también están instalados.

Ahora vamos a crear un **nuevo archivo de texto** para editar con Vi. Para esto debemos ingresar el comando:

### vi archivodeprueba

Y aparecerá la ventana de edición que vemos en la **Figura 1**.

Es importante destacar que hay dos modos de operación de Vi. Uno de ellos es el **modo de comandos**, que nos permite tipear comandos como abrir archivo, guardar, cortar línea, etcétera. El otro modo es el modo de edición, en el que podemos modificar el archivo de texto a gusto. Para entrar al modo de

## GUÍAS IMPRESAS

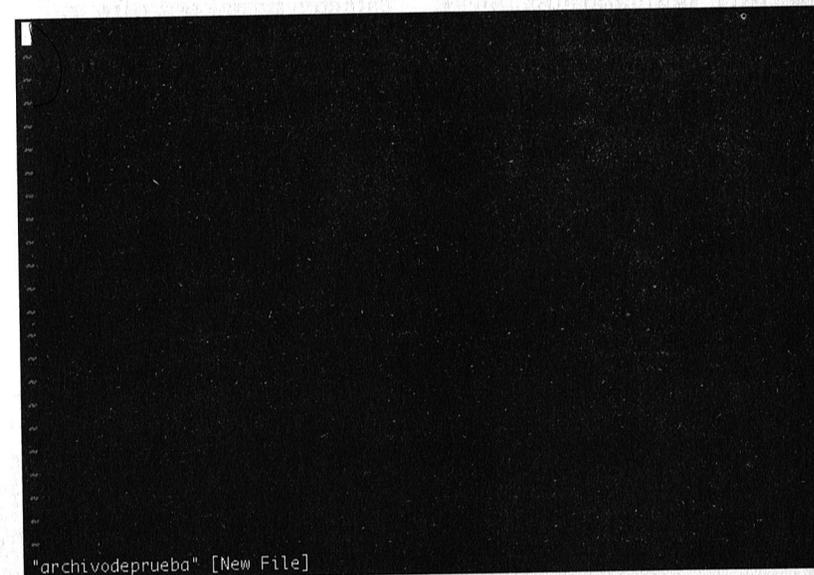
Vi es un editor que posee una gran cantidad de comandos y es posible que nos resulte difícil recordar siquiera un puñado de ellos. Por eso, nunca es mala idea armarnos una guía rápida, imprimirla y tenerla cerca del teclado cada vez que vayamos a utilizar este poderoso editor de texto.

comandos debemos presionar la tecla **ESC** seguida del símbolo **:** (dos puntos), y para entrar al modo de edición tenemos que presionar la tecla **ESC** seguida de alguna de las siguientes letras:

**a**: entra en modo de edición y agrega el texto tipeado justo detrás de la posición del cursor.

**i**: entra en modo de edición e inserta el texto justo delante de la posición actual del cursor.

**A**: añade el nuevo texto al final de la línea actual indicada por el cursor.



**Figura 1.** Un nuevo archivo creado con Vi.

**I:** inserta el texto al comienzo de la línea indicada por el cursor.

**O:** inserta una nueva línea entre la línea actual y la línea inferior a la posición del cursor.

Bien, ahora que sabemos cómo comenzar a editar en Vi, podemos escribir un bloque de código y utilizar las flechas de desplazamiento para acostumbrarnos a su funcionamiento. Una vez que hayamos terminado, podemos usar algunos de los siguientes comandos:

- Para guardar el archivo y continuar editándolo entramos al modo comando y tipeamos el comando **:w**.
- Para guardar el archivo y salir del editor entramos al modo comando y tipeamos el comando **:wq**.
- Para salir del editor sin guardar el archivo, entramos al modo comando y tipeamos el comando **:q!**.

El próximo paso es aprendernos los comandos para desplazarnos por el contenido de la ventana de Vi. Para todos los siguientes atajos es necesario presionar antes la tecla **ESC** para entrar en modo de comando:

**w:** mover el cursor hacia la siguiente palabra.

**e:** mover el cursor hacia el final de la palabra.

**b:** mover el cursor hacia el comienzo de la palabra.

**):** mover el cursor hacia el inicio de la próxima oración.

**(:** mover el cursor hacia el inicio de la oración actual.

**}:** mover el cursor hacia el inicio del próximo párrafo

**{:** mover el cursor hacia el inicio del párrafo actual.

**G:** mover el cursor hacia el final del archivo.

También disponemos de algunas combinaciones de teclas para desplazarnos por la ventana de edición:

**CTRL+F:** mueve una pantalla completa hacia adelante.

**CTRL+B:** mueve una pantalla completa hacia atrás.

Ahora vamos a aprender a modificar nuestro texto. El comando más simple es **r**, que se encarga de reemplazar el carácter actual por otro indicado justo a continuación del comando (como por ejemplo **rb**). Para borrar caracteres disponemos del comando **x**, que acepta prefijos numéricos para definir cuántos caracteres se deben borrar (por ejemplo **10x**). Por su parte, el comando **d** es un poco más versátil. Veamos algunas combinaciones:

## III PREFIJOS PARA LOS COMANDOS DE DESPLAZAMIENTO

Las combinaciones de las teclas **CTRL + F** y **CTRL + B** aceptan prefijos numéricos en el modo comando. Por ejemplo, la combinación **5 + CTRL + F** se encargará de avanzar cinco pantallas hacia adelante y **4 + CTRL + B** retrocederá cuatro pantallas.



Vi era el editor de texto más popular y utilizado hasta que aparecieron las primeras versiones de Emacs (desarrollado por Richard Stallman) a principios de la década de los 80. En la actualidad, la batalla todavía continúa y no está claro cuál de los dos editores cuenta con mayor cantidad de usuarios.

**dw:** borra desde la posición actual hasta el final de la palabra.

**10dw:** borra la palabra actual y las nueve próximas.

**D:** borra todo lo que hay hasta el final de la línea actual.

**3d\$:** borra las tres próximas líneas.

**d):** borra todo hasta el comienzo de la próxima línea.

**d):** borra todo hasta el comienzo del próximo párrafo.

**dd:** borra la línea actual.

Pero... ¿qué sucede si nos equivocamos y queremos deshacer los cambios? Bueno, Vi nos ofrece varias herramientas para llevar a cabo las tareas de restauración del texto contenido anteriormente en el buffer:

**u:** deshace el último cambio.

**U:** deshace todos los cambios realizados a una línea desde que el cursor se posicionó en ella.

**et:** deshace todos los cambios desde el momento en el que se abrió el archivo. Hay que tener mucho cuidado con esta opción porque no se puede deshacer.

Ahora veamos las operaciones básicas de **copiar y pegar** que tan útiles son a

la hora de trabajar con grandes cantidades de texto. El proceso de copiar y pegar en Vi implica tres acciones: copiar el contenido, ubicar el cursor en la nueva posición y, finalmente, pegar el contenido en esa ubicación. Para copiar utilizamos algunas de las siguientes combinaciones del comando **y**.

**yw:** copia una palabra.

**y\$:** copia todo hasta el final de la línea actual.

**y):** copia todo hasta el final de la oración actual.

**y):** copia todo hasta el final del párrafo actual.

**5Y:** copia cinco líneas, comenzando de la línea actual.

Una vez que se copió el texto deseado, movemos el cursor a la ubicación en donde queremos pegar el texto y presionamos el comando **p**.

Una característica sumamente interesante de Vi es que nos permite trabajar con múltiples buffers de texto (es algo

## III ¿VI O VIM?

Muchos usuarios se confunden al leer los nombres Vi y Vim. Lo cierto es que el editor original (proveniente del mundo UNIX) se llama Vi. Luego de muchos años, surgió una versión mejorada de Vi llamada Vim (por Vi "Improved") y esa es la versión que actualmente viene en la mayor parte de las distribuciones de Linux. En esencia, ambos son iguales, la única diferencia es que Vim incluye varias funcionalidades adicionales a las ya numerorísimas de Vi.

así como si tuviéramos un montón de portapapeles en donde pegar muchos pedacitos de texto diferente). Los buffers están definidos con las letras del alfabeto (a,b,c, etcétera) y para hacer referencia a que un comando de copiado y pegado trabaje con el texto de un buffer en particular, debemos usar el comando "[letra]. Veamos algunos ejemplos:

"a5Y: copia cinco líneas en el buffer a.

"b10dd: borra diez líneas de texto y las copia hacia el buffer b.

"bp: inserta todo el contenido del buffer b a continuación del cursor.

Poco a poco, vamos cubriendo todas las funcionalidades esenciales de un editor de texto. Ahora veamos cómo realizar funciones de búsqueda. Para **buscar una cadena** en todo el archivo utilizamos el comando / seguido de la cadena que queremos buscar (por ejemplo /archivo). Vi posicionará el cursor en el primer resul-

tado de la búsqueda. Para repetirla y continuar buscando en el archivo, utilizamos el comando //. Si estamos editando a mitad del archivo, notaremos que el comando / realiza búsquedas a partir de la posición actual del cursor. Podemos realizar búsquedas hacia atrás del cursor utilizando el comando ? (por ejemplo ?archivo).

Antes de terminar, vamos a mencionar un comando sumamente importante para poder continuar con nuestro aprendizaje de este editor de texto. El comando :help, que nos muestra la pantalla inicial de ayuda de Vi y nos indica qué parámetros podemos utilizar para obtener ayuda sobre algún aspecto en particular del editor (Figura 2).

Con esto llegamos al final de nuestro tutorial de diez minutos de Vi. Es importante que practiquemos el uso de este editor de texto con varios archivos temporales para acostumbrarnos a los co-

mandos y a los modos de trabajo. Vi es un editor de texto muy completo (y complejo), por eso seguramente la siguiente selección de trucos nos será de gran utilidad para descubrir todo el potencial que esta herramienta esconde.

## Cómo visualizar caracteres ocultos en Vi

Una necesidad típica de los programadores de scripts es la visualización, en los archivos de texto, de caracteres ocultos no imprimibles, como los retornos de carro. En Vi podemos visualizarlos con el siguiente comando:

```
set list
```

## Acceso rápido a la shell desde Vi

Vi ofrece un sistema de acceso rápido a la shell del sistema mientras estamos editando nuestro archivo de texto. Para acceder a la línea de comandos, debemos ingresar lo siguiente:

```
!sh
```

## ¿CUÁL ES EL SIGNIFICADO DEL UNIVERSO Y DE LA VIDA?

Es posible que Vi tenga la respuesta a esta pregunta. ¿Difícil de creer? Para comprobarlo, basta con tipear el comando :h 42 y ver lo que sucede. Además, se rumorea por ahí que en Vi también está guardada la información sobre dónde podemos encontrar el Santo Grial (debemos ingresar :h holy-grail). Por último, si estamos muy desesperados y tipeamos :help!, Vi nos dará un mensaje para calmarnos.

## MINÚSCULAS, MAYÚSCULAS

Es posible convertir una línea completa en letras minúsculas o mayúsculas con la combinación de teclas **ESC + guu** para convertir en minúsculas o **ESC + gUU** para convertir en mayúsculas.

En donde **sh** es el nombre del shell que se va a ejecutar (en el caso del ejemplo, bash). Una vez que terminamos de usar el shell, debemos utilizar el comando **exit** para volver a la pantalla de edición de Vi.

## ¿En qué línea estoy?

Una de las características de Vi es que por defecto, no muestra en pantalla el número de línea en la que está ubicado el cursor. Para eso disponemos de dos comandos muy útiles:

```
:=
```

Nos indica en qué número de línea está ubicado el cursor.

```
::=
```

```
help.txt* For Vim version 6.2. Last change: 2003 May 11

      VIM - main help file

Move around: Use the cursor keys, or "h" to go left,      h l
              "j" to go down, "k" to go up, "l" to go right.  j
Close this window: Use ":q<Enter>".
Get out of Vim: Use ":qa!<Enter>" (careful, all changes are lost!).

Jump to a subject: Position the cursor on a tag between |bars| and hit CTRL-].
With the mouse:   ":set mouse=a" to enable the mouse (in xterm or GUI).
                  Double-click the left mouse button on a tag between |bars|.
jump back:        Type CTRL-T or CTRL-O (repeat to go further back).

Get specific help: It is possible to go directly to whatever you want help
on, by giving an argument to the ":help" command |:help|.
It is possible to further specify the context:
                  *help-context*
                  WHAT          PREPEND  EXAMPLE
                  Normal mode commands (nothing) :help x

help.txt [help][R0]
df
archivodeprueba [+]
recording
```

Figura 2. El sistema de ayuda nos será de gran utilidad a la hora de aprender nuevas funcionalidades de Vi.

Nos indica la cantidad de líneas que posee el archivo de texto que estamos editando. Finalmente, si queremos ver una columna que nos indique el número de todas las líneas del archivo, debemos usar el siguiente comando:

```
:set number
```

## Insertar el texto de un archivo en la posición del cursor

Vi nos ofrece un sistema rápido para insertar todo el texto contenido en un archivo definido en la posición actual del cursor. Esta funcionalidad será muy bienvenida por los programadores que tienen que usar una misma porción de texto en todos sus archivos de código (por ejemplo, avisos de li-

## GRABACIÓN DE COMANDOS

Podemos grabar una serie de comandos para luego ejecutarlos todos juntos (algo así como la idea de grabar un procesador de texto). Para aprender esto, veamos un ejemplo práctico. En primer lugar, habilitamos el **modo grabación** con la combinación de las teclas **ESC + qq**. El mensaje **recording** aparecerá en pantalla. Ahora podemos tipear todos los comandos de Vi que queremos que queden grabados. Para finalizar el proceso de grabación, usamos la combinación de las teclas **ESC + q**. Ahora, cada vez que queramos ejecutar la secuencia de comandos, debemos usar la combinación de las teclas **ESC + @q**. Simple, ¿no?

encia o declaraciones de archivos de cabecera). Para hacer esto, tenemos que utilizar el siguiente comando:

```
:r [archivo]
```

En donde **[archivo]** es la ruta completa al nombre de archivo cuyo texto se quiere insertar en la posición actual del cursor.

## Cómo insertar la salida de un comando como texto en Vi

Muchas veces suele aparecer la necesidad de almacenar la salida de texto de un comando como un archivo que después se pueda editar. La forma típica de hacer esto consiste en ejecutar el comando con un símbolo de redireccionamiento de su flujo de salida hacia un archivo determinado (por ejemplo **ls > archivo.txt**) o editar ese archivo en Vi. Pero... las cosas se complican cuando queremos insertar ese archivo en una posición particular de otro archivo mientras estamos trabajando en Vi. No es para preocuparse, porque la solución está al alcance de la mano. Mientras estamos trabajando en cualquier archivo, podemos insertar la salida de cualquier comando en esa posición del cursor con el siguiente comando de Vi:

```
:.! [comando]
```

En donde **[comando]** puede ser cualquier comando que devuelva una cadena de texto, como por ejemplo:

```
:.! banner "Hola Mundo"
```



Existen muchas versiones derivadas del editor Vi original. ElVis, Vim y Vigor son sólo algunos de los nombres de estas modificaciones. En Wikipedia hay un artículo interesante que lista todas las versiones e informa cuál es la característica de cada una de ellas. La dirección del artículo es <http://en.wikipedia.org/wiki/Vi>, y se encuentra en idioma inglés.

## No tengo permisos de root para guardar este archivo

Muchas veces sucede que trabajando en nuestro propio equipo, abrimos un archivo que tiene permisos solamente para que **root** lo modifique desde una sesión de usuario común. Vi lo abrirá sin problemas, nosotros lo editaremos, pero cuando lo queramos guardar, el sistema nos informará que como no somos usuario **root** no podemos guardarlo. ¿Está todo perdido? ¡Claro que no! Podemos usar el siguiente comando:

```
:w ! su root -c 'cat > %'
```

El sistema pedirá la clave de **root** antes de proceder con la actualización del archivo.

## ALGUNAS HERRAMIENTAS VISUALES

Una de las características que usan los programadores es la herramienta de **coloreado de la sintaxis de programación**, que les permite identificar más rápidamente las funciones y las partes del código fuente. Si esta funcionalidad no se encuentra habilitada, podemos hacerlo utilizando el comando **:syntax on**. Otra herramienta visual muy útil es la **regla**, que nos permite visualizar la línea y columna en la que se encuentra ubicado el cursor. Para habilitarla debemos usar el comando **:set ruler**.

## Cómo realizar búsquedas sin respetar la capitalización de las letras

Seguramente habremos notado que los sistemas de búsqueda tradicionales de Vi sólo encuentran resultados si la cadena de búsqueda es exactamente igual a la cadena encontrada (y con esto queremos decir que no haya diferencias entre las letras en mayúscula y minúscula). Entonces, si buscamos la cadena **PERRO** y en el archivo hay una sola palabra **perro**, Vi no la encontrará nunca porque las mayúsculas y minúsculas no concuerdan. Para realizar búsquedas sin importar si la capitalización de las letras se corresponde entre la cadena a buscar y la cadena encontrada, usamos el siguiente comando:

```
:set ignorecase smartcase
```

## Guardar la configuración en un archivo de texto

Hay muchas opciones de configuración del funcionamiento de Vi. Podemos utilizar el comando **:set all** para visuali-

zar en pantalla todas las opciones posibles (son muchas, hay que tener paciencia). Una vez que sepamos cuáles son todas las opciones que nos sirven, podemos guardarlas en un archivo de texto que debe llamarse `.vimrc` en nuestro directorio personal. De esta forma, cada vez que abramos el editor Vi, todas las **opciones set** contenidas en ese archivo serán definidas de manera automática.

## Cómo usar ventanas en Vi

Además de todas las opciones que hemos visto en lo que va del capítulo, Vi nos sorprende con la posibilidad de dividir la pantalla en ventanas (**Figura 3**) y así poder editar varios archivos en forma simultánea. Para tipear comandos de control de ventanas debemos presionar la tecla **ESC** seguida de la combinación de las teclas **CTRL + W**.

El comando **n** se encarga de crear una nueva ventana.

El comando **h** mueve el foco a la ventana de la izquierda.

- El comando **j** mueve el foco a la ventana de abajo.
- El comando **k** mueve el foco a la ventana de arriba.
- El comando **l** mueve el foco a la ventana de la derecha.
- El comando **H** mueve la ventana hacia la izquierda.
- El comando **J** mueve la ventana hacia abajo.
- El comando **K** mueve la ventana hacia arriba.
- El comando **L** mueve la ventana hacia la derecha.
- El comando **w** mueve el foco a la próxima ventana.

## Cómo usar solapas en la versión 7 de Vim

La versión 7 de Vim incluye la posibilidad de usar solapas tanto en el modo gráfico como en el modo texto.

De esta forma se simplifica muchísimo la edición de múltiples archivos en forma simultánea. Es importante destacar

que esta característica solamente funciona en la versión 7 de **Vim** (**vi improved** o mejorado).

Para abrir una nueva solapa podemos usar el comando **:tabnew [archivo]**, que se encargará de crear una nueva solapa y ubicar el contenido del archivo definido en **[archivo]** en ella. También podemos abrir múltiples archivos en diferentes solapas cuando iniciamos Vim por primera vez. El comando **vim -p archivo1 archivo2 archivo3** abrirá una nueva sesión de Vim con **archivo1** en la primera solapa, **archivo2** en la segunda y **archivo3** en la tercera. Por defecto, Vim sólo soporta hasta 10 solapas en la ventana de edición, pero este valor lo podemos cambiar editando el archivo `.vimrc` de nuestro directorio personal y agregando la siguiente entrada:

```
set tabpagemax=[valor]
```

Donde **[valor]** es el número máximo de solapas que queremos que Vim soporte. Para movernos por la solapas podemos usar los comandos **:tabn** y **:tabp** (avanzar y retroceder solapa). Si estamos en el modo gráfico, con hacer clic con el mouse en la solapa deseada podremos acceder a su contenido de manera automática, tal como en las solapas de Firefox. Finalmente, el comando **:tabs** nos

## VIM EN MODO GRÁFICO

Si trabajamos mucho en modo gráfico, entonces quizá nos resulte de mayor utilidad la herramienta **gvim**, una versión visual de Vim que utilizan las librerías **Gtk**. La mayor parte de las distribuciones la incluyen en sus repositorios de paquetes.

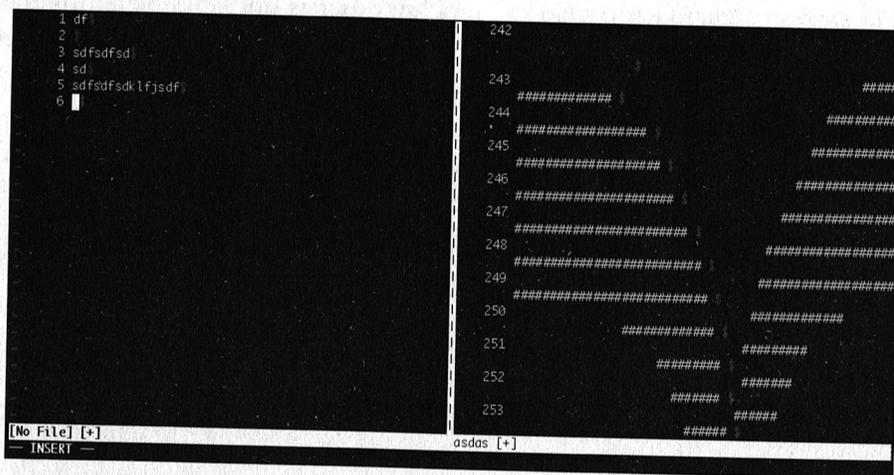
muestra en pantalla un listado de todas las solapas abiertas y el archivo que se está editando en cada una de ellas.

## Tutorial de Emacs en 10 minutos

Emacs es un editor de textos sorprendente no sólo porque incluye infinidad de funcionalidades para procesar archivos de texto sino porque además esconde un sinfín de aplicaciones de todo tipo: desde una simple calculadora hasta un psicólogo personal (si, leyeron bien). Creado originalmente por Richard Stallman (actual presidente de la Fundación del Software Libre y creador del sistema operativo GNU/Linux), Emacs se ha convertido en una de las aplicaciones pilares de UNIX y Linux. Tanto es así que una de las guerras típicas de nuestro ambiente es la de los usuarios de Vi contra

## BACKUP AUTOMÁTICO

Podemos indicarle a Vi que realice un backup (copia de seguridad) automático de cada archivo antes de guardarlo. Para esto, simplemente utilizamos el comando **:set writebackup**. El archivo de respaldo tendrá el mismo nombre que el archivo original pero con el agregado del símbolo `~`.



**Figura 3.** El editor Vi con la pantalla dividida en dos ventanas de edición.

los usuarios de Emacs. Lo cierto es que nunca se sabrá cuál es mejor que el otro, y ambos editores resultan demasiado complicados de aprender para los usuarios que recién se inician en el uso del sistema operativo UNIX/Linux. Por eso, antes de ver algunos trucos para el popular editor de texto de Richard Stallman, vamos a realizar (al igual que con Vi) un tutorial que en diez minutos nos permitirá tener una idea bastante clara de cómo se utiliza esta herramienta.

Comenzaremos por asegurarnos de que la aplicación esté instalada. Al igual que con Vi, ejecutamos el comando **emacs** con el parámetro **—version**:

**emacs —version**

```
GNU Emacs 21.2.1
Copyright (C) 2001 Free Software
Foundation, Inc.
GNU Emacs comes with ABSOLUTELY
```

```
NO WARRANTY.
```

```
You may redistribute copies of
Emacs
under the terms of the GNU
General Public License.
For more information about these
matters, see the file named
COPYING.
```

Si el sistema nos llegara a devolver un mensaje de error informándonos que el comando no existe, debemos recurrir a los CDs o DVDs de nuestra distribución para instalar los paquetes correspondientes. Cuando lo tenemos instalado, vamos a crear un nuevo archivo de texto:

**emacs archivonuevo**

Como podemos apreciar (**Figura 4**), la pantalla de la aplicación es ligeramente diferente a la de Vi. A diferencia de

éste, no hay que decirle a Emacs que queremos comenzar a escribir. Aquí lo podemos hacer directamente y todos los comandos los indicaremos por medio de las combinaciones de teclas (¡que son muchas!).

Vamos a escribir algunas líneas de texto para probar la respuesta de la aplicación a las teclas de desplazamiento. Luego vamos a salir de la aplicación con la combinación de las teclas **CTRL + X CTRL + C**. Cuidado que se trata de dos combinaciones de teclas que se realizan una después de la otra. Como se trata de un archivo nuevo, Emacs nos preguntará si queremos guardar los cambios, a lo que debemos responder **yes** o **no**. Volvemos al editor y ahora vamos a aprender algunas combinaciones de teclas para desplazarnos por la ventana de edición del texto:

- **CTRL + A**: mueve el cursor hacia el comienzo de la línea actual.
- **CTRL + E**: mueve el cursor hacia el final de la línea actual.
- **CTRL + N**: desplaza el cursor hacia la próxima línea.
- **CTRL + P**: desplaza el cursor hacia la línea anterior.
- **CTRL + V**: mueve el cursor una pantalla hacia delante.

### III INSERTAR UN ARCHIVO

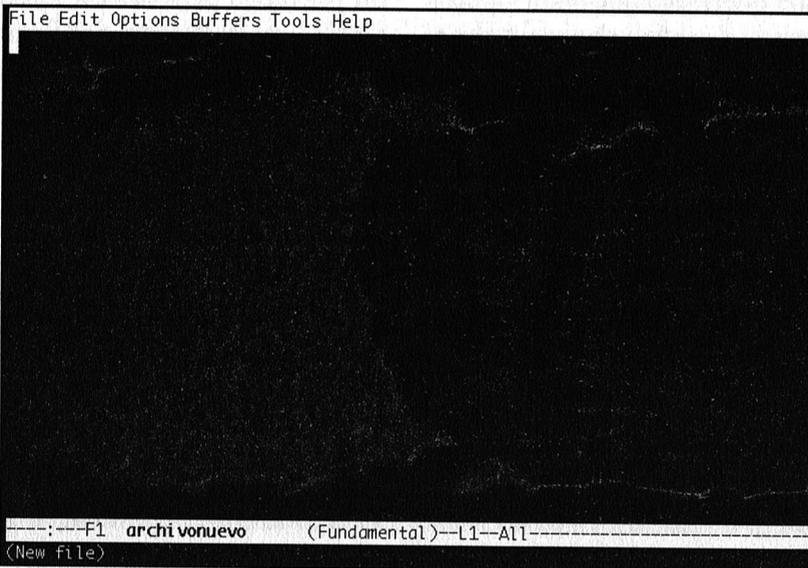
Si en algún momento queremos insertar el contenido de un archivo en un buffer de Emacs, simplemente debemos presionar la combinación de las teclas **CTRL + X I** y definir la ruta completa al archivo.

- **ESC + V**: mueve el cursor una pantalla hacia atrás.

Recordemos que también podemos utilizar las flechas de desplazamiento y las teclas de avance y retroceso de página. Ahora veamos algunas combinaciones de teclas que nos permiten borrar rápidamente el texto:

- **DEL**: borra la letra anterior.
- **CTRL + D**: borra la letra actual.
- **ESC + DEL**: borra la palabra anterior.
- **ESC + D**: borra la palabra actual.
- **CTRL + K**: borra todo lo que exista hasta el final de la línea.

Bien, ya sabemos cómo abrir archivos y cómo desplazarnos por ellos. Ahora veamos cómo llevar a cabo algunas funciones básicas de edición. Una de las características de Emacs es que nos permite **definir marcas** en diferentes ubicaciones del tex-



**Figura 4.** Pantalla inicial del editor de texto Emacs.

### III ¿DÓNDE ESTÁ EMACS?

Algunas distribuciones sólo traen el editor de texto Vi instalado por defecto, ya que es mucho más liviano que Emacs en términos de espacio de disco requerido para su utilización. Aun así, generalmente podemos encontrarlo en los repositorios de paquetes listos para ser instalados y usados, por lo que no hay problema si no funciona el comando **emacs**. Como otro recurso, podemos cargar la aplicación de actualización del sistema, y allí seguramente lo encontraremos.

to para llevar a cabo una serie de funciones de edición (como cortar y pegar). Para definir una marca, ubicamos el cursor en la posición deseada y usamos la combinación de las teclas **CTRL + @**. Si ya existía una marca, ésta será eliminada y reemplazada por la nueva. Ahora podemos mover el cursor hacia cualquier otra parte del documento de texto para definir lo que se conoce como **región**. La región es la zona comprendida entre la marca y la posición actual del cursor. Podemos movernos de un extremo a otro con la combinación de las teclas **CTRL + X CTRL + X**. Para cortar la región completa usamos la combinación de las teclas **CTRL + W** y para pegarla en otra ubicación del texto usamos la combinación de las teclas **CTRL + Y**. En realidad, lo que sucede cuando tipeamos la combinación de las teclas **CTRL + W** es que el texto se remueve del buffer de edición y se envía a un buffer especialmente dedicado al texto removido. Por lo tanto, podemos usar la combinación de teclas **CTRL + W** para borrar cualquier región sin obligación de tener que pegarla en otra ubicación del archivo.

Para continuar avanzando, la otra funcionalidad importante que debemos aprender de cualquier editor de texto es el sistema de **búsqueda**. En Emacs, buscar cadenas de texto es un proceso real-



El editor Emacs fue creado por Richard Stallman (actual presidente de la Fundación del Software Libre) y fue liberado en el año 1975. Ya en aquel entonces se utilizaba la

metodología de desarrollo colaborativo, algo que más tarde derivó en lo que en la actualidad se conoce como la metodología de desarrollo de código abierto.

mente simple que incluye la utilización de las siguientes combinaciones de teclas:

- **CTRL + S**: busca una cadena hacia adelante desde la posición actual del cursor.
- **CTRL + R**: busca hacia atrás.

Veamos una característica más de Emacs: la de dividir la pantalla en múltiples ventanas para editar el archivo en diferentes zonas. Podemos presionar en cualquier momento la combinación de las teclas **CTRL + X 2** para dividir la pantalla en 2 y visualizar diferentes zonas del archivo al mismo tiempo. Para mover el cursor de una ventana a la otra usamos la combinación de teclas **CTRL + X o**.

Es muy importante destacar que podemos realizar la división de la pantalla cuantas veces sea necesario, siempre con la misma combinación de teclas. Si queremos volver al punto inicial de

una sola ventana, tenemos que ubicarnos en aquella que queremos que se convierta en la vista principal y presionar la combinación de las teclas **CTRL + X 1**. Seguramente, la pregunta que surge es por la posibilidad de editar dos archivos diferentes en dos ventanas diferentes. Esto es muy simple: lo único que hay que hacer es ubicar el cursor en la ventana que deseamos que contenga otro archivo de texto y presionar la combinación de las teclas **CTRL + X CTRL + F** para abrir un archivo nuevo.

Y así hemos llegado al final de nuestro tutorial en diez minutos de Emacs. ¡Ya sabemos todo lo que necesitamos para dar nuestros primeros pasos en Emacs! Una cosa más: es importante que nos acostumbremos a usar el sistema de **ayuda** contenido en la herramienta para poder ir descubriendo nuevas posibilidades. Hay mucha documentación contenida en Emacs y hay varias formas de acceder a ella:

- **CTRL + H t**: abre en pantalla un tutorial de primeros pasos sobre Emacs.
- **CTRL + H I**: abre el documento de hipertexto que permite navegar por todo el sistema de ayuda de Emacs.
- **CTRL + H a**: Emacs solicita una cadena de búsqueda y nos muestra todos los comandos y las descripciones que contengan esa palabra clave.

Habiendo hecho la descripción correspondiente, ahora vamos a conocer los mejores trucos para sacarle el máximo provecho a este popular editor de texto.

## Cómo abrir una terminal de comandos en Emacs

Dentro de una sesión de Emacs, podemos abrir una terminal de comandos del sistema para ejecutar cualquier herramienta de la consola (**Figura 5**). Para esto, simplemente debemos presionar la

```
File Edit Options Buffers Tools In/Out Signals Help
Macbook-de-Facu:~ hfarena$ whoami
hfarena
Macbook-de-Facu:~ hfarena$ pwd
/Users/hfarena
Macbook-de-Facu:~ hfarena$ who
hfarena console May 4 10:22
hfarena tty1 May 4 14:06
Macbook-de-Facu:~ hfarena$ exit
exit

Process shell finished

---:**-F1 *shell* (Shell:no process)--L12--All
```

Figura 5. Podemos ejecutar un shell dentro de la ventana de edición de Emacs.

## PLANTILLAS DE COMANDOS

Al igual que Vi, Emacs es un editor repleto de funcionalidades que se traducen en infinidad de comandos y combinaciones de teclas. Como es muy difícil recordar incluso las más frecuentes, siempre es recomendable armarnos una plantilla de comandos, imprimirla y tenerla pegada en algún lugar visible para poder refrescar nuestra memoria rápidamente.

combinación de las teclas **ESC + X** y luego teclear **shell**. Una terminal aparecerá justo dentro de la ventana de edición. Todo el texto será impreso en el buffer de edición. Una vez que terminamos de tipear los comandos, podemos ingresar **exit** para volver al modo de funcionamiento tradicional de Emacs.

## Usar emacs como gestor de archivos

Una de las aplicaciones más interesantes de Emacs es la de funcionar como gestor de archivos. Para esto, debemos tipear el comando **emacs** seguido de la ruta a un directorio del sistema de archivos. Por ejemplo, el comando **emacs /Users** devolverá una pantalla como la de la **Figura 6**. Podemos utilizar las teclas de desplazamiento para movernos por los directorios del sistema de archivos. Si presionamos la tecla **ENTER** podemos ingresar

```
File Edit Options Buffers Tools Operate Mark Regexp Immediate Subdir Help
/Users:
used 16 available 30076520
drwxrwxr-t 6 root admin 204 Oct 24 2006
drwxrwxr-t 29 root admin 1038 May 4 10:21
-rw-rw-r-- 1 hfarena admin 6148 Oct 26 2006 .DS_Store
-rw-r--r-- 1 root wheel 0 Jan 13 2006 .localized
drwxrwxrwt 6 root wheel 204 Oct 26 2006 Shared
drwxr-xr-x 60 hfarena hfarena 2040 May 4 11:41 hfarena
```

Figura 6. Es posible navegar los directorios del sistema desde Emacs.

## MINIBUFFER

El minibuffer es la línea inferior en donde tecleamos comandos y rutas de archivos en Emacs. Allí, si presionamos la barra espaciadora, podremos acudir al **autocompletado**, que es un sistema que ofrece Emacs para completar automáticamente nombres de funciones y nombres de archivos.

a un directorio. Podemos borrar directorios y archivos marcándolos con la tecla **D** y presionando luego la tecla **X** para borrarlos todos juntos. Éstas son otras funcionalidades posibles:

- La tecla **V** muestra el contenido de un archivo.
- Con la tecla **R** se pueden renombrar archivos.
- Con la tecla **E** se edita un archivo.
- Con la tecla **C** se copia un archivo.
- Con la tecla **U** se deshace el último proceso de borrado.

## BACKUP AUTOMÁTICO

Podemos editar el archivo **.emacs** para agregar algunas funcionalidades respecto al sistema de resguardo automático:

```
(defconst use-backup-dir t)
; Directorio de backups
(setq backup-directory-alist (quote (("." .
"/backups/"))))
; Usar números de versión para los backups
version-control t
; Cantidad de nuevas versiones que se almacenaran (16)
kept-new-versions 16
; Cantidad de viejas versiones que se guardaran (2)
kept-old-versions 2
; Solicitar confirmación para borrar versiones de backup
delete-old-versions t
; Copiar archivos enlazados
backup-by-copying-when-linked t)
```

## Asignar teclas de función a operaciones típicas de edición

Si nos molesta estar tecleando muchas combinaciones de teclas para cortar, copiar, pegar y cualquier otra operación que llevamos a cabo con cierta frecuencia, este truco es ideal. Emacs nos permite **asignar acciones** a ciertas teclas (como las teclas de función **F1**, **F2**, etcétera). Lo único que debemos hacer es editar el archivo **.emacs** ubicado en nuestro directorio personal y agregar las siguientes líneas:

```
(global-set-key [f5] 'copy-region-as-kill) ; Copiar
```

```
(global-set-key [f6] 'kill-region) ; Cortar
(global-set-key [f7] 'yank) ; Pegar
```

Con esta configuración les estamos asignando las acciones de **Copiar**, **Cortar** y **Pegar** a las teclas **F5**, **F6** y **F7**. Podemos utilizar el mismo sistema para asignarles otras funciones a otras teclas. Otras opciones son:

```
(global-set-key [kp-subtract] 'undo) ;
```

Para asignarle a la tecla - del keypad la acción deshacer.

```
(global-set-key "\C-l" 'goto-line) ;
```

Para asignarle a la combinación de las teclas **CTRL + L** la función ir a línea.

```
(global-set-key [f2] 'split-window-vertically)
```

```
(global-set-key [f1] ')
```

Para asignarle a la tecla **F2** la función de dividir una ventana de forma vertical y a la tecla **F1** la función de eliminar esa división.



En sus más de treinta años de vida, GNU Emacs ha mutado en un abanico realmente muy amplio de versiones derivadas. Por ejemplo, hay versiones reducidas en funcionalidades (para sistemas con pocos recursos), versiones para otros sistemas operativos (como DOS) y mucho más. Podemos encontrar un listado completo en <http://en.wikipedia.org/wiki/Emacs>.

## Hacer a emacs un poco menos irritable

Hay ciertas cosas de Emacs que luego de utilizarlo por un tiempo se vuelven irritables, aunque por suerte todas se pueden solucionar muy fácilmente editando el archivo de configuración del perfil (`.emacs`, ubicado en el directorio personal de nuestro usuario). Veamos algunas de ellas:

```
(setq inhibit-startup-message t)
```

Deshabilita el mensaje de bienvenida.

```
(fset 'yes-or-no-p 'y-or-n-p)
```

¿Cansados de tener que teclear **yes** para confirmar las cosas? Con esta línea ahora sólo habrá que teclear **y**.

```
(setq next-line-add-newlines nil)
```

Se encarga de deshabilitar la creación automática de nuevas líneas cuando presionamos la tecla **FLECHA ABAJO** en la última línea del buffer.

## Los juegos ocultos de Emacs

La distribución oficial de Emacs incluye un paquete de aplicaciones y juegos desarrollados en el lenguaje **LISP**. Acceder a ellos es muy simple: lo primero que debemos hacer es saber qué juegos tenemos disponibles. Para eso tipeamos el siguiente comando:

```
ls /usr/share/emacs/21.2/lisp/play
```

## FAVORITOS EN BUFFERS

Podemos crear nuestro propio sistema de favoritos de buffers. Si trabajamos con múltiples archivos, podemos registrarlos todos y acceder a ellos rápidamente. Les asignaremos las funcionalidades de **Agregar a favoritos** y **Listar favoritos** a las teclas **F11** y **F12** en el archivo `.emacs` de nuestro directorio personal:

```
(local-set-key [f11] 'point-stack-push)
(local-set-key [f12] 'point-stack-pop)

(defun point-stack nil)

(defun point-stack-push ()
  "Push current location and buffer info onto stack."
  (interactive)
  (message "Location marked.")
  (setq point-stack (cons (list (current-buffer) (point)) point-stack)))

(defun point-stack-pop ()
  "Pop a location off the stack and move to buffer"
  (interactive)
  (if (null point-stack)
      (message "Stack is empty.")
      (switch-to-buffer (caar point-stack))
      (goto-char (cadr point-stack))
      (setq point-stack (cdr point-stack))))
```

Éste devolverá un listado de archivos como el siguiente:

```
5x5.el          dissociate.el
gametree.el     meese.el
```

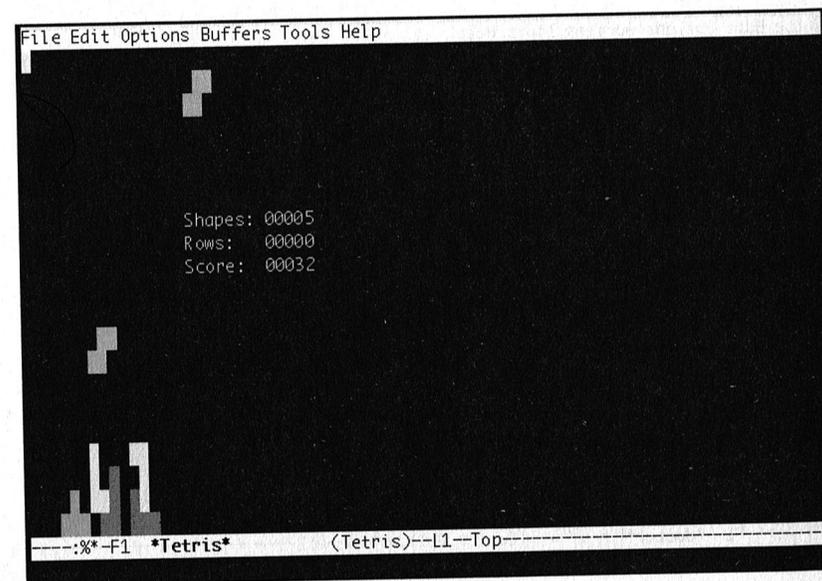
```
solitaire.elc
5x5.elc        dissociate.elc
gomoku.el      meese.el
spook.el
animate.el     doctor.el
gomoku.elc     morse.el
spook.elc
animate.elc    doctor.elc
handwrite.el  morse.elc
studly.el
blackbox.el    dunnet.el
handwrite.elc mpuz.el
studly.elc
blackbox.elc   dunnet.elc
hanoi.el       mpuz.elc
tetris.el
bruce.el       fortune.el
hanoi.elc      pong.el
tetris.elc
cookie1.el     fortune.elc
landmark.el    pong.elc
yow.el
cookie1.elc    gamegrid.el
```

```
landmark.elc  snake.el
yow.el
decipher.el   gamegrid.elc
life.el       snake.elc
zone.el
decipher.elc  gametree.el
life.elc      solitaire.elc
zone.elc
```

Vamos a jugar, por ejemplo, al **Tetris**. Cargamos Emacs y presionamos la combinación de las teclas **ESC + X**. Ahora tecleamos el nombre del juego sin la extensión (en el caso del ejemplo, **tetris**). Aparecerá el juego listo para ser disfrutado, como vemos en la **Figura 7**.

## Cómo acceder a una sesión con un psicoterapeuta

Luego de varias horas de trabajar con Emacs, de leer sus tutoriales y de utili-



**Figura 7.** Podemos jugar al Tetris incluido en el paquete de Emacs.

ar sus interminables combinaciones de teclas para acceder a los comandos, probablemente necesitemos una sesión con un psicoterapeuta que nos vuelva a poner los pies sobre la tierra. Los desarrolladores de Emacs pensaron en esto y decidieron incluir uno virtual. De hecho, el psicoterapeuta de Emacs es uno de los agregados más populares de este editor porque, más allá de toda broma, es un juego muy divertido que ayuda a relajarse. Para acceder a este ameno juego, sólo hay que presionar la combinación de las teclas **ESC + X**. Luego debemos teclear el nombre del programa que vamos a ejecutar: **doctor**. La principal desventaja es que este programa sólo responde a mensajes ingresados en idioma inglés, por lo que aquellos que no dominen aunque sea un poco esta lengua, no podrán hacer uso de sus bondades. Veamos una conversación de ejemplo:

Usuario: Good afternoon, Dr. (Buenas tardes, Dr.).

Psicoterapeuta: How are you? What brings you here? (¿Cómo estás? ¿Qué te trae a verme?).

Usuario: Well, I'm not ok... (Bueno, no estoy bien...).

Psicoterapeuta: Why do you say so? (¿Por qué decís eso?).

Usuario: Because my girlfriend left me. (Porque mi novia me dejó).

Psicoterapeuta: Is it because your girlfriend left you that you came to see me? (¿Y vienes a verme porque tu novia te dejó?)

Usuario: Yes. (Sí).

Psicoterapeuta: I see... Well, what

```
File Edit Options Buffers Tools Help
I am the psychotherapist. Please, describe your problems. Each time
you are finished talking, type RET twice.

Hello

How do you do? What brings you to see me?

Well, I'm not well...

Why do you say you are not well?

because my girlfriend left me

Is it because your girlfriend left you that you came to me?

yes

I see... Well, what makes you believe this is so?

:~**F1 *doctor* (Doctor Fill)--L20--All
```

Figura 8. El psicoterapeuta de Emacs es un programa muy divertido incluido en el editor.

makes you think this is so? (Bueno... ¿qué te hace pensar que esto es así?)

Y la conversación puede durar horas y horas. Sólo hay que probarlo para verlo divertido que resulta este programa incorporado en la distribución estándar de Emacs.

## Algo sobre nuestro futuro, todos los días

¿No sería lindo que cada vez que ingresamos a Emacs el programa nos devuelva automáticamente algo sobre nuestro futuro para ese día? ¿Es muy

fácil! Sólo debemos editar el archivo **.emacs** de nuestro directorio personal y agregar las siguientes líneas:

```
(add-hook 'after-init-hook
  (function
    (lambda ()
      (pop-to-buffer (get
        buffer-create "
          *Fortune*"))
      (shell-command "fortune
        -a" t))))
```

Y listo, cada vez que se inicie Emacs, se cargará el programa **fortune** y se imprimirá el mensaje en pantalla.

Para Vim

Para Emacs

- 1 Con **:set autoindent** se habilita la indentación automática.
- 2 Con **:set autowrite** se habilita la grabación automática.
- 3 Con **:set mesg** se permite que se reciban mensajes de otros usuarios.
- 4 Con **:set warn** se habilitan los mensajes de advertencia en pantalla.
- 5 Con **:set showmode** se muestra el modo de edición en el que estamos.

- 1 **CTRL + X CTRL + F** busca un archivo.
- 2 **CTRL + Z** suspende a Emacs.
- 3 **CTRL + G** detiene la ejecución del comando actual.
- 4 **CTRL + X CTRL + L** convierte una región en letras minúsculas.
- 5 **CTRL + X CTRL + U** convierte una región en letras mayúsculas.

# Optimizar el rendimiento y la seguridad

Todas las distribuciones de Linux están pensadas para ser compatibles con la mayor cantidad de equipos posible, y por eso no funcionan de forma óptima para nuestro equipo en particular. En este capítulo, veremos algunos consejos para adaptar el sistema a nuestro equipo y a nuestras necesidades. Además, dedicaremos una sección a analizar formas de aumentar la seguridad.

Cómo medir el rendimiento de nuestro equipo	160
Consejos para mejorar la experiencia de la usabilidad	163
Cómo optimizar el proceso de inicio de Linux	165
Cómo acelerar GNU/Linux en sistemas con pocos recursos	170
Consejos de uso del usuario root	174
Cómo optimizar el desempeño de nuestros discos IDE con hdparm	175
Cómo instalar programas desde el código fuente (.tar.gz)	176
Cómo saber si nuestro sistema está seguro	177
Trucos para sacarle el máximo provecho a Nmap	179
Tutorial en 10 minutos de iptables	180
Cómo montar una DMZ en pocos pasos	183
Beagle: un buscador rápido e inteligente	186

## Cómo medir el rendimiento de nuestro equipo

En general, el único que sabe exactamente cómo funciona el sistema es el propio usuario. ¿Quién no se da cuenta cuando la computadora se vuelve una carreta cuando intentamos abrir una imagen? Para evitar esto, lo mejor es conocer en qué situaciones ocurre. De más está decir que si el sistema siempre está lento (o sea, desde que inicia hasta que lo apagamos), entonces lo mejor será cambiar de distribución o hacer un arduo trabajo de afinación. Pero cuando el sistema se vuelve lento sólo en ocasiones, lo mejor es utilizar alguna herramienta para saber cuándo ocurre. Para esto disponemos del típico comando **top** que nos informa en pantalla, entre otras cosas, cómo está la carga del CPU, de la memoria RAM, el espacio libre y más. Si queremos algo más visual, **Ktop** (<http://linux.maruhn.com/sec/ktop.html>) posiblemente sea la herra-

mienta adecuada. Nos informa en tiempo real y con gráficos el estado del rendimiento de nuestro sistema. Los usuarios de GNOME encontrarán entre los apliques para la barra de tareas herramientas que hacen cosas similares. Las mejores distribuciones de GNU/Linux para jugar al tuning. Dentro del mundo de GNU/Linux hay muchas distribuciones, y algunas nos permiten afinar el sistema de forma más fácil que otras. Debemos decidir qué distribución nos conviene más: una fácil de instalar y configurar con todos sus paquetes compilados para que funcionen en la mayoría de las PCs del mercado o una distribución que si bien es difícil de instalar y configurar nos da la libertad de modificar el sistema completo y compilar cada uno de sus paquetes. Distribuciones como las del último tipo son Debian ([www.debian.org](http://www.debian.org)), Gentoo ([www.gentoo.org](http://www.gentoo.org)) y Slackware ([www.slackware.org](http://www.slackware.org)), entre otras. Estas distribuciones son de las más

documentadas y modificadas a lo largo de la historia, y si bien Gentoo Linux es relativamente nuevo en comparación con los otros dos, es espectacular al momento de querer optimizar nuestro GNU/Linux ya que podemos compilar absolutamente todos los paquetes que queramos con las opciones y flags que deseemos (mientras nuestro hard y el programa lo soporte, si no nunca compilad). A diferencia de las otras dos distribuciones, podemos hacerlo de forma práctica y sencilla gracias a un sistema llamado Portage, que automáticamente descarga de Internet el código fuente del paquete que queremos compilar, busca sus dependencias (librerías u otros programas que necesite ese paquete para funcionar o para poder compilarlo), descarga los fuentes y los compila. De esta forma, podremos tener todos los paquetes compilados a nuestro gusto y optimizados para nuestro tipo de procesador. En la página de Gentoo encontraremos documentación en varios idiomas (incluido el español) para aprender a

## III EL SISTEMA VNC

Algunas distribuciones incluyen preconfigurado un servidor de acceso remoto en modo gráfico. Este es considerado un protocolo inseguro ya que al igual que **telnet**, tampoco gestiona los datos con encriptación. Por tal razón, si no lo usamos, lo más conveniente es desactivarlo.

usarlo, además de los foros en donde podremos plantear nuestras preguntas o encontrar nuevas formas de optimizar nuestro sistema. Pero, como mencionamos antes, no todo es color de rosa. Lo que ganamos en velocidad de procesamiento en los programas al tenerlos compilados para nuestra PC en realidad lo perdemos compilando cada uno de estos programas (para tener un sistema básico con KDE se tarda 20 horas aproximadamente entre la creación del sistema de archivos y el último paquete compilado en un equipo actual).

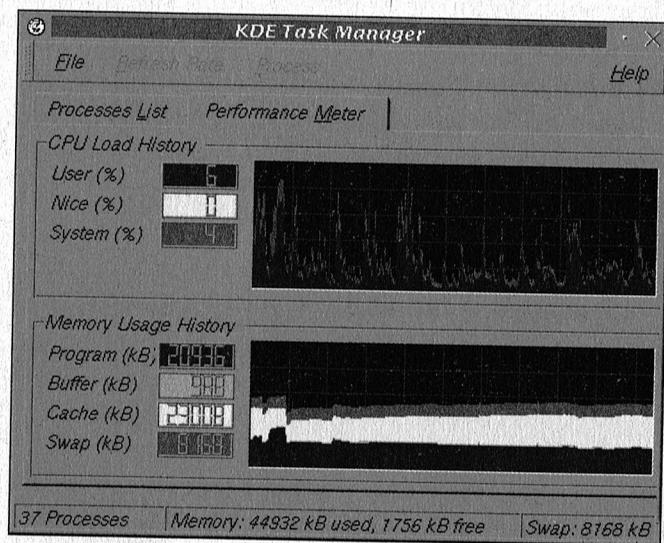


Figura 1. El monitor de actividad Ktop en acción.

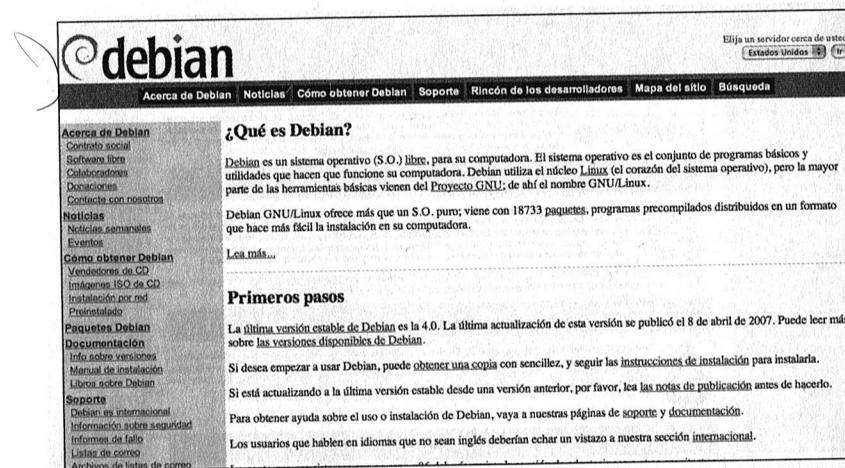


Figura 2. En el sitio oficial de Debian ([www.debian.org](http://www.debian.org)) hay muchas variantes de esta distribución para diferentes plataformas y necesidades.

La otra distribución es Debian que, a diferencia de Gentoo, ya viene con un montón de paquetes compilados. Debian incluye un sistema de instalación en modo texto no demasiado sofisticado, pero suficiente. En cuanto a la configuración, prácticamente no incluye herramientas de configuración visuales ni autodetección de hardware, por lo que también nos obliga a configurar todo a mano con la ayuda de un simple editor de texto. Una de las cosas realmente interesantes de Debian es el sistema de actualización de paquetes **APT** que permite instalar y gestionar aplicaciones instaladas de forma sencilla. Por último, casi todos los paquetes de Debian están optimizados para funcionar en su máximo rendimiento. Slackware, por su parte, es considerada la distribución de GNU/Linux más parecida al UNIX original. Como Debian y Gentoo, no incluye muchas herramientas de configuración automática y nos obliga a configurar todo a mano. El punto flojo de Slackware es que



En realidad es posible modificar cualquier distribución para adaptarla a cualquier equipo. Sin embargo, lo cierto es que este trabajo a veces puede ser demasiado arduo (sobre todo cuando hay que hacer muchas modificaciones de configuración). Por eso, es recomendable buscar una distribución que esté pensada para ser utilizada en el sistema que tenemos.

aún utiliza un sistema de empaquetado muy rústico: el viejo **.tar.gz**. De más está decir que las distribuciones que autodetectan todo, que preguntan poco y que configuran todo de forma automática no son ideales para hacer tuning, ya que nos darán mucho más trabajo que las distribuciones como Debian, Gentoo y Slackware. Esto se debe a que éstas, al no tener casi nada preconfigurado, nos dejan el camino libre para definir lo que queremos en lugar de tener que eliminar decenas de configuraciones que nos son inútiles para luego configurar el sistema como queremos.

## Consejos para mejorar la experiencia de la usabilidad

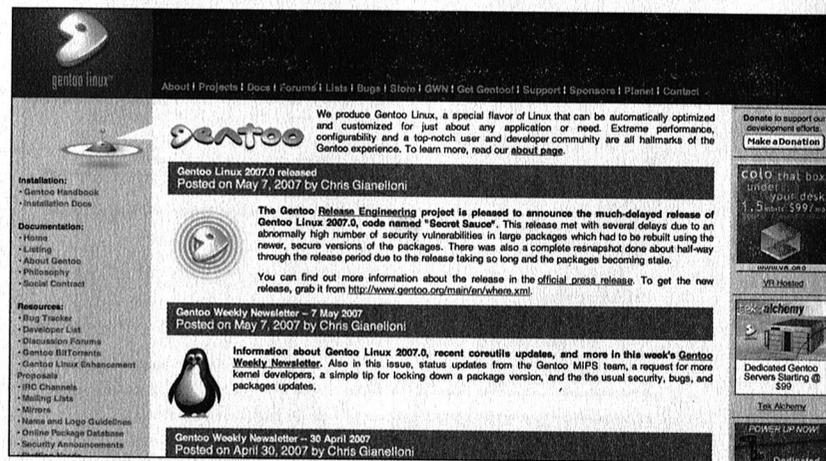
A continuación vamos a conocer algunas recomendaciones para mejorar la usabilidad del sistema operativo:

- Linux tiene **escritorios virtuales** que hay que aprovechar. Los usuarios de Windows y Mac OS X quedan fascinados con esta funcionalidad. En las versiones nuevas de KDE incluso es posible mover las ventanas de un escritorio a otro con sólo presionar sobre el cuadrado que las representa en la sección escritorios de la barra de tareas.
- Uno de los mayores logros en cuanto a usabilidad de Mac OS X es el sistema **Exposé**, que permite gestionar las ventanas fácilmente. Por ejemplo, cuando tenemos muchas ventanas abiertas y no sabemos en dónde está un programa en particular, si presionamos **F9** todas las ventanas se alejan de la pantalla para que podamos elegir cuál acercar. Bueno,

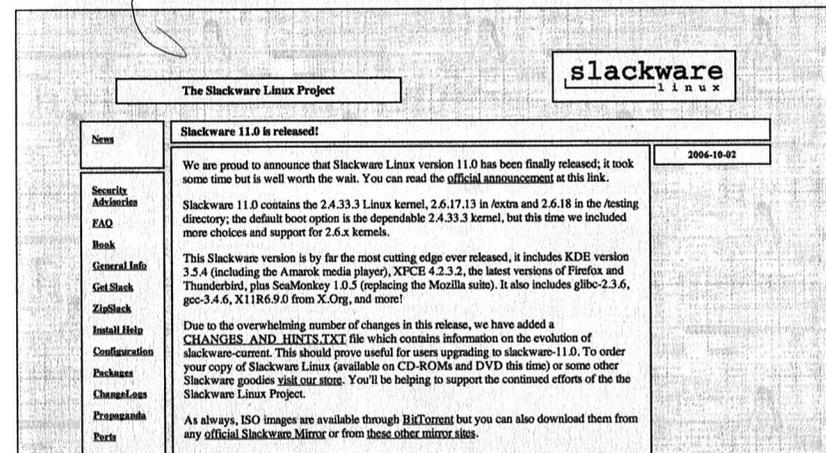
## OPTIMIZAR LA USABILIDAD

La **usabilidad** es el grado de efectividad que ofrece el gestor de escritorios para utilizar el sistema operativo. Tanto KDE como GNOME nos ofrecen una infinidad de herramientas para personalizar la interfaz y así poder mejorar la experiencia de usabilidad, pero como usuarios también debemos hacer nuestra parte. En primer lugar, debemos evitar que el escritorio esté repleto de iconos. Éste sólo debe tener los iconos que sean absolutamente necesarios, como un enlace a las unidades de disco duro, a la papelera de reciclaje y algún documento en el que estemos trabajando.

ahora los usuarios de KDE pueden tener esa funcionalidad gracias a **Komposé** (<http://kompose.berlios.de>). La instalación es muy sencilla y no requiere de librerías raras. Los usuarios de GNOME también tienen su versión de esta funcionalidad llamada **Skippy** (<http://thegraveyard>).



**Figura 3.** ([www.gentoo.org](http://www.gentoo.org)) es una distribución en la que hay que compilar de los fuentes prácticamente todos los componentes del sistema.



**Figura 4.** Slackware se considera una distribución para entendidos ya que no ofrece muchas herramientas de autoconfiguración del sistema.

org/skippy.php). Si tenemos Debian o un derivado, lo podemos instalar con **apt-get install skippy**.

- No es conveniente llenar el escritorio de iconos ya que éstos dificultan la visibilidad y la gestión de la información en nuestro trabajo cotidiano. En cambio, es mucho mejor utilizar los directorios que vienen

creados en la distro para gestionar los datos personales.

- Aquellos usuarios de GNOME que quieran un **dock** (un panel con las aplicaciones más usadas y los botones para acceder a ubicaciones del sistema de archivo rápidamente), pueden instalar **Gimmie** (<http://beatnik.infogami.com/Gimmie>).

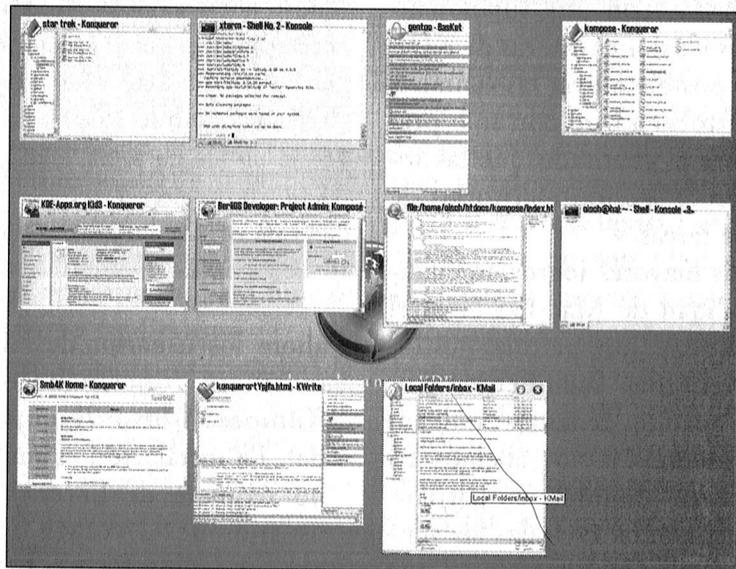


Figura 5. Komposé simula el efecto Exposé de Mac OS X en KDE.



Figura 6. Gimmie es un excelente reemplazo del menú de inicio para GNOME.

Los usuarios de KDE que quieran más cosas en el escritorio y más funcionalidad a la hora de personalizar la barra de tareas, deben instalar **SuperKaramba** (<http://netdragon.sourceforge.net/ssuperkaramba.html>).

## Cómo optimizar el proceso de inicio de Linux

El proceso de inicio de GNU/Linux es uno de los procesos más largos, cansadores y tediosos para todo el mundo. Se toma como mínimo un minuto para entrar a modo texto por una simple razón: se cargan muchos **servicios**. Ahora bien, lo más probable es que no utilicemos muchos de estos servicios que se están cargando en nuestra computadora. Por ejemplo, una instalación típica de Debian, SuSE o Mandriva nos instala un servidor de **base de datos, mail, ssh,**

## OTROS SERVICIOS...

Es muy posible que encontremos otros servicios que no mencionamos ni vemos listados aquí. En tal caso, para resolver nuestras dudas, una buena opción es recurrir a la documentación que se encuentra instalada en el directorio `/usr/doc` (o `/usr/share/doc`) para tener una idea más clara de la utilidad de cada uno de ellos.

**ftp, web, samba,** etcétera. Muchos de esos servicios se cargan automáticamente al inicio, y si no los usamos, simplemente están allí para hacernos perder la paciencia cuando encendemos nuestra computadora. Entonces, lo que hay que hacer para optimizar el proceso de inicio de GNU/Linux es **sacar servicios** que no utilicemos. Más tarde, si necesitamos alguno de ellos, podemos cargarlo manualmente. Ya veremos cómo.

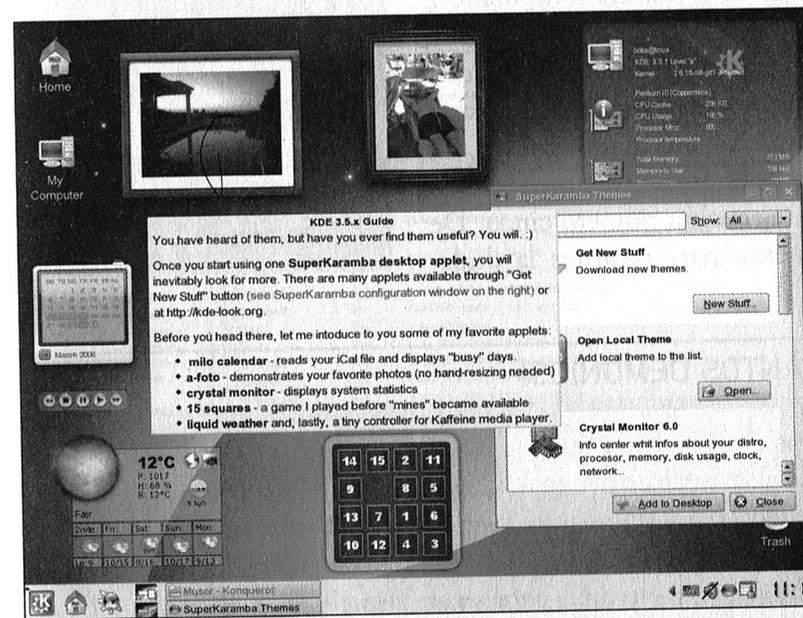


Figura 7. Si queremos adornar el escritorio KDE, SuperKaramba es una excelente opción.

El sistema GNU/Linux tiene definidos lo que llamaremos **niveles de ejecución**. Un nivel de ejecución es una definición de qué servicios se levantan o se bajan según el estado del sistema. Por ejemplo, el nivel de ejecución correspondiente al inicio de la computadora tiene configurados los servicios que se cargarán al inicio. El nivel de ejecución correspondiente al apagado de la computadora tendrá la configuración de cuáles son los servicios que deben ser bajados antes de apagar el sistema. Bien, lo que nosotros tenemos que lograr es que en nuestro nivel de ejecución del inicio sólo se carguen los servicios que necesitamos sí o sí para usar nuestra computadora. Es decir, aquellos que sean prescindibles vamos a quitarlos. Recordemos que si más adelante los necesitamos, los podemos levantar a mano.

El nivel de ejecución correspondiente al inicio del sistema es el tres o el cinco, dependiendo de si nuestra computadora bootea automáticamente en modo texto o en modo gráfico. Si utilizamos alguna distribución popular que bootea de manera automática en modo gráfico, lo más probable es que el sistema boote en el nivel de ejecución número cinco. De todas maneras, para sacarnos la duda,

vamos a visitar el archivo `/etc/inittab` y buscar la siguiente línea:

**id:5:inittdefault:**

Si en esa línea encontramos el número **5** definido, significa que el equipo bootea automáticamente en modo gráfico. Si queremos cambiarlo, podemos poner un **3** y entonces la computadora booteará automáticamente en modo texto.

Bien. Ahora que sabemos que nuestro sistema bootea en el nivel de ejecución número cinco, vamos a ver cuáles son los servicios que se cargan de manera automática cuando inicia el sistema. Para esto debemos dirigirnos al directorio `/etc/rc.d/rc5.d` (o `/etc/rc5.d` en las distribuciones Debian y sus derivadas). Allí ejecutamos `ls` y nos encontramos con los siguientes archivos:

```
K55routed@      S11mysql@
S20xfs@         S56xinetd@
S90webmin@
K59dund@        S11portmap@
S24messagebus@ S60nfs@
S91smb@
K59hidd@        S12syslog@
S25bluetooth@  S75keytable@
S92httpd@
```

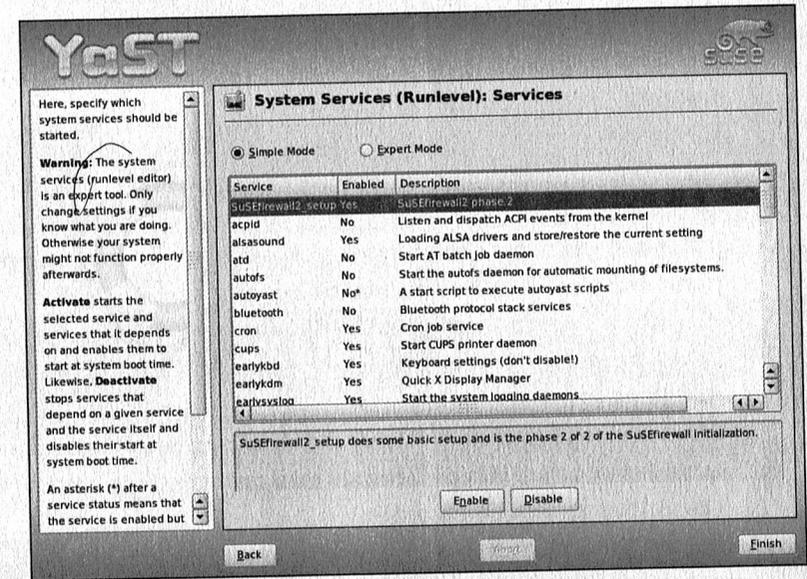
```
K59pand@        S13partmon@
S25netfs@       S80postfix@
S95kheader@
K01udev@        S14nfslock@
S30dm@          S85numlock@
S98mailman@
K03iptables@   S15gpm@
S40atd@
S85postgresql@ S99local@
K05harddrake@  S17alsa@
S55sshd@        S85proftpd@
K10network@    S18sound@
S56rawdevices@ S90crond@
[facundo@dhcp-3223-178 rc5.d]$
```

¿Qué es todo esto? ¿Chino? No. Son todos los servicios que se están ejecutando de manera automática al inicio del sistema. En realidad no son todos, sino que son aquellos que empiezan con la letra **S**. La manera ruda de hacer las cosas es borrar de este directorio to-

dos los archivos correspondientes a los servicios que no queremos que se carguen al inicio. No hay problema al hacer eso porque estos archivos no representan a los servicios en sí mismos sino que son simples enlaces simbólicos que luego se pueden volver a crear. Sin embargo, decimos que ésta es la manera ruda porque en realidad lo ideal sería utilizar la herramienta de configuración de la distribución que estemos usando para habilitar y deshabilitar los servicios del arranque (**Figura 8**). Tanto si optamos por la manera ruda como por la manera ideal, tenemos que saber para qué sirve cada uno de estos servicios, o si no nunca vamos a saber si nos son de utilidad o no. Por eso, veamos un pequeño listado de nombres de servicios populares en cualquier distribución para poder decidir si los necesitamos cargados automáticamente al inicio o no.

**III ¡CUÁNTOS DEMONIOS!**

Podemos notar que cuando `httpd` (o `apache`) está corriendo en nuestro equipo, muchos procesos iguales están activos en el sistema. Esto es porque generalmente, este servidor levanta muchos demonios al inicio. Aunque en realidad esto no consume demasiados recursos, si no lo usamos podemos dar de baja al servicio y ver cómo nuestro listado de procesos puede entrar en una sola pantalla para facilitarnos la lectura.



**Figura 8.** Casi todas las distribuciones incluyen una herramienta para gestionar los servicios del sistema.

SERVICIO	DESCRIPCIÓN
<b>adsl</b>	Este servicio carga todos los módulos del sistema <b>pppoe</b> para conectarnos a Internet mediante redes ADSL. Si utilizamos otro método de conexión a Internet, podemos deshabilitarlo sin problemas.
<b>alsa</b>	ALSA son los módulos de sonido de GNU/Linux. Son esenciales para escuchar algo que salga por nuestros parlantes. No los deshabilitemos.
<b>atd</b>	Éste es un servicio que gestiona el cronometrado de tareas. Con el comando <b>at</b> podemos agendar tareas como a las 12 de la noche, quiero que se ejecute el comando <b>halt</b> . De esta manera, el servicio <b>atd</b> se irá encargando de ejecutar las tareas una por una y en el momento indicado. Si esta funcionalidad no nos es de interés, podemos deshabilitar este servicio.
<b>bluetooth</b>	Este servicio carga los módulos (drivers) necesarios para utilizar el sistema de comunicación Bluetooth en GNU/Linux. Hoy en día, muchos teléfonos celulares incluyen soporte para este protocolo, por lo que no es una mala idea dejarlo activado. O si no, podemos deshabilitarlo y habilitarlo sólo cuando tengamos algún dispositivo que soporte esta tecnología.
<b>crond</b>	Este servicio es similar al <b>atd</b> pero es bastante más avanzado. La diferencia principal entre ambos es que éste permite cronometrar tareas que se ejecuten automáticamente una vez por día, por mes, por semana, etcétera. No es recomendable deshabilitar este servicio ya que las distribuciones de GNU/Linux generalmente lo utilizan para cronometrar las tareas de mantenimiento del sistema.
<b>cupsd</b>	Éste es el servidor de impresión de GNU/Linux. En el Panel de control de Mandriva, en la sección <b>Hardware</b> , tenemos una interfaz visual para este sistema, por lo que conviene tenerlo funcionando. Pero, si no tenemos impresora, podemos deshabilitarlo tranquilamente.
<b>dhcp</b>	El servicio <b>dhcp</b> es el que ofrece una dirección IP de manera automática a los demás equipos de una red. Si tenemos una red y nuestros equipos reciben una dirección IP en forma automática, entonces es posible que se esté ejecutando este servicio. No es mala idea dejarlo habilitado ya que muchas tecnologías de conexión a Internet (como el cablemódem) lo utilizan para configurar la conexión.
<b>dm</b>	Es el servicio que se encarga de cargar el modo gráfico de Mandriva al inicio de la distribución. No lo deshabilitemos si queremos que cuando nuestro sistema inicie se cargue KDE en forma automática.
<b>exim</b>	Exim es un servidor de correo electrónico para GNU/Linux. Si no tenemos necesidad de nuestro propio servidor de correo, podemos deshabilitarlo para ganar recursos y seguridad.
<b>gdm</b>	Éste es el gestor de sesiones gráficas de GNOME (similar al servicio <b>dm</b> que vimos antes). Las distribuciones como Debian, Ubuntu y sus derivados lo utilizan para cargar el modo gráfico automáticamente.
<b>gpm</b>	Se trata de un módulo para utilizar el mouse en modo texto. Por ejemplo, podemos seleccionar las opciones del menú de Midnight Commander con el puntero de nuestro mouse. Si no le vamos a dar uso, podemos deshabilitarlo también.
<b>haldaemon</b>	Es un sistema de configuración de hardware que recaba información de Internet y la analiza con la de nuestro sistema. Es bueno dejarlo habilitado para tener el hardware configurado siempre de la mejor manera.

SERVICIO	DESCRIPCIÓN
<b>harddrake</b>	Ésta es la herramienta de autodetección de hardware de Mandriva. Conviene dejarla habilitada en esta distribución también.
<b>hid</b>	Funciona en conjunto con el servicio <b>bluetooth</b> y ofrece soporte para los dispositivos de control inalámbricos, como el mouse y el teclado. Si todos nuestros periféricos usan cables, podemos deshabilitarlo.
<b>httpd (o apache)</b>	Se trata del servidor web más popular. La recomendación que dimos para <b>exim</b> es igualmente válida: si no vamos a brindar servicios web, podemos deshabilitarlo y ganar recursos y seguridad.
<b>iptables</b>	Es el sistema de firewall (seguridad) de GNU/Linux. Es conveniente dejarlo habilitado (en el capítulo de seguridad veremos cómo manejarlo).
<b>kdm</b>	Éste es el gestor de sesiones gráficas de KDE, y es lo que el servicio <b>dm</b> utiliza para cargar la interfaz gráfica en Mandriva.
<b>keytable</b>	Éste servicio en general configura de forma automática la configuración del idioma de nuestro teclado. No consume muchos recursos, por lo que podemos dejarlo habilitado. Recordemos que podemos configurarlo desde el panel de control de Mandriva, en la sección <b>Hardware</b> .
<b>mysql</b>	Se trata del motor de base de datos MySQL. Si no planeamos desarrollar aplicaciones o utilizar programas que funcionen con bases de datos, podemos deshabilitarlo. Ganaremos una mejor performance del sistema.
<b>network</b>	En casi todas las distribuciones, este servicio configura todos los parámetros de nuestra red. Pero si nuestra computadora no forma parte de ninguna red (inclusive Internet), entonces podemos deshabilitarlo.
<b>nfs</b>	Este servicio gestiona la carga automática del sistema de archivos de red o NFS (siglas de <i>Network File System</i> ). Este sistema nos permite exportar directorios para que puedan ser accesibles desde otras máquinas con GNU/Linux o alguna versión de UNIX. En cuanto a deshabilitarlo, vale la misma recomendación que para los demás servidores: si no lo usamos lo podemos quitar.
<b>numlock</b>	Este servicio habilita y deshabilita la tecla <b>NUMLOCK</b> . Es imprescindible.
<b>portmap</b>	Se trata del soporte de muchos otros servicios que funcionan bajo el protocolo UDP (no TCP/IP), como NFS. Si vamos a usar NFS, entonces es absolutamente necesario que este sistema funcione correctamente.
<b>postfix</b>	Análogo al servicio <b>exim</b> .
<b>postgresql</b>	Análogo al servicio <b>mysql</b> .
<b>proftpd</b>	Gestiona la carga automática del servidor de archivos FTP (transferencia de archivos por el protocolo TCP/IP) <b>proftpd</b> . Si no vamos a utilizar nuestra computadora como un servidor FTP, entonces podemos deshabilitarlo.
<b>sendmail</b>	Éste es el servidor de correo electrónico más viejo de UNIX, que aún hoy viene incluido en muchas distribuciones. Es útil que esté habilitado ya que muchos otros programas lo utilizan para enviar comunicados internamente.
<b>shorewall</b>	Es una herramienta de configuración del firewall <b>iptables</b> . En el capítulo de seguridad veremos cómo configurar el firewall sin herramientas, por lo que podemos deshabilitarlo.
<b>smbd</b>	Carga automáticamente el servidor Samba (de interconexión con redes Windows).

SERVICIO	DESCRIPCIÓN
sshd	El servidor de acceso remoto <b>ssh</b> permite que desde otra computadora nos conectemos a la nuestra para así controlarla como si estuviéramos con las manos sobre el teclado. Lo utilizaremos en el próximo capítulo, por lo que no es mala idea dejarlo habilitado.
syslog	Este servicio es el que registra toda la actividad del sistema en los denominados archivos log. Estos archivos se ubican en el directorio <code>/var/log</code> y les permiten a los administradores tener un control sobre lo que pasa en el sistema. Es bueno dejarlo habilitado.
vsftpd	Análogo a <b>proftpd</b> .
webmin	Es un sistema de configuración y administración del sistema mediante el navegador web. Permite hacer diversas tareas como administrar usuarios, configurar servicios, configurar la red, armar firewalls y más. Podemos deshabilitarlo si no nos interesa.
xfs	Éste es el servidor de fuentes del sistema gráfico X. Es indispensable.
xinetd	Éste es el llamado súper servidor del sistema, y se encarga de gestionar la carga y descarga de varios servicios de red. Con sólo cargar XinetD, nuestro sistema puede estar cargando los servicios de red que dependan de él y que no aparezcan en este listado. Por ejemplo, algunos servicios que funcionan comúnmente bajo XinetD son <b>telnet</b> , <b>vnc</b> , <b>ftp</b> , <b>samba</b> , <b>swat</b> (un sistema de configuración de Samba vía web), entre otros. Por eso, una vez que hayamos sacado todos los servicios que no queremos del listado de servicios que se ejecutan automáticamente al inicio de nuestra computadora, el próximo paso será sacar los servicios que son controlados por XinetD. Para eso, entramos al directorio <code>/etc/xinetd.d/</code> . Allí nos encontraremos con varios archivos. Cada archivo corresponde a un servicio gestionado por el súper servidor. Son archivos de texto, por lo que podemos abrirlos con cualquier editor. Para deshabilitarlos, tenemos que poner yes en la línea que dice <code>disable</code> de cada servicio.

**Tabla 1.** Listado de **servicios** más comunes en el nivel de ejecución del inicio del sistema.

Con seguridad, luego de haber seleccionado minuciosamente todo lo que necesitamos, la carga inicial de nuestro sistema operativo tomará mucho menos tiempo y sentiremos que todo funciona más rápido. Ahora veamos cómo optimizar el rendimiento del entorno gráfico **KDE**.

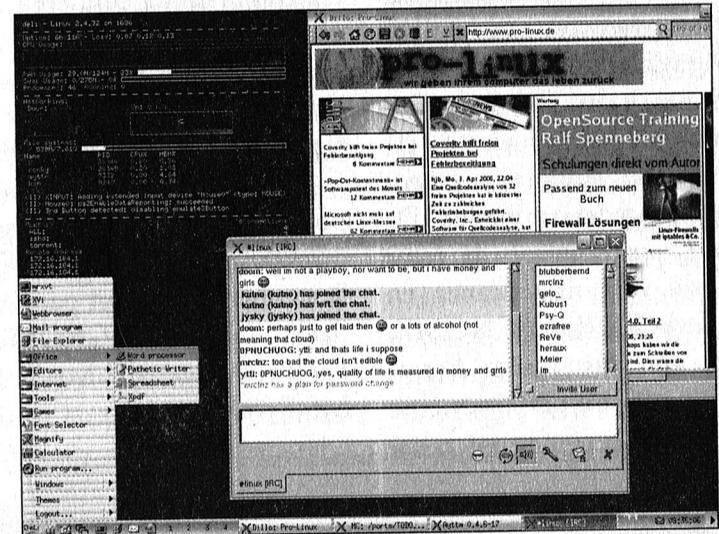
### Cómo acelerar GNU/Linux en sistemas con pocos recursos

En computadoras viejas (digamos, Pentium 200Mhz) con poca memoria RAM (64 MB de RAM), cualquier distribución popular actual se arrastrará frente a

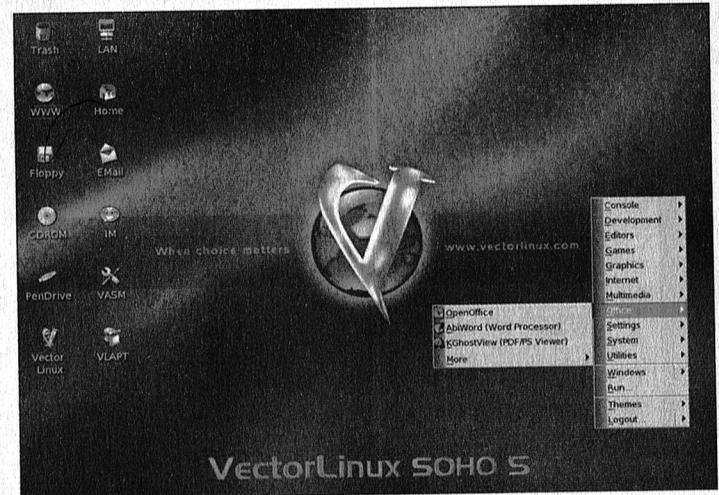
nuestros ojos al punto de llegar a parecer imposible de utilizar debido a los largos tiempos de carga. Por eso, es necesario que sepamos que existen sistemas operativos y componentes que nos permiten hacer lo mismo que los programas más populares pero con menos recursos. Comencemos por las distribuciones de GNU/Linux. En primer lugar, podemos asegurar que tanto Debian como Slackware se pueden instalar en computadoras viejas de las características que mencionamos o incluso con menos recursos. Eso sí, no podemos pretender usar KDE o GNOME en ellas (más adelante conoceremos algunos

manejadores de ventanas y aplicaciones que andarán mucho mejor). Si queremos una distribución especialmente diseñada para las computadoras con bajos recursos, entonces podemos optar por instalar **DeLi Linux** ([www.delilinux.de](http://www.delilinux.de)), que es una distribución armada con componentes livianos y

funcionará sin problemas en una 486 con 16 MB de RAM (**Figura 9**). Otra opción es utilizar alguna de las primeras versiones de **VectorLinux** también ([www.vectorlinux.com](http://www.vectorlinux.com)). Cualquier versión anterior a la 4 funciona muy bien en las máquinas con pocos recursos (**Figura 10**).



**Figura 9.** DeLi Linux es una distribución ideal para los equipos viejos que incluye un montón de software de gran utilidad.



**Figura 10.** VectorLinux es otra opción a la hora de elegir una distribución para un equipo con pocos recursos.

hora bien, si ya tenemos la distribución instalada, entonces conviene optar por utilizar otras herramientas en lugar de las populares para que todo funcione bien. En primer lugar, es muy recomendable **cambiar KDE y GNOME** por algún manejador de ventanas más liviano como **IceWM (www.icewm.org)**, **Window Maker (www.windowmaker.info)** o **FVWM (www.fvwm.org)**. Todos ellos vienen incluidos en casi todas las distribuciones, y casi todas éstas incluyen herramientas para hacer de esos manejadores de ventanas nuestros sistemas visuales predeterminados. De más está decir que ninguno de estos manejadores de ventanas alcanza en funcionalidades a KDE o a GNOME, pero tampoco son austeros en herramientas, por lo que son perfectamente



Recordemos que, siempre que sea absolutamente necesario, podemos correr Microsoft Word bajo Linux con herramientas como CrossOver Office o Wine.

utilizables y aun hoy muchos expertos en GNU/Linux prefieren usar uno de estos sistemas a usar KDE o GNOME. En cuanto a las herramientas de uso cotidiano, conviene reemplazar OpenOffice por **AbiWord (www.abiword.org)** y Gnumeric (**www.gnumeric.org**) o por Siag Office (**http://siag.nu**). En el caso de AbiWord y GNUMeric, estas herramientas forman parte de lo que se llama el proyecto **GNOME Office**, y permiten cargar archivos de MS Word y MS Excel.

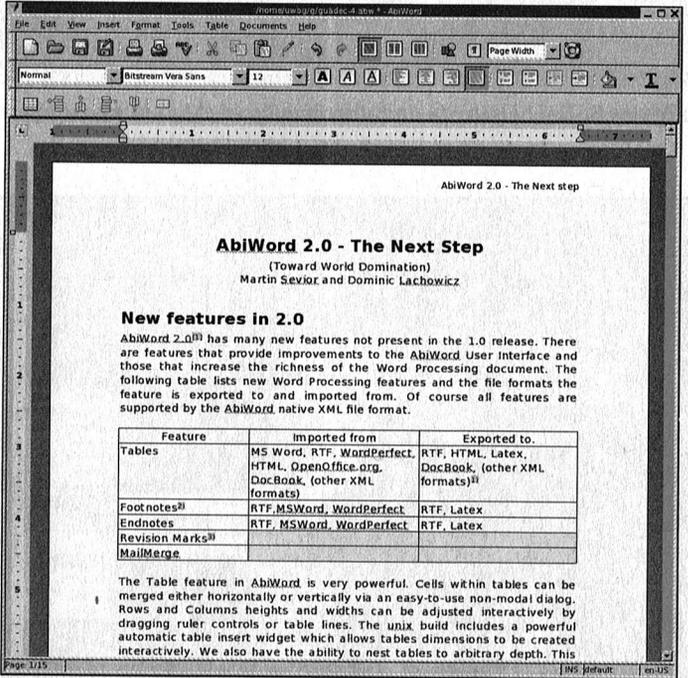


Figura 11. AbiWord es un excelente procesador de texto que contiene casi todas las funcionalidades de OpenOffice Writer pero no consume tantos recursos.

Para navegar, muchos consideran a **Opera (www.opera.com)** el navegador más rápido del planeta. Incluso, VectorLinux lo incluía como navegador principal ya que funcionaba muy bien en las computadoras con pocos recursos. De más está decir que Opera es altamente compatible con los estándares actuales de la Web y no tendremos problemas para navegar. Otros navegadores livianos son **Konqueror** (del proyecto KDE) y **Epiphany (www.gnome.org/projects/epiphany)**. Éste último es el navegador oficial del proyecto GNOME y es más rápido que Konqueror, Mozilla y Firefox, a costa de algunas funcionalidades que aún no tiene. En cuanto a los navegadores, no podemos dejar de mencionar a **Dillo (www.dillo.org)**, un navegador que no es muy compatible con los estándares web actuales pero cuyo archi-

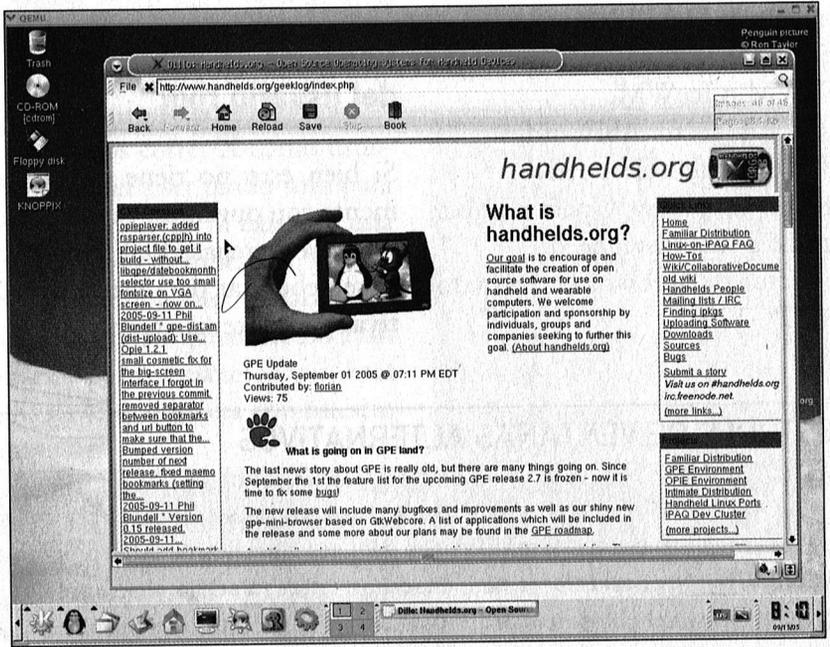


Figura 12. Dillo es un navegador que ocupa muy poco espacio, lo que lo convierte en el ideal para los sistemas de bajos recursos.

## BLUETOOTH NO ES WIFI

Es muy común que se confunda **Bluetooth** con **WiFi** y se habilite este servicio esperando que una red inalámbrica funcione mágicamente. Sin embargo, eso es un error: **Bluetooth** y **WiFi** son dos protocolos diferentes. Vale la pena investigar la utilidad de cada uno para saber cuál activar según nuestras necesidades particulares.

vo binario ocupa tan sólo 350 kb. ¿Qué más se puede pedir? Si estamos buscando un **cliente de correo electrónico** que funcione más rápido y sea más liviano que Evolution, entonces **Sylpheed (http://sylpheed.sraoss.jp/en/)** es una buena opción para elegir. Se trata de un cliente de correo electrónico desarrollado en **GTK**,



Cuando nos desanimemos porque la distribución que acabamos de instalar en la computadora funciona demasiado lento, recordemos que existen versiones de Linux que corren en computadoras Atari ST con 1MB de RAM (equipos que datan de hace más de veinte años). Si Linux corre ahí, corre en cualquier cosa. ¡Sólo es cuestión de invertir tiempo en configuración y adaptación!

Para el resto de las herramientas hay muchas opciones, pero dependerá de cada uno de nosotros elegir las. Como consejo general, hay que tratar de **evitar todas las aplicaciones desarrolladas en Java**, ya que resultan muy pesadas hasta en las máquinas actuales. Esto se debe al diseño de la máquina virtual Java que fue desarrollado pensando en la **portabilidad** y no en la velocidad.

### Consejos de uso del usuario root

Si bien esto no tiene que ver directamente con optimizar nuestro sistema, es importante repararlo. El usuario **root** sólo se debe usar para **tareas administrativas** del sistema, como su configura-

ción. Pero para las tareas cotidianas como leer e-mails, navegar por la Web, escuchar música, etcétera, deberíamos usar un usuario que no tenga privilegios de administrador. Generalmente, basta con que pertenezca al grupo **users**. De esta forma, protegemos nuestro sistema de ataques, ya sea de virus como de códigos maliciosos, nada ni nadie que use nuestra cuenta podría llegar a tocar más que nuestro directorio **home**.

El usuario **root** es el usuario más importante de nuestro sistema. De hecho, si un atacante obtiene acceso a nuestro **root**, la única forma de volver a confiar nuevamente en nuestro sistema es reinstalándolo. Para evitar esto hay ciertas reglas que debemos tener en cuenta.

- Siempre crear un usuario para el uso diario y si este usuario necesita cada tanto el acceso a **root** agregarlo al grupo **wheel**. Esto hace posible que un usuario normal use el comando **su** para tener temporalmente los privilegios de administrador.
- Nunca debemos correr X como usuario **root**. Éste debe ser usado sólo para tareas administrativas. Si existiera una vulnerabilidad en la aplicación y la estamos corriendo como usuario, un atacante sólo podría ganar acceso a nivel usuario, pero si estamos con **root** puede hacer lo que quiera con nuestro sistema.
- Si un usuario debe correr un par de comandos como **root**, en lugar de darle todos los privilegios de súper usuario podríamos utilizar **sudo**, aunque también hay que tener cuidado con esto.
- Nunca debemos dejar una terminal abierta cuando estamos conectados como **root**.

## Cómo optimizar el desempeño de nuestros discos IDE con hdparm

Algo importante que debemos hacer es verificar el buen desempeño de nuestros discos duros. Esto lo hacemos con el programa **hdparm**, que podemos descargar de <http://freshmeat.net/projects/hdparm>. Antes de usarlo debemos tener en cuenta que según la documentación, por un mal uso de esta utilidad podemos llegar a perder datos o hasta dañar nuestros discos.

El comando **hdparm** nos permite visualizar y modificar los parámetros de nuestros discos rígidos IDE. Para ver qué parámetros tenemos activados en nuestro disco escribimos:

### # hdparm /dev/hda

Un ejemplo de lo que muestra ese comando es lo siguiente:

```
/dev/hda:
multcount      = 16 (on)
IO_support     = 1 (32-bit)
unmaskirq     = 1 (on)
using_dma      = 1 (on)
keepsettings   = 0 (off)
readonly       = 0 (off)
readahead     = 256 (on)
geometry      = 59323/16/63,
sectors = 30616363008,
start = 0
```

Para visualizar los parámetros que soporta nuestro disco rígido ponemos en la línea de comandos:

con muchísimas funcionalidades y diseñado desde el principio con el objetivo de que su funcionamiento sea muy rápido. Finalmente, si necesitamos escuchar archivos **MP3**, quizá los reproductores populares actuales para el modo gráfico sean demasiado para usar en los equipos con poca memoria RAM. Por ese motivo, una excelente opción es utilizar **MPG123** ([www-ti.informatik.uni-tuebingen.de/~hippm/mpg123.html](http://www-ti.informatik.uni-tuebingen.de/~hippm/mpg123.html)). Éste es un reproductor de archivos MP3 para la consola que, si bien se maneja por medio de parámetros, no es muy difícil de poner en funcionamiento.

En resumen, la recomendación para utilizar GNU/Linux de manera óptima en una computadora con poca capacidad de hardware (digamos, 32 MB de RAM y 100MHz de procesador) es:

- Distribución: Debian, Slackware, VectorLinux o DeLi Linux.
- Manejador de ventanas: Window Maker, FVWM o IceWM.
- Procesador de texto: AbiWord.
- Planilla de cálculo: Gnumeric.
- Navegador: Epiphany, Opera o Dillo.
- Cliente de e-mail: Sylpheed.
- Reproductor de archivos en formato MP3: MPG123.

## III GESTORES DE VENTANAS ALTERNATIVOS

Mucha gente nota que KDE consume demasiados recursos de su equipo y opta por instalar otros gestores de ventanas mucho más livianos. Algunos ejemplos de estos programas son: **FluxBox** (<http://fluxbox.sourceforge.net>), **Blackbox** (<http://blackboxwm.sourceforge.net>), **Enlightenment** ([www.enlightenment.org](http://www.enlightenment.org)) y **Openbox** (<http://icculus.org/openbox>). La mayor parte de ellos está incluida en las distribuciones más populares.



Uno de los problemas que tienen todos los usuarios de GNU/Linux está relacionado con la instalación de aplicaciones. Para simplificar las cosas, lo mejor es buscar aplicaciones que estén en el formato de empaquetamiento de la distribución que usemos. Esto nos permitirá utilizar las herramientas de gestión de paquetes del sistema, simplificando el proceso de instalación.

# **hdparm -i /dev/hda**

Uno de los parámetros que puede **acelerar el rendimiento** del sistema es la habilitación del sistema **UDMA**. Para habilitarlo, debemos teclear:

# **hdparm -d1 /dev/hda**

Finalmente, para ver la **velocidad de transferencia** de nuestro disco debemos cerrar la mayoría de los programas y servidores que estén corriendo en nuestra PC (también el modo gráfico) y ejecutar:

# **hdparm -Tt /dev/hda**

Para más información acerca de cómo utilizar este comando debemos leer su ayuda. Esto se hace con:

# **hdparm -help**

O en su página del manual, a la que accedemos mediante:

# **man hdparm**

## Cómo instalar programas desde el código fuente (.tar.gz)

**Compilar** un programa significa obtener un **binario óptimo** para nuestro sistema, ya que se genera según las características de nuestro procesador y demás recursos disponibles. Por eso, si tenemos paciencia, compilar muchos programas de uso frecuente derivará también en una mejora en el rendimiento. Como mencionábamos al principio de esta sección, los programas distribuidos como código fuente vienen en formato **.tar.gz**. En general, funcionan en todas las distribuciones de GNU/Linux siempre y cuando en ellas estén todas las **librerías de soporte** y otras aplicaciones requeridas. Por eso, siempre es importantísimo que antes de proceder a instalar un programa desde su código fuente leamos atentamente el archivo **INSTALL** que viene incluido en todos estos paquetes. El proceso de instalación de un paquete de código fuente se puede resumir en los siguientes pasos:

1. Copiar el paquete a un directorio temporal y descomprimirlo con el comando **tar -zxvf archivo.tar.gz**.
2. Ejecutar el comando **./configure**, que se encargará de verificar que el sistema esté listo para compilar el programa. Si no lo está nos informará qué componentes no tenemos y debemos instalar (siguiendo este procedimiento o el correspondiente al formato de empaquetamiento).
3. Iniciar el proceso de compilación con el comando **make**. Esto puede tardar algunos minutos o algunas horas dependiendo del programa y de las capacidades de nuestro equipo.
4. Cuando la compilación termina, podemos proceder con la instalación (que consiste en la copia de cada componente a su lugar correspondiente) con el comando **make install**.

El proceso puede variar según el programa, pero generalmente son los mismos cuatro pasos. Para evitar inconvenientes, siempre leamos toda la documentación contenida en el programa. Allí se nos indicará si hay alguna diferencia sustancial en cuanto a la compilación de dicho programa.

## Cómo saber si nuestro sistema está seguro

A la hora de buscar información sobre algún **host** de nuestra red de forma remota, tenemos que llevar a cabo un primer paso primordial: verificar qué **puertos abiertos** existen en el sistema. De esta forma, tendremos conoci-

miento de cuáles son las **puertas de acceso** que posee. Ésta es una tarea que debemos realizar:

- Cuando instalamos un sistema operativo y queremos ver con qué servicios viene activado por defecto.
- Cuando notamos cierto tráfico extraño en la red.
- Cuando un usuario nos reportó un ataque a su máquina.
- Cuando hacemos auditorías de seguridad en la red.

**Nmap** permite realizar rastreos sumamente rápidos a hosts remotos e incluso a nuestro host local. Entre otras cosas, nos informa sobre los puertos abiertos, las versiones de los servicios instalados en dicho servidor y, en algunos casos, qué sistema operativo está corriendo dicho host. Podemos obtener Nmap del sitio oficial del proyecto, en **http://insecure.org/nmap**. Una vez instalado (es un proceso muy sencillo de compilación), hagamos entonces nuestro primer rastreo a un host:

```
Alternarama:~# nmap 10.0.0.1

Starting nmap 3.50 (
  http://www.insecure.org/nmap/)
at 2004-06-09 14:28 ART
Interesting ports on 10.0.0.1:
(The 1657 ports scanned but not
shown below are in state:
closed)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
```

## III ¿Y CON LOS ARCHIVOS .TAR.BZ?

Muchos paquetes se distribuyen actualmente en el formato **.tar.bz**. En lugar de usar **gzip**, los programadores optaron por **bzip**, otro programa para comprimir. Para descomprimir estos archivos, simplemente debemos tipear **bzip2 -cd archivo | tar xvf**.

```
Nmap run completed - 1 IP
address (1 host up) scanned
in 10.031 seconds
Alternarama:~#
```

Como podemos observar, el único parámetro básico que requiere Nmap para funcionar es la dirección IP del host que queremos rastrear. Esto nos devuelve en pantalla solamente los puertos que están funcionando. En el caso del ejemplo, solamente está abierto el servicio FTP y SSH.

Si queremos obtener más información acerca del funcionamiento de Nmap cuando está rastreando un determinado host, podemos usar la opción `-v`. Veamos otro ejemplo:

```
The SYN Stealth Scan took 4
seconds to scan 1659 ports.
Interesting ports on 10.0.0.10:
(The 1642 ports scanned but not
shown below are in state:
closed)
```

```
PORT      STATE SERVICE
9/tcp    open  discard
13/tcp   open  daytime
21/tcp   open  ftp
22/tcp   open  ssh
```

```
Nmap run completed - 1 IP
address (1 host up) scanned
in 3.892 seconds
Alternarama:~#
```

### III SEGURIDAD TOTAL

A veces estamos trabajando de forma local en nuestro equipo y no necesitamos ningún servicio de red. En esos casos, lo mejor que podemos hacer para aumentar la seguridad del equipo es deshabilitar la interfaz de red. Para eso, simplemente utilizamos el comando `ifconfig eth0 down` (cambiar `eth0` por el nombre de la interfaz de red que corresponda).

En este caso, podemos ver que estamos frente a un host que tiene algunos problemas graves de seguridad, ya que tiene muchos puertos abiertos que, en su gran mayoría, son puertos que corresponden a servicios que tienen un largo historial de inseguridad. Por ejemplo, algunos de ellos son los siguientes: `telnet`, `finger` y `daytime`, entre otros.

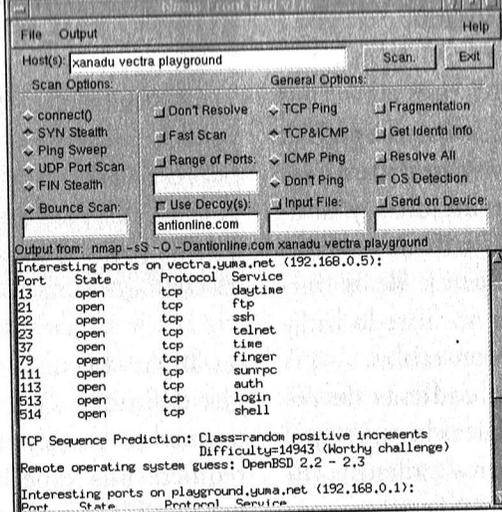


Figura 13. NmapFE es una interfaz visual para Nmap que ofrece rápido acceso a todas sus funcionalidades.

## Trucos para sacarle el máximo provecho a Nmap

Nmap provee, además, varios tipos diferentes de escaneos que se definen con el parámetro `-s[x]`. A saber:

PARÁMETRO	CARACTERÍSTICAS
<code>-sS</code> (TCP SYN scan)	No se abre una conexión TCP completa sino que simplemente se envía un paquete SYN para saber si el puerto está abierto. Éste es el modo de escaneo por defecto para los usuarios privilegiados.
<code>-sT</code> (TCP connect() scan)	Es el modo más básico de escaneo ya que utiliza la función <code>connect()</code> del sistema operativo. Si usamos este modo, lo más probable es que en el host destino queden muchos registros de nuestro escaneo en los archivos log.
<code>-sF -sX -sN</code> (stealth, árbol de Navidad y modo nulo)	Estos tres tipos de escaneo son realmente avanzados y pueden atravesar firewalls y otros sistemas de seguridad sin ser detectados.
<code>-sP</code> (escaneo vía ping)	Para saber qué hosts están encendidos.
<code>-sV</code> (detectar versión)	Este parámetro permite indicarle a Nmap que intente comunicarse con los puertos abiertos y detectar la versión del servicio que está ejecutando.
<code>-sU</code> (rastrear UDP)	Con este parámetro le indicamos a Nmap que rastree los servicios UDP de usuario abiertos en dicho host.
<code>-sO</code> (protocolos IP)	Este método sirve para rastrear qué servicios IP están funcionando en el host.
<code>-sA</code> (ataque ACK)	Este método de escaneo es una herramienta útil para determinar qué reglas de firewall están siendo utilizadas en dicho host.
<code>-sL</code> (listado de hosts)	Genera un listado de IPs y nombres de hosts.

Tabla 2. Lista de parámetros que se pueden utilizar con Nmap.

## Tutorial en 10 minutos de iptables

El **firewall** de Linux se llama iptables (su sitio oficial es [www.netfilter.org](http://www.netfilter.org)) y generalmente viene incluido y activado en todas las distribuciones que contengan núcleos a partir de la versión 2.4. Tal como su nombre lo indica, iptables funciona con tablas. Estas tablas contienen **cadena de reglas**. Por ejemplo, la tabla principal del sistema contiene tres cadenas: **INPUT**, **OUTPUT** y **FORWARD** (cadenas para definir reglas para lo que ingresa, lo que sale y lo que pasa por nuestro sistema). La forma básica del comando **iptables** es la siguiente:

```
iptables [DEFINICION] [CADENA] [OPCIONES] [ACCION]
```

El parámetro **[DEFINICION]** es un parámetro general. Aquí hay varias opciones posibles como utilizar el parámetro **A** para agregar una nueva regla a la cade-

## III ESCANEAR PUERTOS

En el **Capítulo 3** de este libro hemos visto otra forma de escanear puertos utilizando el comando **netcat**. Si bien esa herramienta no es tan potente como Nmap, seguramente nos sacará de un apuro.

na, **D** para borrarla o **L** para listar las reglas definidas. De más está decir que si usamos un parámetro de definición que requiere una especificación de cadena, debemos ingresarlo a continuación.

En el parámetro **[OPCIONES]** definimos todos nuestros filtros. Por ejemplo, todo lo que sale a Internet, todo lo que viene desde la IP 200.10.22.33 hacia mi host:80, todo lo que sale de mi máquina hacia la IP 200.122.33.22, etcétera. Finalmente, en el campo **[ACCION]** definimos qué hacer cuando una regla se cumple (lo permitimos, no lo permitimos, etcétera). Para comprender un poco mejor las cosas, veamos un ejemplo del mundo real:

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

En este ejemplo, hemos creado (**A**) una regla para la cadena de entrada (**INPUT**) en la cual definimos que todo lo que entre al puerto 80 del protocolo TCP (**-p tcp --dport 80**) sea aceptado (**-j ACCEPT**). El parámetro **--dport** requiere sí o sí que siempre usemos el parámetro **-p tcp**, si no no sabe de qué protocolo es el puerto definido. Veamos otro ejemplo:

```
iptables -A INPUT -p icmp -j DROP
```

En este ejemplo, simplemente filtramos y descartamos todo lo que ingresa por el protocolo ICMP a nuestra computadora. El puerto ICMP es el puerto utilizado por el comando **ping**, por lo que con esta regla, nadie podrá realizar un **ping** exitoso a nuestra computadora. Sigamos con los ejemplos:

```
iptables -A OUTPUT -p tcp -d 10.0.0.11 -j REJECT
```

Aquí, en lugar de definir un firewall para todo lo que entra, definimos un firewall para lo que sale de nuestro sistema y se dirige a la IP 10.0.0.11.

La acción **REJECT** rechaza dicho paquete. Como no hay definido ningún puerto en particular rechaza todo. Pero lógicamente podemos combinarlo con el ejemplo anterior:

```
iptables -A OUTPUT -p tcp -d 10.0.0.11 --dport 80 -j REJECT
```

Aquí, rechazamos todo lo que se dirija al puerto 80 del host 10.0.0.11. Es importante aclarar que no podemos poner las reglas de iptables en el orden que se nos plazca, porque lo más probable es que nos encontremos con un funcionamiento impredecible. Entonces, para evitar problemas, vamos a comenzar definiendo que iptables lee las reglas de arriba para abajo. Sí, suena un poco raro, pero es así. Veamos el siguiente ejemplo:

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j DROP
```

Como vemos, las reglas son contradictorias. Primero decimos que aceptamos el acceso al puerto 80 de nuestra computadora y luego decimos lo contrario. Bien,

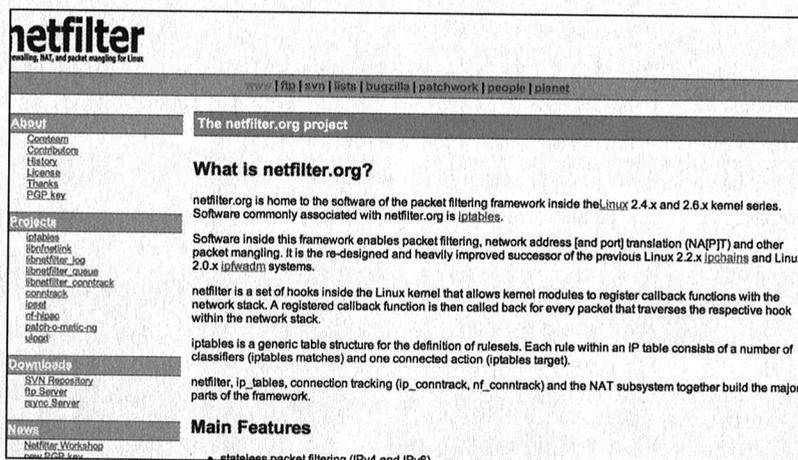


Figura 14. En el sitio oficial de iptables podemos encontrar mucha documentación relacionada con este sistema.

## III CUIDADO CON EL SERVICIO TELNET

Cuando controlemos los servicios gestionados por el súper servidor, prestemos especial atención a que el servicio **telnet** no esté corriendo en el sistema. Este servicio se trata de un protocolo muy antiguo de **acceso remoto**, que al no poseer ningún tipo de encriptación, es uno de los servidores más inseguros que existen en la actualidad. De hecho, está totalmente obsoleto y fue completamente reemplazado por SSH.



Tenemos dos opciones disponibles para probar que las reglas de iptables que estamos implementando funcionan de manera correcta. La forma más sencilla es utilizar Nmap y verificar el estado de los puertos filtrados. La segunda opción es un poco más incómoda, porque implica utilizar otra computadora como cliente para probar la comunicación.

es evidente que no entrarán en juego las dos reglas. Como mencionamos antes, iptables lee las reglas de arriba hacia abajo. Entonces, en el ejemplo anterior, lo que sucederá es que sí se permitirá el acceso al puerto 80. ¿Qué sucede con la segunda regla? Directamente se saltea. Una vez que iptables realizó lo pedido con respecto al puerto 80, **no vuelve a procesar reglas con filtros iguales**. En el caso anterior, la segunda regla directamente no tiene efecto.

Habiendo aclarado esto vamos al próximo paso: las **políticas**. Las políticas son **reglas** que se definen de manera predeterminada, como por ejemplo decir: por defecto, permito todo acceso a mi maquina o por defecto, niego todo acceso a mi maquina. Entonces, haciendo uso de las políticas y las re-

glas, tenemos que pensar seriamente cómo armar nuestro firewall.

Existen dos formas básicas para hacerlo, una de ellas es la **gran pared de fuego** en la que nosotros definimos por política que **negamos todo** el acceso a nuestro sistema y luego, por medio de reglas llamadas **excepciones**, abrimos pequeños **huecos** en la pared para permitir el acceso a ciertos puertos.

Es muy común que se piense que la política del **niego todo** es mucho mejor que la del **permiso todo**, pero debemos saber que en parte es así y en parte no. Cuando uno utiliza como política el **niego todo**, después tiene un arduo trabajo para abrir aquellos puertos que son necesarios para la vida cotidiana en la computadora: que tal usuario quiere el MSN, que tal otro quiere usar el Kazaa, que yo me quiero bajar un archivo con el BitTorrent, es realmente toda una cuestión. Ahora bien, la política del **acepto todo** tiene sus problemas también, porque tenemos que tener perfecto conocimiento de cuáles son los puertos abiertos en nuestra computadora, y así definir las reglas para que quienes no deban tener acceso a ellos no puedan hacerlo. Para definir políticas se utiliza el parámetro **-P** de la siguiente manera:

### III VERIFICACION DE LA POLÍTICA

Prácticamente todas las distribuciones incluyen en su configuración de **iptables** la política de permitir todo. Aun así, siempre es recomendable verificar esta configuración cuando se realicen modificaciones en el firewall, porque nos permitirá ahorrarnos mucho tiempo de investigación y pruebas. Para esto debemos tipear **iptables -L**, y así podremos ver en pantalla la configuración correspondiente.

### iptables -P INPUT DROP

En este caso negamos absolutamente todo lo que ingresa. En el próximo ejemplo, veremos cómo habilitar el ingreso de todos los paquetes:

### iptables -P INPUT ACCEPT

Como ejemplo final, supongamos que tenemos un **servidor proxy** con una sola placa de red. El servidor proxy se encarga de **compartir una conexión de Internet** a varias computadoras de una red interna que tienen direcciones del tipo 10.0.0.1, 10.0.0.2, 10.0.0.3, etcétera. Hasta ahora, todo bien. La cuestión es que el equipo tiene un **servidor Apache** corriendo en el puerto 80 y nuestra intención es que sólo se pueda ingresar en las páginas que contiene desde la red interna y no desde Internet. Nuestro firewall debería lucir como el siguiente (usando política de aceptar todo lo que ingresa):

### iptables -F

```
iptables -A INPUT -s 10.0.0.0/24
-p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport
80 -j DROP
```

### Cómo montar una DMZ en pocos pasos

Supongamos que nuestra computadora es el proxy de la red (o sea, la máquina que utilizan las demás computadoras de la red para conectarse a Internet). Y supongamos que una de las

### III BASES DE DATOS

Muchas personas, en su afán por instalar programas en Linux, instalan al mismo tiempo **PostgreSQL** y **MySQL**. Es importante tener en cuenta que cuando el sistema inicie, levantará ambos servidores, lo que representa una aniquilación de los recursos de nuestro equipo. En general, es recomendable elegir alguno de los dos servidores o hacer que sólo uno de ellos se inicie de forma automática.

máquinas internas posee un servidor web que queremos poner en Internet. Recordemos que el Proxy es la única computadora que realmente tiene conexión a Internet porque es quien tiene una IP verdadera (pública). Todas las demás computadoras tienen IPs privadas asignadas por nosotros. Por esa razón, si desde Internet se conectan al puerto 80 de la IP pública del proxy se estarán conectando al proxy en sí y no a la máquina de la red que tiene el servidor web. Lo que tenemos que hacer es crear una regla de tipo: si entra algo por el puerto 80, redirigirlo al puerto 80 de la máquina X. Esto es muy sencillo, y se realiza con una regla como la siguiente:

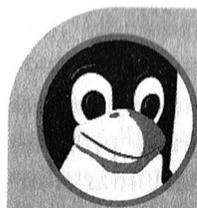
```
iptables -t nat -A PREROUTING -i eth0
-p tcp --dport 80 -j DNAT --to
10.0.0.1:80
```

Bien, nos encontramos ya con algunas cosas nuevas. En primer lugar, el parámetro **-t nat** nos permite gestionar reglas correspondientes a la tabla NAT

(Network Address Translation) del sistema. Esta tabla es la que se encarga de hacer **redirecciones y enmascaramiento de paquetes**. En esa tabla creamos una regla de la cadena **PREROUTING** (antes de despachar) que dice que todo lo que entra por la interfaz **eth0** (interfaz que suponemos está conectada a Internet) por el protocolo TCP y el puerto 80, sea redirigido (acción **DNAT**) hacia la IP 10.0.0.1 puerto 80. ¿Simple, no? Recordemos que si tenemos la conexión a Internet por medio de otro dispositivo que no sea **eth0** debemos cambiar el parámetro **-i**. También tengamos en cuenta que cuando trabajamos con la tabla NAT debemos ejecutar el siguiente comando para que los cambios tengan efecto:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Ahora sí, vamos a analizar el funcionamiento de un **DMZ (Demilitarized Zone)**. Tal como vimos anteriormente, cualquier máquina de nuestra red ahora puede ser un servidor de Internet. El problema está en que al existir una conexión directa entre Internet y una máquina interna de la red, no sólo tenemos que preocuparnos porque el proxy sea



Las DMZ (zonas desmilitarizadas) sólo son una forma estratégica de usar el firewall iptables, pero existen muchas más. De hecho, si investigamos un poco, veremos que hasta podemos inventar nuestras propias formas de montar sistemas de seguridad. Si estamos escasos de ideas, siempre podemos recurrir a Google para encontrar lo que necesitamos.

seguro sino también porque esa máquina-servidor sea segura. Si alguien logra violar la seguridad del servidor, entonces podrá acceder al resto de la red. Entendiendo este problema, la idea básica del DMZ es **aislar al servidor** del resto de la red. ¿Y cuál es la mejor manera? ¡Poniéndolo en otra red! Para entender esto, pongamos nuestra atención en el proxy. Hasta ahora, esta máquina tiene dos placas de red: una conectada a Internet y la otra a la red local en donde están todas las computadoras. La que está conectada a Internet tiene una IP pública, y la que está conectada a la red local tiene una IP definida por nosotros mismos, que es compatible con las IPs de la red local. Bien, para montar un DMZ, necesitamos agregar una tercera placa de red al proxy con una dirección

## III MÁS INFORMACIÓN SOBRE DMZ

El modelo de Demilitarized Zone presentado en este libro es un DMZ sumamente simple, que se puede escalar mucho en cuestiones de funcionalidad y seguridad. Para obtener más información sobre DMZ, podemos visitar el siguiente enlace que contiene un tutorial paso a paso en español, sumamente didáctico y con gráficos explicativos: [www.wikilearning.com/firewall\\_de\\_una\\_lan\\_con\\_salida\\_a\\_internet\\_con\\_dmz-wkccp-9755-6.htm](http://www.wikilearning.com/firewall_de_una_lan_con_salida_a_internet_con_dmz-wkccp-9755-6.htm).

de red nueva, para allí poner el servidor web. Entonces podremos, mediante diferentes reglas de firewall, aislar el paso entre la red del servidor web (que de ahora en más llamaremos DMZ) y la red local. Habiendo explicado esto, simplificaremos el montaje de un DMZ en los siguientes pasos:

1. Agregar una placa de red al proxy y allí conectar el servidor web.
2. Crear reglas de NAT para la red local y para el DMZ.
3. Redireccionar todo el tráfico que entra al puerto 80 del proxy hacia el puerto 80 del servidor web ubicado en el DMZ.
4. Cerrar todo acceso desde el DMZ hacia la LAN.
5. Cerrar todo acceso desde el DMZ hacia el proxy.
6. Cerrar el acceso desde Internet a los puertos que no son el 80 en el proxy.

El secreto de la DMZ son los puntos 4 y 5. Si cerramos todo acceso desde la DMZ hasta la LAN, entonces en el supuesto caso que alguien logre hackear nuestro sistema nunca podrá pasar a la LAN, ya que se encuentra en una red diferente filtrada por el proxy. Ahora veamos cómo traducir la idea explicada anteriormente a reglas de iptables. Para este ejemplo, supondremos que la placa **eth0** está conectada a Internet y tiene la IP 192.168.1.2, la placa **eth1** está conectada a la red local y tiene la IP 192.168.10.1 y la placa **eth2** está conectada a la DMZ y tiene la IP 192.168.3.1. Como vemos, son tres subredes diferentes.

```
# Redirigimos todo lo que entra al puerto 80
iptables -t nat -A PREROUTING -i eth0 -p tcp -dport 80 -j DNAT -to 192.168.3.2:80
-to 192.168.3.2:443

# Permitimos todo acceso local
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -s 192.168.10.0/24 -i eth1 -j ACCEPT

# Hacemos NAT del DMZ y de la LAN
iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o eth0 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 192.168.3.0/24 -o eth0 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward

# Cerramos todo paso entre la DMZ y la LAN
iptables -A FORWARD -s 192.168.3.0/24 -d 192.168.10.0/24 -j DROP

# Cerramos el acceso de la DMZ al propio firewall
iptables -A INPUT -s 192.168.3.0/24 -i eth2 -j DROP

# Ahora cerramos el rango de puertos del 1:1024 en el firewall.
```

```
iptables -A INPUT -s 0.0.0.0/0
-p tcp -dport 1:1024 -j DROP
iptables -A INPUT -s
0.0.0.0/0 -p udp -dport
1:1024 -j DROP
```

Como mencionamos antes, éste es un modelo básico, pero se pueden hacer cosas más complejas como **dobles DMZ**, montar un DMZ por servidor o incluso aplicar reglas de filtrado más estrictas en cuanto a los puertos abiertos del proxy. Lo que debe quedar claro es que el DMZ es un modelo de seguridad que permite **independizar el servidor** que tiene acceso a Internet del resto de la red. Así, con un solo sistema estamos haciendo **triple protección**: por un lado protegemos a la red interna, por otro lado al proxy y por otro lado al servidor. Luego, si lo deseamos, podemos agregar otro firewall en el DMZ y otro firewall en algunos equipos de la red interna. Todo depende del nivel de paranoia que tengamos.

Existen ciertos casos en los que es necesario transmitir información entre la DMZ y la LAN. Para esto, lo que debemos hacer es indicar explícitamente en el proxy que permitimos el paso de un puerto en particular. Es decir, no abrimos todos los puertos sino sólo el que necesitamos. Si abriésemos todos los puertos, no existiría DMZ. Veamos un ejemplo:

```
iptables -A FORWARD -s 192.168.3.1
-d 192.168.10.1 -p tcp --dport 5432
-j ACCEPT
iptables -A FORWARD -s 192.168.10.1
-d 192.168.3.1 -p tcp --sport 5432
-j ACCEPT
```

Debemos poner este código antes de bloquear todas las conexiones entre la DMZ y la LAN. Utilizando el ejemplo original, podemos agregarlo justo después del comando **echo**.

En resumen, lo que hacemos es abrir los puertos en uno y otro sentido para que las dos computadoras puedan comunicarse sin problemas.

## Beagle: un buscador rápido e inteligente

Las herramientas de búsqueda de la consola generalmente son buenas y efectivas, pero no son lo suficientemente rápidas y amigables para los usuarios que necesitan buscar en diferentes fuentes de información. En toda computadora personal, la información contenida es mucha: documentos de texto, archivos PDF, feeds, música y video, conversaciones de chat, etcétera. Buscar esa información se vuelve una tarea tediosa si no contamos con las herramientas adecuadas.

Una de las opciones es utilizar **Beagle** ([http://beagle-project.org/Main\\_Page](http://beagle-project.org/Main_Page)), una gran utilidad para buscar cosas. Se puede buscar en distintos dominios como el sistema de archivos, en los e-mails de Evolution y KMail, en las conversaciones de Gaim y Copete, y también en sus logs, en la caché de Firefox, en Epiphany y Konqueror, en los feeds RSS, en las notas de Tomboy, en los archivos de las suites de oficina OpenOffice, en Abiword, en archivos RTF, PDF, en archivos de ayuda como los Manual Pages, Monodoc, TextInfo y Windows Help (chm), en las imágenes (.JPG, .PNG, .BMP, .TIFF,

GIF), en los archivos de audio .MP3, OGG y .FLAC, en archivos de video y en los cargadores de sus aplicaciones. La lista no termina ahí pero no vamos a nombrar a todos, aunque podemos ver la lista completa en [http://beagle-project.org/Supported\\_Filetypes](http://beagle-project.org/Supported_Filetypes). ¡Hasta podemos crear nuestros propios filtros!

Beagle corre un demonio (**beagled**), que es el encargado de indexar nuestros datos. Por defecto indexa nuestro **\$HOME**, pero podemos especificar otras rutas, por ejemplo, si tenemos nuestra música en otro lugar. También podemos impedir que **beagled** indexe ciertos directorios. Todo esto lo hace de tal manera que no afecta la actividad del usuario en la computadora, aunque podemos especificarle que indexe lo más rápidamente posible (por ejemplo, cuando no vamos a usar la PC), con los siguientes comandos:

```
$ beagle-shutdown
$ export
    BEAGLE_EXERCISE_THE_DOG=1
$ beagled
```

Podríamos utilizar esto cuando, por ejemplo, se activa el protector de pantalla. Para buscar tenemos varias alternativas: la primera es **beagle-search**, una interfaz gráfica programada en **Gtk#**. Los usuarios de KDE pueden estar más a gusto si utilizan **Kerry**. Para los que prefieren la consola, **beagle-query**. Es bue-

no conocer la sintaxis de búsqueda, que es similar a lo que ya conocemos. Si nuestra búsqueda proporciona demasiados resultados, podemos probar con la sintaxis **standard** si ponemos asteriscos, frases entre comillas, el signo menos (-) para quitar resultados que contengan una determinada palabra, palabras clave como **OR** (es case sensitive) para palabras opcionales. También hay otras opciones que se pueden tener en cuenta, como las propiedades de los ítems indexados. Para buscar ítems con una determinada fecha o en un rango, se usa **date:2007** (año 2007) o **date:200612-2007** (entre diciembre de 2006 y 2007, inclusive).

Beagle indexa los metadatos de los documentos. ¿Cómo buscar ese .PDF que alguna vez vimos y recordamos que el autor es Pepe? Con el comando: **ext:pdf creator:Pepe**. También podríamos buscar e-mails que hayan sido enviados desde una determinada dirección de correo. Beagle soporta varios plug-in para realizar búsquedas. Por ejemplo, para Evolution, Gaim, Kopete, etcétera. Estos plug-in pueden proporcionar nuevas palabras clave para realizar búsquedas. Con el comando **beagle-query -keywords** podemos encontrar otras para realizar búsquedas más detalladas.

Finalmente, podemos buscar en determinadas fuentes. Por ejemplo, con la palabra clave **type:IMLog** buscaremos en los logs de chat, o con **filetype:image** sólo las imágenes.

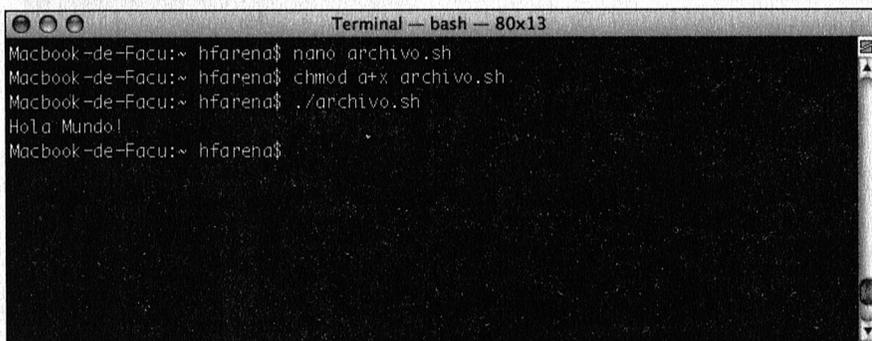
- 1 Es posible monitorear el uso de CPU y de memoria de las aplicaciones con el comando **ps aucwx**.
- 2 Los servicios que más recursos del sistema consumen son los que manejan grandes cantidades de datos, como las bases de datos MySQL y PostgreSQL, el sistema de búsqueda Beagle, BIND, etcétera. Si no los utilizamos, podemos deshabilitarlos.
- 3 En sistemas que no usan interfaz gráfica podemos utilizar el navegador web **Links** (<http://links.sourceforge.net>), que permite acceder a una gran cantidad de sitios web visualizando sólo su texto.
- 4 Automatrix es otra forma de agilizar la instalación de aplicaciones y plug-in del sistema. Para más información podemos visitar [www.getautomatrix.com](http://www.getautomatrix.com).
- 5 Nessus (y sus derivados libres) es una excelente herramienta para auditar la seguridad de un sistema. Realiza un rastreo de puertos de uno o más equipos y genera un informe de las vulnerabilidades con algunas sugerencias de cómo solucionarlas. Más información en [www.nessus.org](http://www.nessus.org).
- 6 Para acceder a la tabla NAT (*Network Address Table* de iptables) se utiliza el parámetro **-t nat**.
- 7 Para redireccionar un puerto hacia otro de otro equipo podemos usar el comando **iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to 10.0.0.1:80**, donde **--dport 80** es el puerto de entrada al sistema y **10.0.0.1:80** es la IP y el puerto del equipo remoto.
- 8 Siempre que generemos scripts de firewalls debemos guardarlos en **/usr/sbin** con permisos de ejecución y modificación sólo para el administrador.
- 9 Si tenemos una distribución Mandriva, Fedora o Red Hat podemos usar el archivo **/etc/rc.d/rc.local** para agregar un script de firewall allí y lograr que se ejecute cada vez que se inicie el sistema.
- 10 Una excelente herramienta para monitorear el funcionamiento del firewall es **IPTraf**, que se puede descargar de <http://iptraf.seul.org>.

# Scripts en bash para aumentar la productividad

Los scripts son pequeños programas que nos ayudan a automatizar procesos y así aumentar la productividad. Este capítulo incluye un mini tutorial de programación en bash para aquellos que aún no han comenzado a dar sus primeros pasos y luego una selección de los mejores trucos para aprovechar al máximo la potencia de este lenguaje.

- Programar en la consola
- Tutorial de bash en 10 minutos
- Cómo crear nuestras propias funciones en bash
- Evaluación de expresiones aritméticas en bash
- Generación de números aleatorios
- Múltiples descargas de BitTorrent desde la consola
- Script para disparar backups de datafiles
- Creación de archivos HTML
- Versión alternativa del comando sleep
- La magia de las llaves
- Función para intercambiar dos valores en una variable
- Varias formas de leer la primera línea de un archivo
- Función para centrar el texto en pantalla
- Utilizar grep dentro de scripts de Bash
- Bonus tips

Mucha gente teme con fervor utilizar a consola de texto, pero lo cierto es que es muy útil para algunas tareas administrativas del sistema. A medida que nos vamos acostumbrando a la consola, descubrimos que son más y más los comandos que tipeamos y el trabajo se puede llegar a volver cada vez más tedioso. Por eso existen los **scripts**. Básicamente, éstos son pequeños programas (archivos de texto con permisos de ejecución, para ser más exactos) que nos permiten **automatizar tareas** por medio de la programación. De alguna manera, se podrían comparar con los archivos **.BAT** de DOS, aunque son algo parecidos pero mucho más completos. Aunque quizás pensemos que la programación no es para nosotros, en realidad no es muy complejo porque no estaremos desarrollando grandes programas sino pequeñas utilidades para facilitar nuestro día a día junto a la administración del sistema. Para aprender a hacer scripts en bash, lo único que necesitamos es un editor de texto y un poco de ganas de aprender. ¡Manos a la obra!



```
Terminal — bash — 80x13
Macbook-de-Facu:~ hfarend$ nano archivo.sh
Macbook-de-Facu:~ hfarend$ chmod a+x archivo.sh
Macbook-de-Facu:~ hfarend$ ./archivo.sh
Hola Mundo!
Macbook-de-Facu:~ hfarend$
```

Figura 1. Salida de nuestro primer script en bash.

Como ya mencionamos, los scripts no son más que **archivos de texto interpretados**. Por esta razón, vamos a iniciar un editor de texto (Nano, Vi, Emacs o el que más nos guste) y crearemos un archivo llamado **archivo.sh** (la extensión **.SH** nos servirá de ahora en más para identificar a todos los scripts hechos en bash). En él, ingresaremos las siguientes líneas:

```
#!/bin/sh
# Ejemplo 1: Hola Mundo!
echo "Hola Mundo!"
```

Una vez que terminamos de tipear esas líneas debemos guardar el archivo y salir del editor de texto. Luego le otorgaremos permisos de ejecución con el siguiente comando:

**chmod a+x archivo.sh**

Una vez concluido el proceso de desarrollo de esta pequeña aplicación, vamos a ejecutarla con el comando **./hello**. Éste devolverá en pantalla el mensaje que podemos ver en la **Figura 1**.

Este pequeño script consta de tres líneas. La primera indica que es un script de bash y debe ser interpretado por éste. La segunda sólo es un comentario (en bash, como en muchos otros lenguajes de script, los comentarios comienzan con el símbolo #). Por último, la tercera línea es el comando **echo** con un valor de cadena (el mensaje **Hola Mundo!**) encerrado entre comillas. Como podemos deducir, el comando **echo** imprime la cadena en la terminal activa. El funcionamiento de este sistema es sencillo y vamos a adentrarnos un poco más en su desarrollo.

Lo primero que debemos aprender ahora es a manejar **variables**. Las variables nos sirven para **almacenar datos en la memoria** (como números, nombres de archivos o lo que fuera) para luego poder utilizarlos como datos en cualquier parte del programa. La definición de variables en el lenguaje de script de bash es igual que la definición de variables de entorno. Si queremos definir la variable **edad** con el valor **24** sólo hay que ingresar:

```
edad=24
```

Como vemos, no se utilizan espacios entre el nombre de la variable y su corres-

Salvo que hayan sido definidas con anterioridad a la ejecución del script, las variables existirán en el sistema mientras el script esté en ejecución. Cuando el script finalice, las variables se borrarán.

pondiente valor. Es muy importante que respetemos esta propiedad porque si no bash tomará cada cadena (**edad**, el símbolo = y el valor) como un comando diferente. Ahora, si queremos imprimir la variable en pantalla, tenemos que utilizar la misma función **echo** pero debemos pasarle como parámetro el nombre de la variable precedido por el símbolo **\$**.

```
echo $edad
```

También es posible incluir variables en medio de una cadena, y se las llama de la misma manera. Por ejemplo:

```
echo "La edad de este usuario es:
$edad años"
```

De esta manera, se imprimirán todos los caracteres encerrados entre comi-

## ¿DÓNDE ES CONVENIENTE GUARDAR LOS SCRIPTS?

A la hora de definir la ubicación de almacenamiento de los scripts, es conveniente seguir respetando la estructura del sistema de archivos de Linux: aquellos scripts que sólo deban ser utilizados por el administrador deben ser almacenados en el directorio **/usr/sbin**; mientras que el resto debe ser almacenado en el directorio **/usr/bin**.

llas y, cuando llegue a **\$edad** se imprimirá el valor de esta variable. Desde ya que podemos utilizar las variables con cualquier comando de Linux. Veamos el siguiente ejemplo:

```
#!/bin/sh
archivo=/etc/lilo.conf
cat $archivo
```

Lo que estamos haciendo en este ejemplo es pasar como parámetro del comando **cat** la variable **\$archivo**, cuyo contenido es la ruta al archivo **lilo.conf**. También podemos guardar en una variable la salida de un comando. Veamos el siguiente ejemplo:

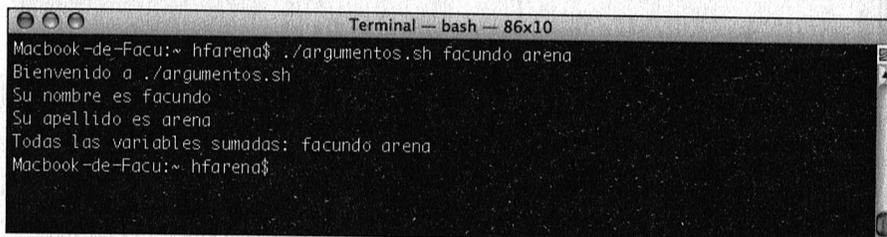
```
#!/bin/sh
variable=`ls`
echo $variable
```

Cuando ponemos el nombre de un comando entre comillas simples invertidas como valor de una variable, lo que estamos haciendo es que bash ejecute ese comando y guarde la salida en la variable definida. Luego, tal cual lo muestra el ejemplo, podemos imprimir esos datos con la función **echo** que aprendimos antes. Cuanto más juguemos y probe-

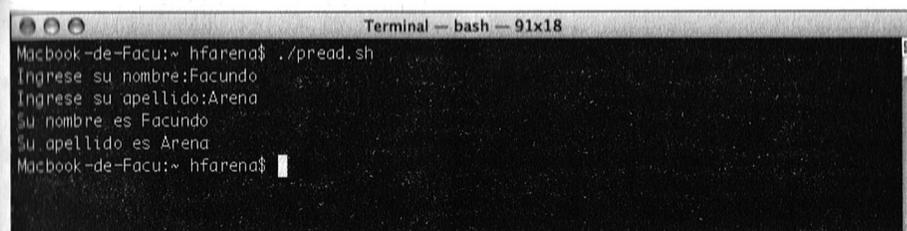
mos con las variables, más funcionalidades nuevas encontraremos. Uno de los recursos más utilizados en la programación de scripts en bash es el **pasaje de parámetros** desde la línea de comandos. Para esto, bash provee una serie de nombres de variables predefinidas. La variable **\$0** almacena el nombre del script, y las variables siguientes (**\$1**, **\$2**, etcétera) guardan los diferentes parámetros. El siguiente ejemplo nos permitirá aclarar estos conceptos:

```
#!/bin/sh
# archivo argumentos
# Modo de uso ./argumentos
  [nombre] [apellido]
echo "Bienvenido a $0"
echo "Su nombre es $1"
echo "Su apellido es $2"
echo "Todas las variables
sumadas: $*"
```

Si ejecutamos: **./argumentos.sh facundo arena** el programa devolverá lo que podemos ver en la **Figura 2**. Como seguramente habremos adivinado, la variable **\$\*** guarda todos los argumentos concatenados. Este método nos será útil para la creación de scripts con múltiples opciones, configuradas desde la línea de comandos en el momento de la ejecución.



**Figura 2.** Pasaje de argumentos a un script de bash.



**Figura 3.** Utilización de la función read para recibir datos del usuario por medio del teclado.

Bien, es hora de que el usuario del programa ingrese algunos datos por medio del teclado. Este proceso es muy sencillo: bash utiliza el comando **read** para el ingreso de datos en variables. Su formato es:

**read [variable]**

Cuando la interpretación del script llegue a este punto, esperará un ingreso de datos proveniente del teclado y lo irá almacenando en la variable indicada. Ahora desarrollaremos el mismo ejemplo de la sección anterior pero utilizando esta función.

```
#!/bin/sh
# Archivo pread
# Uso ./pread

echo -n "Ingrese su nombre:"
read nombre
echo -n "Ingrese su apellido:"
read apellido
echo "Su nombre es $nombre"
echo "Su apellido es $apellido"
```

Prestemos atención al parámetro **-n** en la función **echo**. Éste hace que luego de la impresión del mensaje, **echo** no baje de línea sino que se mantenga en ésta. Para

### OTRAS VARIABLES

Otras variables predefinidas con respecto a los parámetros son:

**\$#:** devuelve la cantidad de parámetros pasados a la función.

**\$?:** contiene el valor devuelto del último programa ejecutado dentro del script.

entender este concepto, podemos probar el ejemplo sin los parámetros **-n** en las funciones **echo** y veremos la diferencia. Ahora vamos a ver algunas estructuras que nos permitirán controlar el flujo de nuestros programas. Las **estructuras condicionales** son lo que hacen interesantes a los scripts. Si las utilizamos es posible crear menús con **opciones**, tomar diferentes **decisiones** según una situación específica, hacer **preguntas**, y un sinnúmero de acciones más. La más famosa y utilizada de estas estructuras es el **if** condicional. Su forma básica es la siguiente:

```
if [ expresion ]
then
  accion1
elif [ expresion ]
then
  accion2
```

```
else
    accion3
fi
```

La primera línea evalúa la expresión entre corchetes. Si es verdadera ejecuta **accion1** y si es falsa, evalúa la expresión de **elif** para saber si corresponde ejecutar **accion2**. Si la expresión de **elif** no es verdadera, salta a la sentencia **else** y ejecuta **accion3**. Luego se cierra la estructura con un **fi**. Recordemos separar los corchetes de los valores de la expresión a evaluar y de su operador, como vemos a continuación:

```
echo "El valor ingresado es
diferente a 20"
fi
```

**Elif** es la contracción de **else if** y su funcionamiento es el mismo. Lo que hace este script es pedirle al usuario que ingrese un número. Luego, evalúa si ese número es igual a 20 o diferente. Para cada caso, muestra un correspondiente mensaje (**Figura 4**). Veamos un ejemplo utilizando los evaluadores que vimos anteriormente:

```
#!/bin/sh
echo -n "Ingrese un valor
numérico:"
read valor
if [ $valor = 20 ]
then
echo "El valor es igual a 20"
else
```

```
#!/bin/sh

echo -n "Ingrese un nombre de
archivo o directorio:"
read valor
if [ -d $valor ]
then
echo "Usted ingreso el nombre de
un directorio"
elif [ -f $valor ]
```

```
Terminal -- bash -- 91x18
Macbook-de-Facu:~ hfarena$ ./juego.sh
Ingrese un valor numerico:40
El valor ingresado es diferente a 20
Macbook-de-Facu:~ hfarena$
```

**Figura 4.** Salida de nuestro primer script con una estructura condicional.

```
Terminal -- bash -- 91x18
Macbook-de-Facu:~ hfarena$ ./juego.sh
Ingrese un nombre de archivo o directorio:bash
Usted ingreso cualquier cosa!
Macbook-de-Facu:~ hfarena$ ./juego.sh
Ingrese un nombre de archivo o directorio:juego.sh
Usted ingreso el nombre de un archivo
Macbook-de-Facu:~ hfarena$ ./juego.sh
Ingrese un nombre de archivo o directorio:Music
Usted ingreso el nombre de un directorio
Macbook-de-Facu:~ hfarena$
```

**Figura 5.** Ejemplo de utilización de la estructura **if** para determinar si el dato ingresado por el usuario es un archivo o un directorio.

```
then
echo "Usted ingreso el nombre de
un archivo"
else
echo "Usted ingreso cualquier
cosa!"
fi
```

Este programa tiene la posibilidad de identificar si lo que el usuario ingresó es el nombre de un archivo o de un directorio por medio de los operadores **-d** y **-f** que vimos antes (**Figura 5**). Si el usuario ingresó una cadena que no corresponde al nombre de un archivo o de un directorio, devuelve el mensaje de respuesta **Usted ingreso cualquier cosa!**. Otra estructura condicional es el **case**. Su forma básica es la siguiente:

```
case [variable] in
opcion1)
funcion1 (...);;
opcion2)
funcion2 (...);;
opcion3)
funcion2 (...);;
*)
funcion4 (...);;
esac
```

La estructura **case** evalúa la variable [**variable**] con las distintas opciones definidas por el usuario. Si se cumple alguna, se ejecutan las funciones correspondientes a esta opción. También se puede usar el símbolo **|** que funciona como un **0**, con lo que ambas opciones pueden ser válidas (no es necesario utilizar la segunda opción). Cada opción finaliza con un doble símbolo **;** (**::**). Si no se cumple ninguna de las opciones, se ejecutará (siempre y cuando esté definida), la opción **\***. Por último, para cerrar la estructura se utiliza **esac**. Veamos un ejemplo:

```
#!/bin/bash

echo "1. Ejecutar el comando ls"
echo "2. Ejecutar el comando
free"
echo "3. Crear un nuevo
directorio"
echo "4. Crear un nuevo usuario"
echo "5. Salir"
read -p "Ingrese una opción: "
OPCION
case $OPCION in
1) ls ;;
2) free ;;
3) echo -n"Nombre del
directorio: "
```

### III ¿BASH O SH?

Cuando busquemos scripts por Internet, veremos que algunos de ellos comienzan con una línea que dice **/bin/bash** (en referencia a la ruta en donde se encuentra el intérprete de comandos), mientras que otros dicen **/bin/sh**. En los sistemas operativos Linux, ambos scripts deberían funcionar sin problemas. Bash es una versión extendida y totalmente compatible con sh.

```

read DIRECTORIO
mkdir $DIRECTORIO ;;
4) echo -n "Nombre del usuario
a crear: "
read NOMBREUSUARIO
useradd $NOMBREUSUARIO ;;
5) exit;;
*) echo "Usted ingreso una
opcion que no esta en
el menu"
exit 1;;
esac
exit 0

```

Este ejemplo le presenta un menú de opciones al usuario que permite ejecutar diferentes comandos típicos de GNU/Linux (Figura 6).

Ya casi tenemos todo. Ahora sólo nos resta aprender a crear **funciones repetitivas**. Diremos que los **bucles** repetitivos son estructuras que nos permiten repetir una acción muchas veces. Bash soporta los bucles repetitivos **while** (repetición condicional) y **for**. El funcionamiento es el mismo que en los otros lenguajes de script y programación. La estructura del ciclo **while** es la siguiente:

```

while [condicion]
do
accion1

```



Utilizar menús en los scripts significa imposibilitarlos para el trabajo en conjunto con otros scripts en segundo plano, utilizando tuberías por ejemplo, por la simple razón de que un script con menú requiere de la intervención del usuario para funcionar. Si necesitamos que un script tenga la posibilidad de cooperar con otros, es mejor utilizar un sistema de parámetros.

```

(...)
done

```

Si la condición se cumple, **while** ejecuta **accion1** y todas las operaciones siguientes hasta que la condición sea falsa. Un ejemplo sencillo de **while** es el siguiente:

```

#!/bin/sh

opcion=1 # Ponemos opción en 1
para que el ciclo se cumpla
la primer vez.
while [ $opcion != 0 ]
do
echo "opcion= $opcion"
echo -n "Ingrese un numero [0
para salir]: "
read opcion
done # esto finaliza el
ciclo while

```

```

Macbook-de-Facu:~ hfarena$ ./juego.sh
1. Ejecutar el comando ls
2. Ejecutar el comando free
3. Crear un nuevo directorio
4. Crear un nuevo usuario
5. Salir
Ingrese una opcion:

```

Figura 6. Menú de opciones creado con la estructura case.

Este programa entra en un ciclo en el que pide el ingreso de un valor infinitamente hasta que se ingrese el 0 (Figura 7). Nótese que en la condición del **while** los valores están separados de los corchetes y de los operadores.

Otro ciclo repetitivo muy utilizado en la programación es el ciclo **for**. Éste permite ejecutar un bloque de funciones una determinada cantidad de veces. Esta cantidad puede estar definida por el contenido de una variable. El funcionamiento de **for** en bash puede parecer raro y complejo al comienzo, pero veamos el siguiente ejemplo para comenzar a conocerlo:

```

#!/bin/sh
# Ejemplo 5: uso del for.

valores='ls`
for x in $valores
do
echo "El valor de x es: $x"
done

```

La variable **valores** contiene la salida del comando **ls** (listado de archivos). Esto se logra asignándole a la variable el nombre del comando entre comillas invertidas. El **for** almacena en la variable **x** el valor que está siendo ciclado.

```

Macbook-de-Facu:~ hfarena$ ./juego.sh
opcion= 1
Ingrese un numero [0 para salir]: 2
opcion= 2
Ingrese un numero [0 para salir]: 5
opcion= 5
Ingrese un numero [0 para salir]: 2
opcion= 2
Ingrese un numero [0 para salir]: 7
opcion= 7
Ingrese un numero [0 para salir]: 3
opcion= 3
Ingrese un numero [0 para salir]: 0
Macbook-de-Facu:~ hfarena$

```

Figura 7. Utilización de la estructura while para crear un ciclo infinito.

```

Macbook-de-Facu:~ hfarena$ ./juego.sh
El valor de x es: Incomplete
El valor de x es: LINTIME
El valor de x es: Library
El valor de x es: Movies
El valor de x es: Music
El valor de x es: Notas
El valor de x es: LU35
El valor de x es: Pictures
El valor de x es: Produccion
El valor de x es: Alternaria
El valor de x es: TV
El valor de x es: 1
El valor de x es: Public
El valor de x es: ROMS
El valor de x es: Sites
El valor de x es: capacitacion-rioja.rtf
El valor de x es: himno-jairo.mp3
El valor de x es: himno_01.mp3
El valor de x es: juego.sh
El valor de x es: path
El valor de x es: videos

```

Figura 8. Utilización de la estructura for para procesar todos los ítems del directorio actual.

dentro de la variable **\$valores**. **For** realiza un ciclo por cada valor dentro de la variable **\$valores** (en el ejemplo, uno por cada archivo). Por lo tanto, si tenemos 10 archivos en un directorio, **for** realizará 10 repeticiones del código. ¡Y con esto acabamos de aprender cómo procesar muchos archivos en un mismo directorio! (**Figura 8**). También es posible utilizar **for** del siguiente modo:

```
#!/bin/sh

for x in 1 2 3 4 5
do
echo "el valor de x es: $x"
done
```

De esta forma, **for** realizará cinco ciclos del código especificado entre las sentencias **do** y **done**. Podemos terminar un ciclo repetitivo con la función **break** o el programa completo con la función **exit**. Los usuarios también pueden terminar el programa en cualquier momento con la combinación de las teclas **CTRL + C**. Esta combinación matará el proceso sin importar en qué estado se encuentre,



Como en muchos otros casos, tenemos alternativas para realizar las mismas cosas. Existen dos formas de listar los directorios sin tener que usar siempre el comando **ls**.

Una de ellas se realiza con la función interna **printf** del siguiente modo: **printf "%s\n" \***. La otra, en cambio, implica utilizar el comando **echo** de la siguiente manera: **echo \***.

por lo que es recomendable no invitar al usuario a utilizarla si no está completamente seguro de lo que hace.

## Cómo crear nuestras propias funciones en bash

Podemos crear nuestras propias funciones en bash. Esto es muy útil cuando tenemos bloques de acciones a realizar que podemos necesitar en diferentes etapas del funcionamiento del programa. Por ejemplo, podemos tener una función que se llame **borrar\_archivos** que, si la utilizamos en cualquier momento del script, nos permite de manera automática borrar los archivos que se le pasan como parámetros. Ésta es su forma básica:

```
mi_funcion() {
    sentencias(...)
}
```

Cabe destacar que estas funciones personalizadas deben ser incluidas al principio del código y no al final como en otros lenguajes. Si son varias, se ponen una debajo de la otra. Si queremos realizar un llamado a la función, solamente se incluye el nombre y los parámetros del siguiente modo:

**mi\_funcion param1 param2 (...)**

Dentro de la función creada, los parámetros se llaman igual que en todos los scripts bash (**\$1**, **\$2**, etcétera). Veamos un ejemplo de esto, cuya salida podemos ver en la **Figura 9**:

```
#!/bin/sh
```

```
imprimir() {
    echo -n "Este mensaje es
    devuelto por la función
    imprimir():"
    echo $1;
}

echo -n "Ingrese una palabra
que se usara como parámetro
para imprimir():"
read palabra
imprimir $palabra;
```

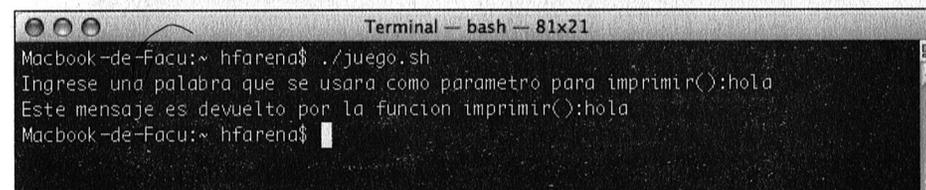
## Evaluación de expresiones aritméticas en bash

Si queremos evaluar una expresión aritmética tenemos dos posibilidades. La primera es escribiéndola entre corchetes y anteponiéndole el símbolo pesos (**\$**). Un ejemplo sería

**NUMERO=\$((25\*4+2))**

También es posible utilizar variables dentro de la ecuación.

Éstas podrán ser especificadas con o sin el símbolo pesos. Por ese motivo, **NUMERO2=\$((NUMERO-2))** es lo mismo que **NUMERO2=\$((NUMERO-2))**.



**Figura 9.** Utilización de las funciones creadas por el usuario en scripts de bash.

## USAR FUNCIONES

Las funciones no sólo sirven para estructurar internamente un script para que sea legible, sino que también sirven para aliviar el trabajo en los futuros scripts. Con tan sólo copiar y pegar una función de un script a otro, tendremos la funcionalidad sin necesidad de volver a escribir todo el código.

La segunda posibilidad es utilizar la función interna **let**. Para ello, siempre debemos asignarle el resultado a una variable. Su utilización se muestra en el siguiente ejemplo:

**let X=\$((X+1))**

El funcionamiento es bastante intuitivo: se resuelve la ecuación que se encuentre a la derecha del signo igual (=) y el resultado se le asigna a la variable indicada a la izquierda de éste. Como se ve en el ejemplo, siempre es necesario asignarle el resultado a una variable. Además, cabe destacar que siempre que se utilicen variables dentro de la ecuación, se las deberá utilizar con el signo pesos (**\$**). Veamos otro ejemplo:

```
#!/bin/sh
C=0
```

```
while read LINEA; do
  echo $LINEA | grep $1 >
    /dev/null
  if [ $? -eq 0 ]; then
    let C=$C+1
  fi
done
echo La palabra $1 se encontro
  $C veces
```

Este programa contará la cantidad de apariciones de una palabra especificada por parámetro en la entrada estándar (Figura 10). Una forma de utilizarlo sería:

```
cat carta.txt | ./contar_palabra hola
```

En lugar de utilizar una condición, en el ciclo **while** especificamos que ejecute el comando **read**. De esta forma, el **while** se ejecutará mientras haya líneas para leer desde **STDIN** (*Standard Input*, Entrada Estándar). Luego enviará la línea leída al comando **grep** al cual se le pasará como parámetro el mismo que a nuestro script (la palabra a buscar). También enviaremos la salida al archivo **/dev/null** para que la salida de **grep** no se vea por pantalla.

Por último, aparece la variable **?**. Esta variable contiene el valor con el que salió el último comando ejecutado (**grep** en nuestro ejemplo). Por lo general, si

```
Terminal — bash — 81x21
Macbook-de-Facu:~ hfarena$ cat carta.txt | ./juego.sh hola
La palabra hola se encontro 1 veces
Macbook-de-Facu:~ hfarena$
```

Figura 10. Podemos contar la cantidad de veces que aparece la palabra hola en el archivo carta.txt.

esta variable contiene el valor **0** es porque el comando se ejecutó correctamente y terminó de forma satisfactoria (en el caso de **grep** significa que encontró una línea que concuerda). Los valores distintos de **0** especificarán la condición por la cual no se terminó satisfactoriamente. Estos valores dependerán del programa que se ejecute, por lo que se deberá consultar el manual para ello.

## Generación de números aleatorios

Este script es un **juego** en donde el usuario debe adivinar un número elegido al azar por la máquina (Figura 11).

```
#!/bin/bash

AZAR=$((RANDOM%10+1))
while [ 1 ]
do
  echo -n "Ingrese un numero: "
  read NRO
  if [ "$NRO" -eq "$AZAR" ]; then
    echo "Adivino!"
    break
  elif [ "$NRO" -gt "$AZAR" ]; then
    echo "No no, es mas grande"
  else
    echo "No no, es mas chico"
  fi
done
```

```
echo "No no, es mas grande"
fi
done
```

Para ello, generamos una variable con un número al azar entre 1 y 10 (la variable **\$RANDOM** la utiliza **bash** para darnos un número entero de forma aleatoria). Luego haremos un ciclo infinito ya que **1** siempre será verdadero. Después se pedirá un número y se lo comparará con el número elegido al azar. Si el usuario lo adivina, se le informa y se ejecuta **break**. Sólo en esta ocasión el usuario podrá salir del programa ya que si no seguirá ciclando indefinidamente porque la condición es siempre verdadera. Las otras dos posibilidades sólo le mostrarán un indicio del número que tiene que adivinar y volverá a pedir otro número.

## Múltiples descargas de BitTorrent desde la consola

Se puede tener un equipo en la red dedicado casi exclusivamente a bajar archivos por medio de la red BitTorrent. De esa forma, se pueden utilizar cinco o diez torrents en simultáneo, incluso en

```
Terminal — bash — 81x21
Macbook-de-Facu:~ hfarena$ ./juego.sh
Ingrese un numero: 2
No no, es mas grande
Ingrese un numero: 3
No no, es mas grande
Ingrese un numero: 6
No no, es mas chico
Ingrese un numero: 5
No no, es mas chico
Ingrese un numero: 4
Adivino!
Macbook-de-Facu:~ hfarena$
```

Figura 11. Un pequeño juego creado a partir de números aleatorios.

## REDIRECCIONAR MENSAJES

Es posible redireccionar los mensajes de error de los comandos que se ejecutan dentro del script hacia **archivos de log** especialmente creados para almacenarlos. Esto lo podemos hacer con el símbolo **2>** y el nombre del archivo que almacenará los mensajes de error al final del comando. Por ejemplo: **cp \*.txt 2> errores.txt**.

un equipo que no tenga monitor: sólo controlándolo por **SSH**. Cuando se debe elegir un cliente de BitTorrent, se puede optar por instalar el estándar para GNU/Linux (que en Debian y sus derivados se instala con el comando **apt-get install bittorrent**). El paquete estándar incluye varios clientes de BitTorrent para la consola, pero uno en particular parece el ideal para el método de computadora **sin cabeza**, conectada sólo a la red y a la electricidad. Ése es **btdownloadheadless** y es un cliente súper básico que no muestra ninguna interfaz. Simplemente nos informa en pantalla el progreso de la descarga. Entonces, si armamos este simple script de Bash, con sólo escribirlo comenzará la descarga de todos los archivos **.TORRENT** que existan en el directorio:

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

```
for archivo in $(ls *.torrent);  
do  
nohup btdownloadheadless  
$archivo &  
done
```

El comando **nohup** da la posibilidad de que **btdownloadheadless** siga funcionando aun cuando nos desconectemos de la terminal. Para finalizar la descarga (o interrumpirla más adelante) habrá que utilizar el comando **kill** seguido del número de proceso correspondiente al **btdownloadheadless** del torrent que deseamos detener.

Si se quiere seguir de cerca el progreso, sólo hay que visualizar las últimas diez líneas del archivo **nohup.out**. Es recomendable borrar ese archivo cada cierta cantidad de tiempo por el crecimiento que tiene.

## Script para disparar backups de datafiles

Este script ha sido desarrollado para reemplazar la funcionalidad que daba un programa compilado que disparaba **tareas concurrentes** y cuya licencia caducó. El script está pensado para ser utilizado como una librería de un script de mayor envergadura, como por ejemplo backup y los distintos tablespaces de una base de datos como puede ser Oracle, DB2, Sybase, o alguna otra tarea que se pueda desmenuzar en unidades de trabajo paralelas. Actualmente,

es estable y está siendo utilizado en servidores productivos UNIX sin ningún inconveniente. El código fuente está comentado para una mejor comprensión. Asimismo, el script genera logs en **/tmp** para su análisis. Finalmente, el script demo que trae simula la ejecución de procesos de distinta duración para que, mediante **pgrep/top/ps -ef|grep** y los logs, se pueda ver cómo administra los procesos.

```
#####
```

```
#####
```

```
# Librerías de bajo nivel
```

```
# By Pablo Niklas - Bajo
```

```
licencia GPL
```

```
#####
```

```
#####
```

```
# paralelo(): Administrador  
de procesos paralelos
```

```
# By Pablo Niklas
```

```
(pablo.niklas@gmail.com)
```

```
# Bajo Licencia GPL
```

```
#
```

```
# Uso: paralelo [<lista de  
procesos a disparar en forma  
concurrente>]
```

```
# Log de la corrida en  
/tmp/corrida_paralela.PID  
($$).<dia>.log
```

```
function paralelo() {
```

```
# PARALELO:
```

```
# Cantidad de procesos en  
paralelo.
```

```
# Depende de el SO y/o la  
arquitectura. Puede ser:
```

```
# 1) Cant. de cpus + 1 (x86)  
# 2) Cant. de cpus * 2 (x86)  
# 3) Cant. de cpus (SPARC)  
if [ -z "$PARALELO" ]; then  
if [ "`uname -s`"="SunOS" ];  
then
```

```
# Para SunOS...
```

```
if [ "`uname -r`" = "5.6" ];  
then
```

```
PARALELO=`/usr/platform/  
uname -m`/sbin/prtdiag  
-v|grep "US-"|wc -l` #  
Solaris 2.6
```

```
else
```

```
if [ "`uname -r`" = "5.9"  
]; then
```

```
PARALELO=`usr/  
platform/`uname -m`/  
sbin/prtdiag -v|grep  
^CPU|wc -l` #  
Solaris 9
```

```
else
```

```
PARALELO=1 # Default
```

```
fi
```

```
fi
```

```
else
```

```
PARALELO=$((`cat /proc/cpuinfo  
|grep ^proces|wc -l`*2))
```

```
# Linux
```

```
fi
```

```
fi
```

```
# DIRLOG:
```

```
# Directorio donde se depositan  
los logs temporarios de cada  
hilo ejecutado.
```

```
DIRLOG=/tmp
```

```
LOGCPU=$DIRLOG/control_corrida.$$
```

```
date +%d'`.log
```

```
# COMIENZO DEL ALGORITMO
```

```
echo ":::: INICIO CORRIDA  
(`date +%d/%m/%Y -  
%H:%M:%S`)" >> $LOGCPU
```

```
echo ":::: Se correran $  
procesos en total ($PARALELO  
en forma concurrente)." >>  
$LOGCPU
```

```
# Inicializo variables del  
sistema
```

```
I=0;SEQ=""
```

```
while [ $I -le $((PARALELO-1)) ]  
; do
```

```
PID[$I]=0
```

```
SEQ=$SEQ+$I " # Como no  
tengo seq, lo genero :)
```

```
I=$((I+1))
```

```
done
```

```
SEQ=`echo $SEQ|sed 's/+/g'` #  
Depuro los "+"
```

```
TERMINO=false
```

```
JOBLOGTMP="job.$$.`date +%d'`"
```

```
TAREA=0
```

```
while [ $# != 0 ] || !
```

```
$TERMINO; do
```

```
A=0
```

```
for A in $SEQ; do
```

```
# Asigno procesos si  
tengo lugar.
```

```
if [ ${PID[$A]} -eq 0 ] && [  
$# != 0 ]; then
```

```
TAREA=$((TAREA+1))
```

```
echo ":::: Job #TAREA -
```

```
Thread #A - `date
```

```
+`date +%d/%m/%Y - %H:%M:%S` -
```

```
INICIADO." >>
```

```

$DIRLOG/$JOBLOGTMP.`printf %.3d
$TAREA`.log
#echo ": COMIENZO detalle
del Job." >>
$DIRLOG/$JOBLOGTMP.
`printf %.3d $TAREA`.log
#echo $1 >>
$DIRLOG/$JOBLOGTMP.
`printf %.3d $TAREA`.log
#echo ": FIN detalle del
Job." >>
$DIRLOG/$JOBLOGTMP.
`printf %.3d $TAREA`.log
#echo ": COMIENZO salida del
Job." >>
$DIRLOG/$JOBLOGTMP.`printf
%.3d $TAREA`.log
$1 1>>
$DIRLOG/$JOBLOGTMP.`printf
%.3d $TAREA`.log 2>&1 &
PID[$A]=$!
TID[$!]=$TAREA

echo ":::: Job #$TAREA -
Thread #$A - `date
+`%d/%m/%Y - %H:%M:%S` -
INICIADO." >> $LOGCPU

shift
fi
A=$((A+1))
done

# Ciclo de control de
finalizacion de cada
hilo.
TERMINO=true
A=0
for A in $SEQ; do

```

```

# Los distintos *nix
manejan los
procesos a su
manera. :)
FINALIZO=false
if [ "`uname -s`" = "SunOS" ]
&& [ ${PID[$A]} -gt 0 ];
then
[ -z "`ps -p ${PID[$A]}|grep
v " PID TTY TIME
CMD`" ] && FINALIZO=true
fi

if [ "`uname -s`" = "Linux" ]
&& [ ${PID[$A]} -gt 0 ];
then
[ -z "`ps -no-heading -pid
${PID[$A]}`" ] &&
FINALIZO=true
fi

if $FINALIZO ; then
#echo ": FIN salida del
Job." >>
$DIRLOG/$JOBLOGTMP.`printf
%.3d ${TID[${PID[$A}]}`.log
echo ":::: Job
#${TID[${PID[$A}]} - Thread
#$A - `date +%d/%m/%Y -
%H:%M:%S` - FINALIZADO." >>
$DIRLOG/$JOBLOGTMP.`printf
%.3d ${TID[${PID[$A}]}`.log
echo >>
$DIRLOG/$JOBLOGTMP.`printf
%.3d ${TID[${PID[$A}]}`.log
echo ":::: Job
#${TID[${PID[$A}]} - Thread
#$A - `date +%d/%m/%Y -
%H:%M:%S` - FINALIZADO."
>> $LOGCPU

```

```

cat
$DIRLOG/$JOBLOGTMP.
`printf %.3d
${TID[${PID[$A}]}`.log
echo >>
$DIRLOG/$JOBLOGTMP.
`printf %.3d
${TID[${PID[$A}]}`.log
PID[$A]=0
fi

# Salgo del ciclo
principal si todas
las tareas fueron
hechas.
if [ ${PID[$A]} -gt 0 ]; then
TERMINO=false
A=$((SPARALELO-1))
fi

A=$((A+1))

done
done

echo ":::: FIN CORRIDA (`date
+`%d/%m/%Y - %H:%M:%S`)" >>
$LOGCPU

# FIN ALGORITMO #

# Mergeo los archivos temporales
en un solo log para toda la
corrada
cat $DIRLOG/$JOBLOGTMP* >
$DIRLOG/corrada_paralela.$$
`date +%d`.log
rm -f $DIRLOG/$JOBLOGTMP*
$LOGCPU
}

```

## VARIABLES DE ENTORNO

La definición es muy similar a la de las variables globales en programación: son variables accesibles por todas las funciones. Éstas son algunas variables de entorno que encontraremos en la mayoría de las distribuciones de GNU/Linux:

- DISPLAY:** el nombre del dispositivo en el que el servidor de ventanas X mostrará su salida.
- HOME:** el directorio **home**.
- LOGNAME:** el nombre utilizado en el login.
- PATH:** una cadena que contiene el listado de directorios donde se buscarán los programas si no se encuentran en el directorio actual.
- PS1:** define el tipo de línea de comandos.
- SHELL:** el shell estándar. En nuestro caso, **SHELL=/bin/bash**.
- TERM:** esta variable define el tipo de terminal que estemos utilizando.

## Creación de archivos HTML

Lo que hace este script es simple: verifica si existe el archivo pasado como parámetro. Si existe, abre el **Vim** con ese archivo. En caso contrario, crea un archivo nuevo con las etiquetas básicas de html ya creadas. Es bueno ubicarlo en la carpeta **/bin/** con permisos **a+x** de forma que puedan ejecutarlo todos los usuarios desde cualquier sitio del sistema operativo.

```

#!/bin/bash
#Funcion para crear el archivo
que contendra las etiquetas

```

```
html.
# Pablo E. Terradillos - Tehsis
# http://tehsis.blogspot.com

genhtml() {
    echo "<html>" > $1
    echo "    <head>" >> $1
    echo "        <title>INGRE
        SAR UN TITULO</title>"
        >> $1
    echo "    </head>" >> $1
    echo "    <body>" >> $1
    echo "
        /...../" >> $1
    echo "    </body>" >> $1
    echo "</html>" >> $1
    vi $1;
}
```

```
#Primero se comprueba que el
archivo pasado como
parametro no exista
if [ -f $1 ]
then
#Si existe, se abre el editor
de textos vim con dicho
archivo
vi $1
```

```
else
#Si no existe, primero crea el
archivo con las etiquetas, y
luego abre el vim con dicho
archivo
genhtml $1;
fi
```

### Versión alternativa del comando sleep

Este script funciona igual que el comando **sleep** (el cual hace una pausa de la cantidad de segundos definida como parámetro), pero a diferencia de él, éste informa en pantalla el progreso de los segundos a medida que van pasando.

```
#!/bin/sh
USAGE="Uso: $0 segundos"
if [ -z $1 ]; then
    echo $USAGE
    exit 1
fi
for i in `seq $1 -1 1`; do
```

```
if [ $i -lt 60 -o $((($i % 60))
    -eq 0 ]; then
    echo -n $i..
fi
sleep 1
done
```

### La magia de las llaves

Los símbolos { y } en bash sirven para encerrar una serie de ítems que luego serán procesados como parámetros independientes. Veamos un ejemplo:

```
echo {uno,dos,tres,cuatro}
```

Mostrará en pantalla:

```
uno dos tres cuatro
```

Ahora veamos otro ejemplo en el que queda más claro el funcionamiento de esto. Vamos a pasarle dos parámetros a **echo**. Uno con una serie de ítems encerrados entre llaves y otro simple:

```
echo {uno,dos,tres,cuatro}perro
```

Mostrará en pantalla:

```
unperro dosperro tresperro cuatroperro
```

Feo, ¿no? Si agregamos espacios obtendremos un funcionamiento no esperado (como si agregáramos un espacio de más a cualquier otra estructura de un script de bash). Por suerte, podemos utilizar las comillas de la siguiente manera para solucionar esto:

```
echo {"uno ","dos ","tres ","cuatro
"}perro
```

### Función para intercambiar dos valores en una variable

Esta función acepta valores como parámetro y alterna entre ellos el contenido de una variable cada vez que se ejecuta:

```
var_toggle()
{
    eval "_VAR_TOGGLE=\$$1"
    [ ${_VAR_TOGGLE:-0} = ${3:-0} ] &&
        _VAR_TOGGLE=${2:-1} ||
            _VAR_TOGGLE=${3:-0}
    eval "$1=\$_VAR_TOGGLE"
}
```

Veamos varios ejemplos posibles de uso de esta función. Si le pasamos dos parámetros, cada vez que la ejecutemos, la variable tendrá uno de esos dos parámetros:

```
var_toggle var 13 5; echo $var
```

El sistema devuelve:

```
5
```

Lo ejecutamos de nuevo:

```
$ var_toggle var 13 5; echo $var
```

Y el sistema devuelve:

```
13
```

## FUNCIONES REPETITIVAS EN LA LÍNEA DE COMANDOS

Es posible utilizar prácticamente todas las estructuras que se usan en los scripts desde la línea de comandos. Y aunque no es muy cómodo programar scripts allí directamente, es posible usar algunas estructuras que nos ayudarán a simplificar y automatizar muchas tareas. Por ejemplo, podemos usar la estructura **for** para crear un ciclo repetitivo y ejecutar un determinado comando muchas veces. El siguiente comando procesa todos los archivos del directorio actual y genera una copia de seguridad de cada uno de ellos:

```
$ for file in *; do cp $file $file.bak; done
```

Finalmente, si le pasamos un solo parámetro, los valores son alternados entre el actual y cero.

## Varias formas de leer la primera línea de un archivo

A veces se nos puede presentar la necesidad de almacenar en una variable la primera línea de un archivo de texto. Veamos algunas formas posibles de llevar a cabo esta tarea:

```
var=`head -1 ARCHIVO`
```

Almacenará la primera línea del archivo **ARCHIVO** en la variable **var**.

Pero, si no queremos usar un comando externo, lo podemos hacer internamente haciendo uso de la función **read** del siguiente modo:

```
read var < ARCHIVO
```

Y si necesitamos leer, por ejemplo, las primeras cuatro líneas, podemos hacer uso de las llaves (ver truco **La magia de las llaves** en este mismo capítulo) del siguiente modo:

```
{
    read var1
    read var2
    read var2
    read var3
    read var4
} < ARCHIVO
```

## MÚLTIPLES COMANDOS

Recordemos que es posible ejecutar múltiples comandos uno tras otro (o sea, el comando dos se ejecuta al finalizar el comando uno y así sucesivamente...) utilizando la siguiente estructura:

```
comando1 ; comando2 ; comando 3
```

## Función para centrar el texto en pantalla

Si ya programamos scripts en bash alguna vez, seguramente habremos tenido la idea de presentar algún texto centrado en la pantalla. La función que presentamos a continuación es una de las posibilidades, y acepta un parámetro para definir la cantidad de caracteres a centrar:

```
centrar()
{
    c_ancho=$1
    shift
    c_texto="$*"
    c_ancho=$(( ($c_ancho +
        ${#c_texto}) / 2 ))
    printf
        "%${c_ancho}.${c_ancho}
        s\n" "$c_texto"
}
```

Luego se utiliza de la siguiente manera:

```
centrar [valor] [cadena]
```

## Utilizar grep dentro de scripts de Bash

Es posible utilizar todos los comandos del sistema dentro de nuestros scripts de Bash. Esto aumenta exponencialmente las capacidades de nuestras aplicaciones ya que así sólo están limitadas a nuestro conocimiento de las aplicaciones del sistema. Un buen ejemplo para demostrar el uso de los comandos externos en Bash es realizar búsquedas en archivos. Para esto, nada mejor que usar el comando **grep** del siguiente modo:

```
#!/bin/sh
```

```
Macbook-de-Facu:~ hfarena$ ./busca
Buscando usuarios con la palabra "nobody" en la base de datos
Los resultados son: nobody
Macbook-de-Facu:~ hfarena$
```

Figura 12. Utilizar el comando **grep** dentro de un script de Bash permite realizar búsquedas dentro de archivos y procesar los resultados como una cadena normal.

```
pedros=`grep -i pedro /etc/passwd |
cut -d: -f1`
```

```
echo "Buscando usuarios
con la palabra \"pedro\" en la base de
datos"
echo "Los resultados son: $pedros"
```

Como vemos, lo que se utiliza para ejecutar un comando es la línea completa entre comillas invertidas simples. Esto se lo asigna a una variable y como resultado obtenemos dentro de ella la salida del comando. Hay que asegurarse de usar las comillas correctas ya que son las únicas que le indican a Bash que debe ejecutar un comando.

1 Para contar la cantidad de líneas que hay en una variable en Bash podemos usar la siguiente sentencia: `NUM_LINEAS=`wc -l file | awk '{ print $1 }`"`.

2 Es posible conectarse a telnet desde un script de Bash y enviar comandos. Para eso armamos el siguiente script:

```
#!/bin/sh
host=10.0.0.1
port=23
login=facundo
passwd=facundo
cmd="ls"

echo open ${host} ${port}
sleep 1
echo ${login}
sleep 1
echo ${passwd}
sleep 1
echo ${cmd}
sleep 1
echo exit
```

Para usarlo simplemente usamos el nombre del script en conjunto con una tubería con el comando telnet. Si los tiempos de espera son demasiado cortos, podemos agregarle uno o dos segundos al comando sleep.

3 Los scripts de usuarios deben guardarse en `/usr/bin`, y los del administrador en `/usr/sbin`.

4 ¡Usemos los comentarios! Cuantas más líneas de comentarios agreguemos en nuestros códigos, más fácil será actualizarlos y modificarlos en el futuro.

5 Si estamos probando scripts que modifican archivos, trabajemos siempre con versiones de resguardo de los archivos a procesar hasta estar completamente seguros de que el script funciona correctamente.

6 Para verificar si un script está siendo usado por `root` podemos usar el siguiente método:

```
# Verifica si el script no esta
siendo ejecutado por root
if [ "$(id -u)" != "0" ]; then
    echo "Este script debe ser
    ejecutado por root."
    exit 1
fi
# Verifica si el script no debe
ser ejecutado por root
if [ "$(id -u)" == "0" ]; then
    echo "Este script no debe ser
    ejecutado por root."
    exit 1
fi
```

# Scripts en PERL

En el ámbito de la gestión de servidores, PERL es uno de los lenguajes de scripting más populares gracias a la infinidad de herramientas de gestión de información que posee integradas. En este capítulo, veremos cómo sacarles el máximo provecho a las mejores características de este lenguaje.

Tutorial de PERL en 10 minutos  
 Cómo leer y escribir archivos de texto  
 Cómo crear nuestras propias funciones de PERL  
 Cómo crear módulos en PERL  
 Listar archivos y directorios  
 Cómo ejecutar comandos externos en Perl  
 Bases de datos simples en Perl  
 Cómo partir y unir cadenas  
 Gestión de archivos desde scripts PERL  
 Control de claves  
 Cómo comentar múltiples líneas de código  
 Bonus tips

Es un tipo de variables de Perl. Estas variables pueden almacenar dentro de sí tanto cadenas de caracteres como números enteros.

Perl fue especialmente diseñado para el manejo de bases de datos textuales. Permite el manejo de variables, arreglos, estructuras condicionales y bucles. Para crear un script en Perl debemos seguir los mismos pasos que para crear uno en bash, tal como vimos en el capítulo anterior. La única diferencia radica en la ruta del intérprete. A continuación veremos un pequeño script en Perl. Para crearlo podemos utilizar el editor VI (luego debemos recordar que tenemos que otorgarle permisos de ejecución).

```
#!/usr/bin/perl
# Ejemplo: ¡Hola Mundo!

print "¡Hola Mundo!";
```

PERL (*Practical Extraction and Report Language*) es un lenguaje que tiene muchos años de vida. En sus comienzos empezó siendo un lenguaje muy utilizado para procesar **bases de datos de archivos de texto** (gracias a las expresiones regulares, uno podía realizar búsquedas y filtros muy fácilmente). Luego, pasó a ser la bandera de las primeras aplicaciones web como el buscador Yahoo!. De hecho, todos los sitios web de los años 90 que necesitaban un poco de procesamiento de datos utilizaban el popular sistema CGI basado en Perl, hasta que apareció PHP y la actual tecnología AJAX.

Hoy por hoy, Perl sigue siendo un excelente lenguaje para desarrollar scripts para la consola, y por eso vamos a focalizarlos en este aspecto en este capítulo. Generalmente, Perl se incluye por defecto en la instalación de la mayoría de las distribuciones. Para asegurarnos de esto, podemos buscar el archivo `/usr/bin/perl` o ejecutar `perl -v`. El programa devolverá algo como lo que vemos en la **Figura 1**.

```
Last login: Mon May 28 08:20:12 on console
Welcome to Darwin!
You have new mail.
Macbook-de-Facu:~ hfarena$ perl -v

This is perl, v5.8.6 built for darwin-thread-multi-2level
(with 3 registered patches, see perl -V for more detail)

Copyright 1987-2004, Larry Wall

Perl may be copied only under the terms of either the Artistic License or the
GNU General Public License, which may be found in the Perl 5 source kit.

Complete documentation for Perl, including FAQ lists, should be found on
this system using 'man perl' or 'perldoc perl'. If you have access to the
Internet, point your browser at http://www.perl.org/, the Perl Home Page.

Macbook-de-Facu:~ hfarena$
```

**Figura 1.** Información de la versión de PERL instalada en el sistema.

Alguna vez puede darse la situación de tener que averiguar, desde un script de PERL, qué sistema operativo se está utilizando. Para esto, nada más simple que verificar el contenido de la variable `$^O`. Bajo Windows, devolverá el valor `MSWin32`. Bajo Linux, simplemente `linux`.

La ruta al intérprete del lenguaje Perl es `/usr/bin/perl`. Después de un comentario, figura la función `print` con la cadena **¡Hola Mundo!** como parámetro. Como resultado, este ejemplo muestra lo que podemos ver en la **Figura 2**. Tal como los otros lenguajes populares de la actualidad, Perl nos permite realizar el manejo de variables de forma sumamente sencilla. Las variables se definen del siguiente modo:

`$variable = valor;`

Nótese el símbolo `$` antes del nombre de la variable. Esto define que es una **variable escalar**. Luego se ingresa el valor, que puede ser un entero, un flotante o una cadena. Para imprimir una variable, podemos utilizar la función `print` del siguiente modo:

```
print "El valor de la variable 'variable'
es: $variable"
```

Por último, cabe destacar que cada función se finaliza con un símbolo `;` como en el lenguaje de programación C. Otra forma de almacenar los datos en la memoria es por medio de los denominados: **arreglos de variables**. Éstos son formas de agrupar variables en un mismo conjunto. Son fáciles de definir, la manera más sencilla es la siguiente:

```
@arreglo = (val1, val2, val3, ...);
```

A diferencia de las variables, los arreglos comienzan con el símbolo `@`. Para imprimir el valor de un arreglo tenemos que utilizar la función `print` del siguiente modo:

```
print "El valor 2 del arreglo es:
$arreglo[2]";
```

Si queremos imprimir el arreglo completo, podemos utilizar la función `print` con el siguiente contenido:

```
print "El arreglo completo contiene:
@arreglo"
```

Una particularidad de Perl es la posibilidad de crear arreglos con etiquetas para cada valor. Para crear un arreglo de este tipo, se utiliza el símbolo `%` antes del nombre del arreglo. El siguiente es un ejemplo:

```
%mi_arreglo("valor1", 10, "valor2",
20, "valor3", 30);
```

```
Macbook-de-Facu:~ hfarena$ chmod +x script
Macbook-de-Facu:~ hfarena$ ./script
Hola Mundo!Macbook-de-Facu:~ hfarena$
```

**Figura 2.** Salida de nuestro primer script de PERL.

El arreglo **mi\_arreglo** contiene tres valores que pueden ser llamados mediante las etiquetas **valor1**, **valor2** y **valor3**.

```
print "El valor 1 es: $mi_arreglo{valor1}"
```

Podemos notar que la llamada a la etiqueta del arreglo se realiza con las llaves. El siguiente es un ejemplo de arreglos con etiquetas:

```
#!/usr/bin/perl
# Ejemplo: Arreglos con
# etiquetas

%familia=("papa", 47, "mama",
45, "hermano", 15, "hermana",
12);
print "Padre: $familia{papa}";
print "Madre: $familia{mama}";
print "Hermano: $familia{hermano}";
print "Hermana: $familia{hermana}";
```

En este ejemplo se define un arreglo con las etiquetas de los integrantes de una familia y sus respectivas edades. Finalmente, se las imprime en pantalla como vemos en la **Figura 3**.

Perl también nos permite evaluar expresiones de forma muy sencilla. Veamos el siguiente ejemplo:

```
10:  #!/usr/bin/perl
20:  # Ejemplo:
```

```
Macbook-de-Facu:~ hfarena$ ./script
Padre: 47Madre: 47Hermano: 47Hermana: 47
Macbook-de-Facu:~ hfarena$
```

**Figura 3.** Utilización de arreglos con etiquetas en un script de PERL.

### Expresiones

```
30: $val1=2;
40: $val2=5;
50: print "Realizando
Multiplicacion...";
60: $val3=$val1*$val2;
70: print "$val3";
```

En este pequeño programa definimos dos variables (líneas 30 y 40) con los valores 2 y 5. Se imprime un pequeño mensaje y se define una tercera variable con el resultado de la multiplicación de las dos anteriores. Por último, se imprime la última variable en pantalla. Perl ofrece otros comandos de evaluación de expresiones muy útiles:

**eq:** compara 2 cadenas.

**ne:** diferencia.

**lt:** menor que.

**gt:** mayor que.

Para concatenar dos cadenas se puede utilizar el operador de punto (.). El siguiente es un ejemplo:

```
$nombre = "Carlos";
$apellido = " Lopez";
$nombre_completo =
$nombre.$apellido
```

Por consiguiente, si imprimimos la variable **\$nombre\_completo**, obtendremos:

### 'Carlos Lopez'

En cuanto al ingreso de datos, en Perl es muy sencillo. Para realizar esta acción se define una variable con el flujo de entrada estándar como ingreso. El siguiente es un ejemplo:

```
$entrada=<STDIN>
```

Recordemos que el flujo de entrada estándar es **STDIN**. Estas definiciones pueden aparecer en cualquier parte del código.

```
#!/usr/bin/perl

print "Ingrese su edad:";
$edad = <STDIN>;
print ("Su edad es: $edad");
```

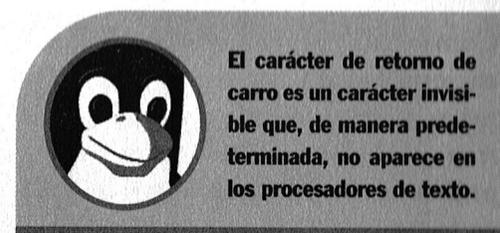
Este pequeño programa muestra un mensaje y luego define una variable para realizar el ingreso de datos. Por último, lo muestra en pantalla como una variable cualquiera (**Figura 4**).

```
Macbook-de-Facu:~ hfarena$ nano script
Macbook-de-Facu:~ hfarena$ ./script
Ingrese su edad:25
Su edad es: 25
Macbook-de-Facu:~ hfarena$
```

**Figura 4.** Ingresar datos por teclado es tan sencillo como asignarle el flujo de entrada estándar a una variable.

## III OPERACIONES ARITMÉTICAS SOBRE ARREGLOS

Es posible llevar a cabo operaciones aritméticas sobre arreglos de variables en Perl. Por ejemplo, **#\$arreglo /= 2**; dividirá el arreglo en dos y descartará la segunda parte. Podemos probar con otras operaciones para descubrir resultados interesantes.



El carácter de retorno de carro es un carácter invisible que, de manera predefinida, no aparece en los procesadores de texto.

Siempre recordemos utilizar la función **chop** para eliminar el carácter de retorno de carro en las cadenas. Su forma es:

```
chop ($variable=<STDIN>);
```

Ahora veamos algunas estructuras condicionales y repetitivas. La estructura **if** en Perl es muy similar a la del lenguaje C. La forma básica es:

```
if ( condicion ) {
    funciones
}

else {
    funciones
}
```

Como podemos ver, la condición se encierra entre paréntesis y las funciones respectivas a cada opción se encierran entre llaves. Veamos un ejemplo práctico cuya salida podemos observar en la **Figura 5**:

```
#!/usr/bin/perl
print "Ingrese un numero:";
$numero=<STDIN>;
if ($numero >= 1000) {
    print "Su numero es mayor
        (o igual) que 1000!";
}
else {
    print "Su numero es menor
        que 1000!";
}
```

Los operadores que podemos utilizar en la condición son los siguientes:

- eq**: igual a.
- >**: mayor que.
- <**: menor que.
- >=**: mayor o igual que.
- <=**: menor o igual que.
- ne**: diferente.

```
Macbook-de-Facu:~ hfarena$ ./script
Ingrese un numero:100
Su numero es menor que 1000!
Macbook-de-Facu:~ hfarena$ ./script
Ingrese un numero:1000
Su numero es mayor (o igual) que 1000!
Macbook-de-Facu:~ hfarena$ ./script
Ingrese un numero:1000
Su numero es mayor (o igual) que 1000!
Macbook-de-Facu:~ hfarena$ ./script
Ingrese un numero:0
Su numero es menor que 1000!
```

**Figura 5.** Utilización de operadores en una sentencia condicional *if*.

Ahora veamos una de las particularidades de PERL. La estructura **unless** ejecuta una serie de funciones sólo si una condición es falsa. Veamos un ejemplo:

```
unless (numero > 1000) {
    print "Su numero es menor
        que 1000!";
}
```

Perl también puede manejar bucles condicionales como el **while**. La estructura del **while** en Perl es la siguiente:

```
while (condicion)
{
    funciones...
}
```

También podemos usar la entrada estándar para recibir los datos utilizando **while**. Su forma es la siguiente:

```
while(<STDIN>) {
    funciones...
}
```

```
Macbook-de-Facu:~ hfarena$ ./script
0
1
2
3
4
5
6
7
8
9
10Macbook-de-Facu:~ hfarena$
```

**Figura 6.** Utilización del ciclo repetitivo *for*.

Como en todo lenguaje de programación, el ciclo **for** nos permite repetir una porción de código una determinada cantidad de veces. Su forma de utilización es la siguiente:

```
for (expresion1; expresion2;
    expresion3) {
    funciones...
}
```

Como en los lenguajes antes mencionados, se puede omitir cualquiera de las expresiones pero se debe incluir el punto y coma (;). El siguiente es un ejemplo simple: un contador del 1 al 10 (**Figura 6**).

```
#!/usr/bin/perl

for($a=0 ; $a <= 10; $a++ ) {
    print "\n$a";
}
```

Bien, ahora que ya conocemos las bases de la programación de scripts en PERL, veamos algunos trucos para sacarle el máximo provecho a este lenguaje de programación.

## Cómo leer y escribir archivos de texto

Perl fue especialmente diseñado para trabajar con datos en archivos de texto. Si bien su estudio completo lleva mucho más que un par de páginas de un libro, en esta sección intentaremos dar una visión general de cómo trabajar el sistema. Perl utiliza la función **open** para realizar la apertura de archivos. Su forma es la siguiente:

**open (NOMBRE, "archivo");**

El nombre es el descriptor del archivo en cuestión, el archivo siempre se encierra entre comillas. Por defecto, Perl abre los archivos en modo sólo lectura. Para

## III ADVERTENCIAS EN PERL

Podemos habilitar y deshabilitar los mensajes de advertencia de Perl dentro de nuestro código de manera rápida y sencilla. Para deshabilitarlos, debemos utilizar la sentencia **no warnings**; y para habilitarlos, la sentencia **use warnings**;

abrirlo en modo escritura se agrega un símbolo > al principio de la ruta al archivo, como vemos a continuación:

```
open(NOMBRE, ">archivo");
```

Si se quiere abrir el archivo para modo **append** (agregar los datos al final), se agregan dos símbolos >:

```
open(NOMBRE, ">>archivo");
```

Una vez que se encuentra abierto el archivo ya podemos escribir sobre él. Como ejemplo, se puede utilizar la función **print** del siguiente modo:

```
print NOMBRE "cadena"
```

Esta función imprimirá la cadena en el archivo identificado con la etiqueta **NOMBRE**. Veamos el siguiente ejemplo:

```
10: #!/usr/bin/perl
20: open (ARCHIVO,
    ">>file.txt");
30: print "Ingrese su nombre: ";
40: chop($nombre=<STDIN>);
50: print "Ingrese su apellido:
    ";
60: chop($apellido=<STDIN>);
70: print ARCHIVO "$apellido,
```

```
$nombre\n";
80: close ARCHIVO;
```

En este ejemplo se abre un archivo llamado **file.txt** para modo **append** (línea 20). En las líneas 40 y 60 se hace el ingreso de datos con la función **chop** para que elimine el carácter final de la cadena (de esta manera, los datos se imprimirán en el archivo uno al lado del otro). Por último, en la línea 70 se imprime en **ARCHIVO** la cadena formateada y en la línea 80 se cierra el archivo. El resultado que obtenemos con este código lo podemos ver en la **Figura 7**. Para llevar a cabo la lectura de archivos, la utilización de funciones es la misma. Primero se debe abrir el archivo del siguiente modo:

```
open (ARCHIVO, "file.txt");
```

Luego pueden almacenarse en una variable los datos del archivo, siempre línea por línea del siguiente modo:

```
#!/usr/bin/perl
open(ARCHIVO, "file.txt");
$linea=<ARCHIVO>;
print $linea;
```

También podemos utilizar una función **while** para procesar todas las líneas sin tener que repetir código. Una forma puede ser la siguiente:

```
#!/usr/bin/perl
open(ARCHIVO, "file.txt");
while (<ARCHIVO>) {
    print $_;
}
close ARCHIVO;
```

## Cómo crear nuestras propias funciones de PERL

Con Perl podemos crear subfunciones para permitirle al programador manejarse en un confortable entorno de programación estructurada. Las funciones en

Perl se definen al principio del código con la función **sub** del siguiente modo:

```
sub mi_funcion {
    (...)
}
```

Luego, para llamar a la función creada, directamente utilizamos su nombre:

```
mi_funcion;
```

También es posible pasar parámetros a las funciones. Por defecto, estos parámetros se guardan en **@\_** dentro de la función:

```
10: #!/usr/bin/perl
20: sub imprime {
30: print "\nEl contenido
    de los parametros es @_";
40: }
50: imprime "Hola Mundo";
```

## !!! ¡CUIDADO CON LA SINTAXIS!

Perl tiene la particularidad de ser uno de los lenguajes que más oportunidades ofrece para ofuscar el código. ¿Qué significa esto? Que es posible escribir código de forma tan pero tan compleja que no se entienda absolutamente nada. Por ejemplo:

```
@P=split//,".URRUU\c8R";@d=split//,"\nrekah xinU / lreP rehtona tsuJ";sub pf
@p{"r$p","u$p"}=(P,P);pipe"r$p","u$p";++$p;($q*=2)+=$f=!fork;map{$P=$P[$f^ord
($p{$_})&6];$p{$_}=/^$P/ix?$P:close$_}keys%p;p;p;p;p;map{$p{$_}=~/^[P.]/&&
close$_}p;wait until?;map{/^r/&&<$_>}%p;$_=$d[$q];sleep rand(2)if/$/;print
```

¿Qué hace? Será cuestión de probarlo y descubrirlo. Podemos encontrar más información sobre código ofuscado en [http://en.wikipedia.org/wiki/Obfuscated\\_code](http://en.wikipedia.org/wiki/Obfuscated_code) o en [http://es.wikipedia.org/wiki/Código\\_ofuscado](http://es.wikipedia.org/wiki/Código_ofuscado).

```
Macbook-de-Facu:~ hfarena$ ./script
Ingrese su nombre: Facundo
Ingrese su apellido: Arena
Macbook-de-Facu:~ hfarena$ cat file.txt
Arena, Facundo
Macbook-de-Facu:~ hfarena$
```

**Figura 7.** Almacenamiento de la información ingresada por el usuario en un archivo de texto.

Este programa crea una función en la línea 20 que imprime el contenido de la variable @\_. En la línea 50 se realiza el llamado a la función, pasándole como parámetro la cadena **Hola Mundo**.

## Cómo crear módulos en PERL

Un módulo es un conjunto de subrutinas. Por lo general, en un módulo se juntan subrutinas que hagan cosas que están relacionadas. Por ejemplo, se puede tener un módulo con subrutinas que se utilicen para conectarse a un servidor web, bajar o subir archivos desde ellos, etcétera. Separar las subrutinas en un módulo nos permite hacer uso de ellas desde otro programa simplemente mencionando que se quiere utilizar ese módulo. Ni siquiera es necesario copiar y pegar las subrutinas en el archivo del programa que estamos haciendo.

Un módulo será un archivo de texto escrito exactamente igual que los programas escritos hasta ahora pero con una extensión diferente, que por lo general es **.PM**. Todo el código que se escriba



Una de las particularidades de PERL es que permite, por medio de la utilización de módulos, extender las capacidades de nuestro programa. De esta manera, es posible hacer programas con interfaces gráficas, con capacidades de acceso a recursos de red, y muchas cosas más. Para poder lograrlo, simplemente debemos familiarizarnos con PERL y aprender a utilizarlo bien.

que no se encuentre dentro de una subrutina en el momento de incluir el módulo, será ejecutado como una suerte de código inicializador de éste. El código podrá hacer cualquier cosa. Lo único diferente que tiene un módulo con respecto a un programa normal, es que deberá terminar con un valor verdadero para que el programa que lo incluya sepa que se pudo inicializar y cargar bien. Para ese fin habrá que agregar: **1**; como última línea del módulo. Como el valor **1** es verdadero, al ejecutarse hará que el módulo devuelva un valor verdadero y, en consecuencia, pueda ser incluido. Para hacer uso de un módulo existen tres formas:

## INSERTAR ESPACIOS EN VARIABLES

La magia de las expresiones regulares nos permite hacer cosas increíbles. Por ejemplo, la siguiente sentencia convierte el contenido de la variable texto **HolaMundoLoco** en **Hola Mundo Loco**:

```
$texto = "HolaMundoLoco";
```

```
$texto =~ s/([a-z])([A-Z])/ $1 $2/g;
```

Esta sentencia sirve para cualquier cadena que tenga palabras pegadas que se encuentren identificadas cada una por una letra mayúscula al comienzo.

- Utilizar la función **do**. Este operador carga el módulo especificado cada vez que se ejecute. Si queremos incluir el archivo **servicios.pl** al programa que se está haciendo habrá que escribir: **do "servicios.pl"**; Cada vez que se ejecute esa línea se volverá a incluir el archivo **servicios.pl**.
- Utilizar la palabra reservada **require**. **Require** funciona exactamente igual que **do** pero posee una única diferencia: carga el archivo especificado sólo la primera vez. Todas las demás veces que se trate de cargarlo serán ignoradas.
- Utilizar la palabra reservada **use**. Ésta es la forma normal de incluir un módulo. A diferencia de las dos anteriores, cuando se utiliza **use**, el módulo se incluye antes de que se ejecute alguna sentencia de código (en tiempo de compilación). De esta forma, si el módulo no existe, Perl no podrá ejecutar el script.

Perl busca un módulo especificado en un conjunto de directorios. Estos directorios están especificados en una lista llamada **@INC**. Si el módulo no se en-

cuentra en ninguno de estos directorios, entonces no existe. Si el módulo sí existe pero está en otro directorio, habrá que poner ese directorio en la lista **@INC**. Para ello existen dos formas:

- Utilizar la subrutina **BEGIN**, que se llama antes de que se incluyan los módulos. Por ejemplo:

```
sub BEGIN {
    push @INC, "/mi/directorio/
    bibliotecas";
}
```

- Utilizar el módulo especial **lib**. Por ejemplo:

```
use lib "/mi/directorio/bibliotecas";
```

## Listar archivos y directorios

Hay varias formas de generar un listado de archivos desde un script de PERL. Una opción es la función **glob**. Esta fun-

Figura 8. El CPAN ([www.cpan.org](http://www.cpan.org)) es un gran repositorio de módulos para PERL.

ción tiene como argumento el directorio y los archivos que se quieren listar. En esta función se especifica de la misma forma que se hace en el shell. Esto significa que el carácter \* y el ? no serán interpretados literalmente, sino que el \* concordará con cualquier tipo y cantidad de caracteres y el ? concordará con un solo carácter cualquiera.

**Glob** devolverá una lista donde cada elemento es el nombre de uno de los archivos encontrados. Por defecto, **glob** omite todos los archivos y directorios que comiencen con un punto. Por ejemplo, para obtener todos los archivos **.JPG** de un directorio de imágenes ingresamos:

```
@archivos = glob("./imagenes/
*jpg");
foreach $archivo (@archivos) {
    print "$archivo\n";
}
```

## Cómo ejecutar comandos externos en Perl

Para ejecutar programas, Perl ofrece diferentes métodos. Uno de ellos es la función **system** que ejecuta directa-

mente como comando la cadena que se le pasa como parámetro. Su forma de utilización es la siguiente:

### system "ls"

También podemos utilizar las comillas invertidas `` encerrando un comando para guardar su salida en una variable. Un ejemplo puede ser el siguiente:

```
$salida=`ls -l`
```

De este modo, si imprimimos la variable **salida**, tendremos el listado del directorio actual.

## Bases de datos simples en Perl

**DBM** es un sistema de bases de datos que ya lleva muchos años en el campo de los sistemas operativos UNIX. Si bien es un sistema muy sencillo en todos sus aspectos, es muy útil cuando se quiere desarrollar pequeñas aplicaciones que no hagan uso intensivo de funciones complejas de manejo de datos. Actualmente, existen varias implementaciones del UNIX DBM (gDBM del proyecto GNU, nDBM y oDBM



Las DBM son un sistema muy simple de base de datos y son un excelente ejemplo para comenzar a adentrarse en la programación en PERL. De todas maneras, animamos a los lectores a que, a medida que avancen y profundicen sus conocimientos, continúen con su aprendizaje y también investiguen sobre la integración de PERL con MySQL o PostgreSQL.

son algunos ejemplos de ellas). Perl utiliza su propia implementación (sDBM), que si bien no es tan potente como la gDBM, posee casi todas las funciones más importantes.

Un detalle muy relevante es que usar el sistema DBM en Perl nos asegura que nuestro sistema funcionará en cualquier computadora que tenga un intérprete de Perl instalado, aun si se trata de diferentes sistemas operativos. De esta forma, un sistema que utiliza DBM desarrollado en Perl para GNU/Linux funciona de forma correcta y sin tener que hacerle modificaciones de código tanto en un sistema que posee Mac OS X como en uno con Windows. En Perl, todo el control de bases de datos DBM se realiza por medio de los ya conocidos **hashes**. Nuestro querido comando **tie** con algunos parámetros extras nos permitirá crear, abrir, modificar y cerrar bases de datos. Estos parámetros extras son los siguientes:

- El nombre del hash.
- El nombre del modulo DBM.
- El archivo de la base de datos.
- Definición de las opciones de acceso al archivo.
- Permisos que tendrá el archivo.

Una función **tie** de ejemplo para crear una base de datos es la siguiente:

```
tie %dbm, 'SDBM_File', "base.db",
O_CREAT|O_RDWR, 0644;
```

Como podemos ver, el uso de esta función es muy sencillo. El primer parámetro (**%dbm**) nos servirá para hacer referencia a nuestra base de datos. El archivo de base de datos lo definimos en el tercer parámetro (**base.db**) y haremos que Perl lo cree con permisos de lectura y de escritura para nosotros (**O\_CREAT|O\_RDWR, 0644**). Una aclaración: el cuarto parámetro define de qué forma Perl accederá al archivo. El quinto parámetro indica los permisos que el archivo tendrá una vez creado. Las bases de datos DBM al usar el sistema de hashes funcionan bajo el modelo de **palabra\_clave=valor**. De esta forma, si usamos la sintaxis de hashes en Perl para agregar un nuevo dato a nuestra base de datos tendremos que tipear algo como lo siguiente:

```
$dbm{'nombre'}="Facundo";
```

Ahora bien, ¿qué sucede si queremos manejar múltiples registros? Bueno, tendremos que ingeniárnosla para arreglarnos con este sistema. Una buena idea puede ser implementar la base de datos bajo la siguiente estructura:

```
1.nombre="Facundo"
1.apellido="Arenas"
2.nombre="Fernando"
2.apellido="Perez"
```

## CHEQUEO DE UNA BASE DE DATOS

Podemos hacer uso de la sentencia **or die** en caso de que la base de datos no haya sido creada. Por ejemplo:

```
tie %dbm, 'SDBM_File', "archivo.db", O_RDWR, O or die "Error abriendo archivo.db\n";
```

Este recurso es muy sencillo de implementar y nos será de gran utilidad cuando llegue la hora de realizar el proceso de detección de errores en el código.

Y así sucesivamente. El primer número es útil para asignarle un valor de registro a cada una de las personas que tengamos registradas en nuestra base de datos. Para acceder a alguno de los datos de nuestra base de datos, la cosa es bastante similar. Por ejemplo, podemos asignarle un registro a una variable del siguiente modo:

```
$variable=%dbm{'1.nombre'};
```

Siguiendo la línea de nuestros ejemplos, **\$variable** contendría la palabra **Facundo**. De todas formas, también podemos realizar un acceso directo al hash. El siguiente es un ejemplo para mostrar el contenido de una palabra clave en pantalla:

```
print "El contenido del primer registro es %dbm{'1.nombre'};
```

Es recomendable que antes de intentar acceder a alguna de las palabras clave de nuestra base de datos realicemos un chequeo, para cerciorarnos de que ese valor realmente exista. Una de las formas de hacer esto es encerrando el acceso a la palabra clave dentro de una función **if** del siguiente modo:

```
if(exists $dbm{'palabraclave'}) {  
  (...)  
}
```

Por último, para borrar una entrada en la base de datos, tenemos que usar la función **delete** del siguiente modo:

```
delete %dbm{'palabraclave'};
```

### III BYTES DE UNA CADENA

Es posible saber rápidamente cuántos bytes ocupa una cadena determinada del siguiente modo:

```
$tamaño = -s "hola mundo!";
```

Y para seguir con la línea de las cosas sencillas, diremos que cerrar una base de datos es cuestión de usar la función **untie** seguida del nombre de la base de datos. A continuación vemos un ejemplo válido de esto:

```
untie %dbm;
```

Recordemos que debemos cerrar la base de datos una vez terminado el acceso a ella. En general, esta operación se realiza al finalizar la ejecución del programa.

### Cómo partir y unir cadenas

Si se posee una cadena, como una entrada del archivo **/etc/passwd**, que tiene campos separados por delimitadores (en ese ejemplo el **:**), se puede separar esta cadena por campos para poder utilizar aquellos que nos sean de interés. Para ello, existe una función que despedaza una cadena en una lista donde cada elemento de la lista es un campo. Esta función se llama **split** y lleva dos parámetros. El primero es una cadena que especifica el delimitador que se quiere usar y el segundo es la cadena que se quiere romper. Así, **split** devolverá una lista con todos los campos. Para el ejemplo del **/etc/passwd**:

```
_ = "root:x:0:0:/root:/bin/bash";  
@campos = split ":", $_;
```

Dejará en **@campos** la lista ("**root**", "**x**", "**0**", "**0**", "**/root**", "**/bin/bash**"). Luego se podrá utilizar fácilmente cualquier campo. Para hacer el proceso inverso existe la función **join**. Esta función recibe como primer parámetro el delimitador con el que se quiera juntar los campos de una lista y como segundo parámetro la lista con los campos a juntar. Luego, **join** devolverá una cadena con todos los campos unidos por el delimitador. Finalmente, para cambiar el delimitador en una entrada del archivo de configuración en el código que se ejecutó antes:

```
$nueva_salida = join "#", @campos;
```

### Gestión de archivos desde scripts PERL

Existen algunas funciones para trabajar sobre archivos y directorios en el sistema. Estas funciones hacen tareas que no necesitan mirar los contenidos sino comunicarse con el sistema operativo para que las haga (como borrar un archivo, crear un directorio, etcétera).

### III ¡CONTROL!

Siempre que trabajemos con funciones que modifiquen archivos o directorios, es esencial que controlemos que la función esté realmente siendo aplicada al archivo o directorio que corresponda. Hay muchas formas de hacer esto, una de ellas es utilizar sentencias de control en las funciones de modificación de archivos y carpetas de forma que si éstas devuelven un código de error por que no se encuentra el archivo o la carpeta definido, el script devuelva un mensaje de error.

Por lo general, estas funciones son similares a los comandos que se utilizan normalmente desde la consola. Sin embargo, no hacen uso de esos comandos sino que se comunican directamente con el sistema operativo. Si bien es posible invocar un programa externo desde Perl, es recomendable no hacerlo a no ser que no quede otra solución o sea realmente conveniente. Hacer esto representa un posible problema de seguridad ya que nuestro programa necesita de código sobre el cual no se tiene ningún control y que, de haber sido alterado, podríamos ejecutar sin darnos cuenta código malicioso con privilegios más altos.

- **Cambio de directorio:** la función **chdir** cambia el directorio actual al que se le especifique por parámetro. La siguiente directiva cambiará el directorio actual a **/usr/bin**:  
**chdir("/usr/bin");**
- **Creación de un directorio:** la función **mkdir** crea un directorio. Esta función lleva como primer parámetro el directorio que se quiere crear y como segundo parámetro (opcional) los permisos que se le quieren otorgar. Vale aclarar que para escribir un valor en octal para usarlo de la mis-

ma forma en que se usa el **chmod**, el número debe empezar con 0.

**mkdir("/home/pepe");**

**Eliminación de un directorio:** para eliminar un directorio se utiliza la función **rmdir**. Para poder llevar a cabo esta acción el directorio debe estar vacío. Esta función sólo lleva un parámetro que indica el directorio que se desea eliminar. El siguiente ejemplo le quita el directorio personal al usuario **pepe**:

**rmdir("/home/pepe");**

**Renombrar un archivo:** la función **rename** sirve para cambiarle el nombre a un archivo. Esta función lleva de parámetros el nombre del archivo viejo y el nombre del archivo nuevo. Si el nombre del nuevo ya existe, entonces se perderán sus datos. Por ejemplo:

**rename("backup.tgz",**

**"backup-2003-04-22.tgz");**

**Eliminación de archivos:** para llevar a cabo esta tarea existe la función **unlink**. Esta función lleva una lista (de 1 a n) con los nombres de los archivos que se quieren eliminar. Por ejemplo:

**unlink("backup.tgz");**

## Control de claves

La función **crypt** nos permite generar una clave encriptada según una porción de texto determinada y una cantidad aleatoria de caracteres de texto (Figura 9). Con esos dos componentes genera una clave. Veamos un ejemplo:

```
#!/usr/bin/perl
print "Ingrese la palabra a
      encriptar:";
$clave = <STDIN>;
my $passwd = crypt($clave,
                  "esto es un poco de texto
                  aleatorio");
print "$passwd\n";
```

Lamentablemente, no existe una función **decrypt** para desencriptar el texto. De todas formas, esta función es muy útil para comprobar si dos porciones de texto son las mismas sin utilizar las porciones de texto en sí mismas. Con esta función y un poco de ingenio, con las estructuras condicionales es posible hacer eficientes sistemas de control de claves.

```
Macbook-de-Facu:~ hfarena$ ./script
Ingrese la clave a encriptar:perro
esEK1qfhglvNM
Macbook-de-Facu:~ hfarena$ ./script
Ingrese la clave a encriptar:caballo
esiMPaMATL0M.
Macbook-de-Facu:~ hfarena$ ./script
Ingrese la clave a encriptar:vaca
esEd0Et0h7KI
Macbook-de-Facu:~ hfarena$ ./script
Ingrese la clave a encriptar:pollito
es0FvXtDNM14E
Macbook-de-Facu:~ hfarena$ ./script
Ingrese la clave a encriptar:pajaro
es.IlpQ1bLJIw
Macbook-de-Facu:~ hfarena$
```

Figura 9. Creación de varias claves encriptadas con la función **crypt**.

## Cómo comentar múltiples líneas de código

Ya sabemos que si agregamos el símbolo numeral (#) logramos comentar una línea de código para impedir que PERL la ejecute. Cuando estamos trabajando y depurando un código, un recurso muy útil es comentar grandes porciones de código para poder focalizarnos en el resto e intentar detectar el error allí. Podemos agregar un símbolo numeral a cada línea de la sección a comentar, pero eso no es muy práctico. He aquí otra solución:

```
#!/usr/bin/perl
$str = "palabra1";
=for comment
$str = "palabra2";
$str = "palabra3";
=cut
print $str;
```

Lo que tenemos aquí es una sección de código encerrada entre las cláusulas **=for comment** y **=cut**. Todo lo que aparece entre **=for** y **=cut** se saltará automáticamente.

The screenshot shows the Perl Circus website. At the top is a navigation menu with links: Home, Usage, Begin, Scalars, Arrays, Hashes, Regex, Subs, Files, Modules, More Modules, Etc. Below the menu is a 'Home' section with a welcome message: 'Welcome. Perl Circus has been serving up fresh buttered Perl tricks since 2001. During that time our code snippets have delighted programmers of all ages. But the life of the Perl coder isn't all fun and games. A good Perl programmer has a skill set somewhere between a hacker and trainer of wild animals. Camels can be dangerous when cornered.' Below this is a paragraph: 'I have a small bit of experience with the rough and tumble life of a Perl programmer and that's what this site is about. These are just a few tricks I've collected over the years. Enjoy.' At the bottom, there are four columns of text advertising services: 'School or Factory Bells', 'CGI Programming', 'CGI Group Jobs Secrets', and 'Mumps Technology'. A small 'Ads by Google' logo is visible on the right side of the service columns. At the very bottom, there is a copyright notice: 'Unless otherwise noted, all code snippets on this site should be considered public domain. Other text is licensed under the Creative Commons Attribution-ShareAlike 2.5 License. Copyright 2006, Michael Matthews. All rights reserved. Happily hosted on Site5.'

Figura 10. Perl Circus ([www.perlcircus.org](http://www.perlcircus.org)) es uno de los más grandes repositorios de trucos y soluciones para PERL.

1 Es posible modificar el identificador **STDOUT** para que en lugar de apuntar a una pantalla apunte hacia un archivo. Veamos un ejemplo:

```
open(MIOUT, "> archivo.txt");
*STDOUT = *MIOUT;
print "holamundo";
```

Ahora, el archivo **archivo.txt** tendrá como contenido la salida del comando **print** tradicional.

2 Es posible escribir en dos descriptores de archivos al mismo tiempo con la siguiente técnica:

```
use IO::Tee;
$tee = IO::Tee->new(">>
  archivo.txt", \*STDOUT);
print $tee "un error ocurrió en
  ".scalar(localtime)."\n";
```

El mensaje de la función **print** se imprimirá en **archivo.txt** y en la pantalla al mismo tiempo.

3 Es posible saber la ubicación de un módulo en particular con **perldoc -l Nombre::Modulo**.

4 Para saber qué módulos de PERL necesitan actualización, ejecutamos el comando **sudo perl -MCPAN -eshell** y luego, desde la línea de comandos, tipeamos la letra **r**.

5 Si necesitamos descargar archivos desde un servidor FTP, el siguiente código de ejemplo nos puede ser de gran ayuda:

```
use Net::FTP;

$ftp = Net::FTP
  >new('ftp.server.com');
$ftp->login('usuario',
  'password');
mget($ftp, '*.txt');

sub mget {
  my ($ftp, $pattern) = @_;
  foreach my $file ($ftp
    >ls($pattern)) {
    $ftp->get($file)
      or warn $ftp->message;
  }
}
```

En el ejemplo, estamos descargando todos los archivos **.TXT** del servidor.

# Consejos para la migración a Linux

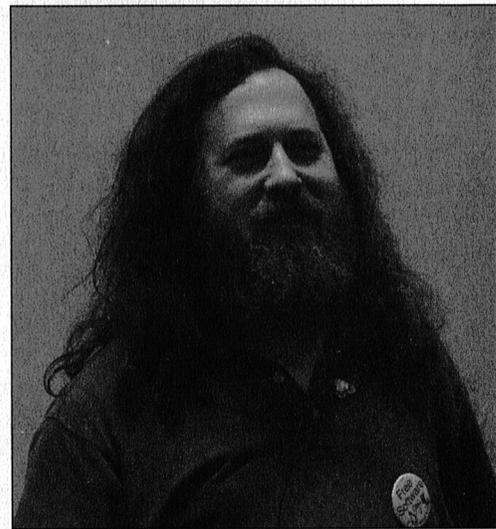
En este capítulo veremos algunos consejos para realizar una migración exitosa a Linux y sus herramientas de Software Libre, tanto en el ámbito doméstico como en el corporativo. Esto incluye planes detallados paso a paso para lograr que el proceso de migración sea lo más rápido y exitoso posible.

Cómo probar Linux sin poner en peligro el sistema actualmente instalado	234
Premisas para elegir una distribución y no cometer el error de elegir la incorrecta	237
Recaudos antes de instalar una distribución de Linux	239
Plan modelo para migrar de Windows a Linux	248
Relevamiento de información del hardware	248
Análisis de compatibilidad de la distribución elegida	249
Actualización del hardware (opcional)	249
Backup de los datos importantes	249
Instalación del nuevo sistema	250
Período de prueba con los dos sistemas operativos	250
Remove el sistema operativo anterior	250
Cómo reemplazar aplicaciones propietarias por sus contrapartes libres	250
Relevamiento de la información del software	251
Análisis de la disponibilidad de las aplicaciones libres	251
Instalación de la aplicación	254
Período de prueba con la nueva aplicación	254
Cómo armar un plan de capacitación efectivo	255
Bonus tips	258

## Cómo convencer a otras personas de probar Linux

¿Por qué pasarse a GNU/Linux? ¿Por qué insistir tanto con el mismo tema? ¿Por qué no dejamos que cada uno viva tranquilo con su sistema operativo? Estas son preguntas que seguramente muchos de nuestros amigos nos formularán cuando les proponamos la migración. Por eso, a continuación veremos varios **argumentos evangelizadores**.

Hay varias cuestiones que hacen de Linux un sistema operativo sumamente interesante. Muchas son de carácter técnico y muchas de carácter filosófico. En primer lugar, comencemos por las de carácter técnico. GNU/Linux es uno de los sistemas operativos más avanzados del mundo. La razón de esto es simple: está basado en UNIX, que tiene un desarrollo de más de 35 años. ¿Qué sis-



**Figura 1.** Se llama evangelizadores a los que se dedican a convencer a otras personas de probar a Linux. Richard Stallman fue el primer evangelista.



Jon Maddog Hall es considerado uno de los más grandes evangelistas de GNU/Linux. Durante los últimos quince años, ha recorrido el mundo brindando charlas evangelizadoras tanto para usuarios como para empresas. Para obtener más información sobre Maddog podemos visitar su sitio en Wikipedia en [http://en.wikipedia.org/wiki/Jon\\_Hall\\_programmer](http://en.wikipedia.org/wiki/Jon_Hall_programmer).

tema tiene más historia? ¿Qué sistema tiene más líneas de código escritas y recritas? ¿Qué sistema tiene más pruebas? La verdad que pocos. Y justamente ésta es la razón principal por la cual **Linux no se cuelga**. En realidad, no es que no se cuelgue jamás, los programas se cuelgan como en cualquier sistema operativo si están mal escritos. Pero el sistema operativo en sí tiene una tasa de **cuelgues** mucho menor a la de cualquier otro sistema, y esto lo demuestran los testimonios de los usuarios y la gran cantidad de estadísticas que hay en Internet. Siguiendo con la cuestión técnica, GNU/Linux es un sistema operativo que **no sufre de virus** porque no los hay. ¿Y cómo puede ser que no haya virus? Es muy simple, no hay virus porque es muy difícil hacer un programa que sea invisible para el usuario. El sistema operativo es totalmente transparente, por lo que cualquiera de nosotros puede saber en cualquier momento qué es lo que está haciendo la computadora. Sí, cualquier persona con un poco de competencia puede hacer un programa que borre todo el disco rígido, pero nosotros lo detectaríamos inmediatamente con sólo pedir un listado de procesos

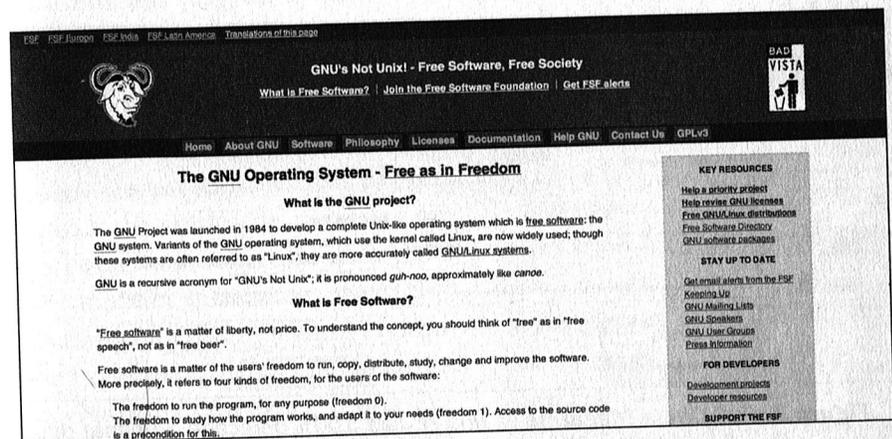
del sistema. Por eso, nadie se molesta en hacer virus para GNU/Linux. Por la misma razón, nadie se molesta en desarrollar software con **spyware** (componentes espías de nuestra información). Y hay más razones técnicas para migrar a GNU/Linux. Una de ellas es que el sistema operativo del pingüino estira la vida útil de cualquier hardware. Ya no estaremos atados a cambiar la computadora una vez por año porque siempre encontraremos una distribución de GNU/Linux que estará optimizada para el hardware que tenemos. De hecho, si es posible instalar GNU/Linux en una computadora Atari de la década de los años 80, entonces podemos instalarlo en cualquier lado.

Las razones que vienen desde otro lado son las filosóficas, que tienen que ver con la libertad, con el hecho de usar software que no fue desarrollado según el interés de las mega corporaciones, sino según el interés de los usuarios mismos. GNU/Linux está desarrollado por usuarios y para usuarios. Si nos interesa saber más sobre esto, podemos visi-

tar el sitio oficial del proyecto GNU (*GNU's Not UNIX*), en [www.gnu.org](http://www.gnu.org) (Figura 2). Pero aún hay una razón más para animarse a probar GNU/Linux. Y para explicarla utilizaremos una analogía. Supongamos que un buen amigo viene con un regalo y nos dice "Éste es mi regalo. Te lo doy porque te aprecio y estoy seguro de que te será de gran utilidad". ¿Qué haríamos? ¿Se lo devolveríamos sin siquiera abrirlo? ¿Sin ver si es realmente de utilidad? ¿Los regalos no se devuelven! ¡Los regalos se agradecen! A lo sumo, podemos abrirlo y probarlo y, si no nos gusta, no lo usamos. Pero siempre le damos una oportuni-

### III LA EVANGELIZACIÓN

Existe un documento maestro sobre la actitud que deben tomar los usuarios de Linux para con el resto de la comunidad. Se llama **Linux Advocacy mini-HOWTO** y podemos descargarlo de [www.faqs.org/docs/Linux-mini/Advocacy.html](http://www.faqs.org/docs/Linux-mini/Advocacy.html).



**Figura 2.** El sitio oficial de la Fundación del Software Libre incluye mucha información sobre la metodología de desarrollo y distribución de Linux.

dad. Bueno, con GNU/Linux sucede exactamente lo mismo. La comunidad del código abierto nos está dando un regalo. Y no nos piden absolutamente nada más a cambio que el uso y que, si nos es de utilidad, animemos a otros a usarlo. Acá no hay manos negras, no hay bolsillos que se llenen de dinero a costa nuestra ni nada por el estilo. Acá hay gente que trabaja por amor al arte, y la mejor forma de retribuirle su trabajo es dándole una oportunidad al producto que genera.

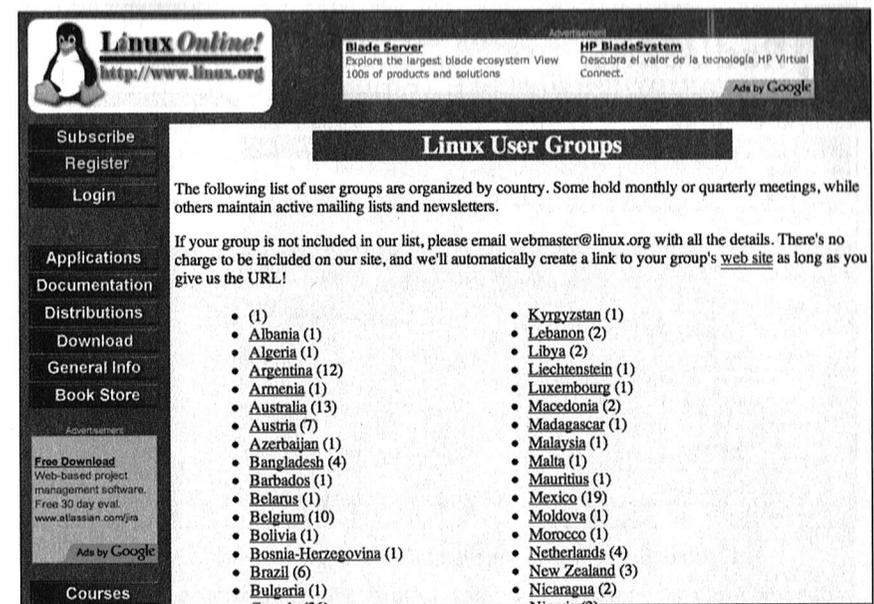
## Fuentes de información indispensables

Antes que nada, siempre es bueno informarse para saber qué esperar del nuevo sistema operativo. Para esto, hay varios recursos a los cuales podemos acudir. En primer lugar, las revistas especializadas nos darán un panorama sobre cómo se está moviendo el

mundo de GNU/Linux actualmente. En general, las revistas nos ofrecen lo mejor de los dos mundos: el de la actualidad (con secciones como noticias y laboratorios de pruebas) y el de la información técnica, con notas y tutoriales para diferentes niveles de usuarios. Otro recurso importante son los sitios de Internet. Hay varios que hay visitar con cierta regularidad para aprender y mantenernos informados. Pero antes de ningún otro, y para entender verdaderamente cómo viene la mano, es recomendable visitar [www.gnu.org/gnu/thegnuproject.es.html](http://www.gnu.org/gnu/thegnuproject.es.html), que no es ni más ni menos que un extenso artículo donde se explican los orígenes del proyecto GNU, su filosofía y hacia dónde va. Luego, podemos visitar [www.linux.org](http://www.linux.org) que, si bien está en inglés, ofrece gran cantidad de artículos y enlaces hacia otros sitios interesantes que seguramente nos serán de gran utilidad. También es recomendable que recorramos la comunidad del código abierto



**Figura 3.** Full Circle Magazine es una revista 100% dedicada a Ubuntu que se distribuye en forma gratuita por Internet en formato .PDF. Está traducida al español por un grupo de colaboradores. Más información en <http://fullcirclemagazine.org>.



**Figura 4.** Linux Online posee el directorio más grande de grupos de usuarios Linux de todo el mundo.

to visitando sitios de LUGs (*Linux USER Group*, o Grupo de Usuarios Linux) y de noticias en nuestro idioma. Para encontrar un LUG cercano a nuestra zona de residencia podemos visitar el sitio [www.linux.org/groups](http://www.linux.org/groups). Allí encontraremos un índice que, aunque no es el más completo del mundo, nos será de gran ayuda.

En cuanto a los sitios de comunidades virtuales, hay varios y muy populares. Barrapunto ([www.barrapunto.com](http://www.barrapunto.com)) es un sitio de noticias en español en el que, día a día, miles de personas entran y comentan sobre los diferentes acontecimientos que tienen algún tipo de relación con GNU/Linux y el mundo del código abierto. VivaLinux! ([www.vivalinux.com.ar](http://www.vivalinux.com.ar)) y Linux Preview ([www.linuxpreview.org](http://www.linuxpreview.org)) son sitios similares, pero con menos comen-

tarios diarios. Igualmente son fuentes importantísimas de noticias relacionadas con el mundo del software libre. Y hablando de comunidad, no podemos dejar afuera al sitio de Foros Linux del portal RedUSERS (<http://redusers.com/foros>). Allí, cientos de lectores de la revista USERS GNU/Linux se juntan todos los días a responder preguntas y ayudarse mutuamente (Figura 5).

## EL SITIO OFICIAL DE LINUX

Linux posee un sitio oficial en la dirección [www.linux.org](http://www.linux.org). Allí podremos encontrar noticias actualizadas con mucha periodicidad sobre la evolución del sistema operativo, recomendaciones de libros y una completa guía de los grupos de usuarios Linux de todo el mundo (Figura 4).

**RedUSERS**  
COMUNIDAD DE TECNOLOGÍA

F.A.Q. | Buscar | Lista de Miembros | Grupos de Usuarios | Regístrate  
Perfil | Conéctese para revisar sus mensajes | Conectarse

Fecha y hora actual: Jue May 31, 2007 10:28 am  
Índice del Foro RedUSERS | Foros

Foro	Temas	Mensajes	Último Mensaje
<b>¡Bienvenidos!</b> Te damos la bienvenida a los foros de REDUSERS.COM. Moderador Moderadores	1	1	Vie Mar 23, 2007 5:58 pm admin *D
<b>Noticias, consultas y sugerencias</b> Noticias en TecTimes, nuestros productos y todos los servicios. Moderadores Moderadores, Redacción	287	2089	Mie May 30, 2007 7:25 pm rivede *D
<b>Revistas</b>			
<b>USERS</b> Computación y tecnología en general. Moderadores Moderadores, Redacción	1409	10992	Mie May 30, 2007 8:33 pm J.Bonc *D
<b>POWER USERS - Hardware</b> Resolvé todas tus dudas sobre Hard. Un foro sólo para fierros. Moderadores Moderadores, Redacción	6448	39736	Jue May 31, 2007 10:20 am luis_rouse *D
<b>LINUX USERS - Linux</b> Sólo para amantes del sistema del pingüino. Moderadores Moderadores, Redacción	2978	12156	Jue May 31, 2007 10:13 am pupipops *D
<b>.CODE - Programación</b> Desarrolladores, analistas, programadores... Moderadores Moderadores, Redacción	1970	7406	Mie May 30, 2007 5:25 pm latengue5 *D
<b>DR. MAX - Principiantes</b> Consultas y consejos, recomendaciones de software. Moderadores Moderadores, Redacción	432	2901	Jue May 31, 2007 8:04 am elpcusers *D

**Figura 5.** Los foros de discusión son un excelente lugar para conocer otros usuarios e intercambiar ideas, soluciones y opiniones.

## Cómo probar Linux sin poner en peligro el sistema actualmente instalado

¡Pruebe antes de comprar! dicen los carteles de los negocios de electrodomésticos por las calles. Bueno, esto es lo mismo. En GNU/Linux podemos probar antes de instalar. ¿Cómo es esto? Sencillo: gracias a un sistema que se denomina **Live CD**.

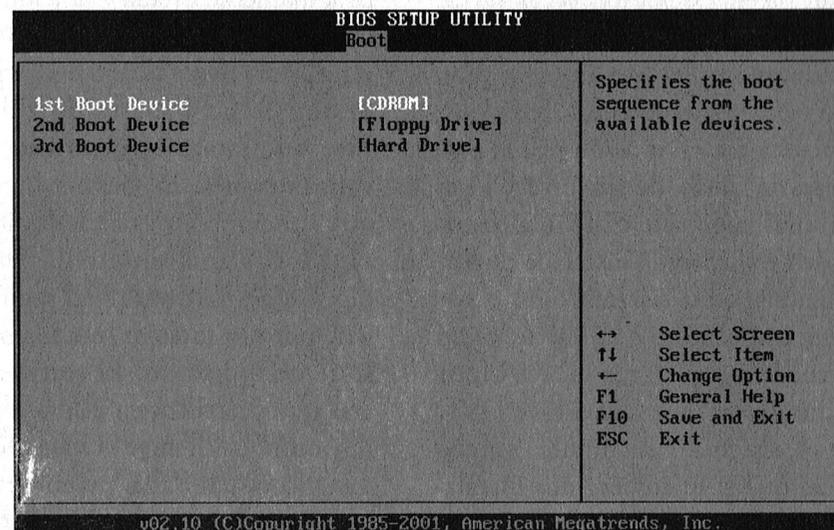
Los Live CD son distribuciones de GNU/Linux diseñadas para funcionar directamente desde una unidad de CDROM o DVDROM. Lo único que tenemos que hacer para que funcionen

es configurar nuestra computadora para que cuando bootee, lo haga primero desde la unidad de CDROM.

Para esto, tenemos que ir al sistema de configuración del BIOS de la máquina y buscar la opción correspondiente, que en general se llama **Boot Sequence** o tiene algún nombre similar (**Figura 6**). Al principio, los Live CD eran distribuciones muy reducidas con poco software para probar. Pero hoy en día, existen distribuciones como **Knoppix** (**Figura 7**) que incluyen todo lo necesario para trabajar con la computadora. Entre otras cosas, incluyen OpenOffice, KDE o GNOME (dos de las inter-

faces visuales más populares de GNU/Linux), el navegador Firefox y mucho más. Los **Live DVD** son exactamente iguales a los Live CD, con la única diferencia de que incluyen mucho más

software instalado y listo para usar. Decimos listo para usar porque todo el software que viene en los Live CDs ya está preconfigurado para que funcione con tan sólo hacer un clic.



**Figura 6.** Selección del primer dispositivo de inicio en el BIOS.



**Figura 7.** Knoppix es una de las distribuciones Live CD de Linux más completas y fáciles de usar.

## COLECCIONES DE LINUX EN LIVE DVD

Así como existen los Live CD, también existen los Live DVD con colecciones de distribuciones Linux en un solo DVD. Podemos visitar [www.livedistro.org](http://www.livedistro.org) para acceder a un completo catálogo de estas distribuciones.



Para saber cuáles son las distribuciones más populares del momento podemos visitar el sitio [www.distrowatch.org](http://www.distrowatch.org), la gran base de datos de distribuciones de GNU/Linux. Allí podremos encontrar noticias sobre las versiones nuevas de las distribuciones, un ranking de las distribuciones más descargadas y las características de cada una de ellas, entre otras cosas.

Otra ventaja de las distribuciones Live CD es que prácticamente no tienen proceso de configuración. En general, las desarrollan de manera tal que funcionen en cualquier hardware por medio de un sistema de autodetección que el usuario ni se entera que está funcionando. Este proceso se lleva a cabo mientras nosotros esperamos que el sistema inicie, y ésa es la razón por la cual el proceso de inicio de un Live CD es un poco más largo que el de una distribución de GNU/Linux instalada de forma tradicional en el sistema.

Algo muy importante que hay tener en cuenta cuando probamos GNU/Linux por medio de un Live CD o Live DVD es que el sistema funcionará mucho más lento que si lo tuviésemos instalado en el disco rígido. Esto es porque la velocidad de acceso a la información de las unidades ópticas como el CDROM es mucho menor a la velocidad que tiene un disco rígido convencional. Por lo tanto, si vemos que nos funciona lento no tenemos que preocuparnos, es simplemente que la lectora de CD o de DVD no es tan rápida como el disco rígido. Algunas distribuciones Live CD recomendadas son:

• **Ubuntu Linux** ([www.ubuntu.com](http://www.ubuntu.com)): sin lugar a dudas, es la distribución más

popular. Puede ejecutarse del CD sin necesidad de instalarlo y posee todo lo necesario para trabajar cómodamente en el hogar y la oficina. Incluye una interfaz visual basada en GNOME que la hace sumamente liviana y funcional en los sistemas de medianos recursos (Figura 8).

• **Knoppix** ([www.knoppix.org](http://www.knoppix.org)): ésta es una de las mejores distribuciones Live que jamás se desarrollaron por varias razones. En primer lugar, porque funciona en casi cualquier computadora actual sin problemas de detección de hardware. En segundo lugar, porque incluye muchísimo software de todo tipo. El software esencial que se utiliza en el día a día está disponible. Incluye OpenOffice, la interfaz visual KDE y cientos de aplicaciones que son accesibles desde el menú de inicio de Knoppix.

• **Slax** (<http://slax.linux-live.org/?lang=es>): Slax es una distribución Live que está basada en Slackware Linux. Para aquellos que no la conocen, Slackware Linux es una de las primeras distribuciones de GNU/Linux, por lo cual tiene muchos años de desarrollo. Recomendamos Slax porque también es muy compatible con el hardware actual, incluye muchas aplicaciones y es realmente muy divertida de probar.

Luego de elegir una distribución, sólo resta descargarla y probarla sin miedo. No podemos romper nada cuando usamos una distribución Live CD o Live DVD, ya que se trata de un medio de sólo lectura. Una vez que nos hayan convencido, entonces podremos pasar al próximo paso: elegir una distribución para instalar formalmente en la computadora.

### Premisas para elegir una distribución y no cometer el error de elegir la incorrecta

¿Por qué hay tantas distribuciones de Linux? ¿No es más fácil que haya una sola? Éstas son preguntas típicas de quienes recién comienzan, porque al tener tanta oferta, no saber cuál elegir.

Empecemos por el principio. La cuestión de que haya tantas distribuciones de Linux reside en que, al ser libre, cualquiera puede hacer su propia versión del sistema operativo. Y cualquiera que haya hecho su propia versión puede distribuirla para que sus amigos puedan usarla. Y éste fue el espíritu inicial del software libre: la **colaboración**. Ahora bien, como mencionamos antes, la diversidad tiene la ventaja de que tenemos más opciones a la hora de elegir, pero tiene la desventaja de que, si no tenemos el suficiente conocimiento, tenemos más probabilidades de elegir una distribución incorrecta. Por eso, a la hora de elegir una distribución, es importante tener en cuenta los siguientes puntos:

- Que sea una distribución popular.
- Que sea una distribución que se actualice con bastante frecuencia.

- Que sea una distribución que resulte fácil de instalar.
- Que sea una distribución que se pueda configurar de forma sencilla.
- Que en el listado de hardware compatible esté todo lo que tenemos en nuestra computadora.
- Preferentemente, que sea libre.

Analicemos punto por punto. El hecho de usar una **distribución popular** nos asegura varias cosas. Las distribuciones populares en general están desarrolladas por grandes empresas que tienen muchos programadores que día a día trabajan para que el producto sea más fácil de usar y más compatible con el hardware existente. Las distribuciones poco populares están desarrolladas por pocos programadores que hacen su mejor esfuerzo pero lógicamente no pueden cubrir todos los aspectos que cubren las grandes compañías. Además, las distribuciones populares son usadas por mucha gente. Esto hace que, a la hora de buscar ayuda, nos sea fácil encontrar gente que esté en la misma situación que nosotros o que ya la haya vivido y sepa cómo solucionar nuestro problema. Por otro lado, al ser distribuciones que usa mucha

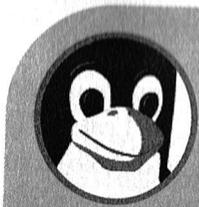
## LINUX EN UN PENDRIVE

También es posible utilizar un pendrive para instalar Linux en él y correrlo sin modificar los datos del disco rígido. El proceso de instalación de una distribución en un pendrive es un poco más complejo que el de grabar una ISO de un Live CD. En [www.pendrivelinux.com](http://www.pendrivelinux.com) encontraremos una colección de tutoriales para la instalación de diferentes distribuciones.

gente, los programadores de aplicaciones se encargarán de armar paquetes de instalación especiales para esas distribuciones. Si tenemos una distribución muy rara, seguramente no tendremos más remedio que bajar el código fuente del programa que queramos y compilarlo nosotros mismos.

El segundo punto, en el que hablamos de la **frecuencia de las actualizaciones**, tiene relación con dos cosas. En primer lugar, con usar un sistema operativo que siempre tenga software actualizado para poder sacar provecho de lo último de la tecnología. Pero en segundo lugar, está relacionado con la seguridad. Una distribución que se actualiza con frecuencia es una distribución que seguramente tiene muchos parches y errores corregidos. Una distribución que se actualiza cada dos días implica un riesgo mayor.

No hay mucho que decir sobre la **facilidad de instalación y de configuración**, pero para asegurarnos esto, con-



Si somos principiantes debemos evitar las distribuciones cuyos sistemas de instalación y configuración sean en modo texto, ya que son más difíciles de usar.

viene instalar distribuciones que estén orientadas al hogar y la oficina. Es recomendable dejar las distribuciones de Linux para servidores para más adelante, cuando tengamos más experiencia.

En cuanto al **hardware compatible**, en general en el sitio oficial de cada distribución se puede encontrar un listado de hardware compatible con la última versión disponible de ésta, separado por categorías.

Por último, recomendamos que sea libre. Usar una **distribución libre** nos asegura que estaremos usando una distribución que no tiene fin en su avance. ¿Quién nos asegura que **Linspire** no se declara en bancarrota el año que viene y cierra sus puertas? Ahora bien,

los proyectos libres no mueren. Cuando un líder lo abandona, uno nuevo lo retoma y continúa su desarrollo.

## Recaudos antes de instalar una distribución de Linux

El proceso de instalación del sistema operativo GNU/Linux difiere bastante del de Microsoft Windows. Pero por suerte, esto no quiere decir que sea más difícil. De hecho, muchas personas consideran que hoy en día instalar GNU/Linux es más fácil que instalar la última versión de Windows. Aunque fácil, el proceso de instalación de GNU/Linux requiere que prestemos atención a algunos puntos importantes para evitar cometer errores y hacer que todo funcione de forma correcta.

En primero lugar, estamos instalando un sistema operativo nuevo en un equipo que seguramente ya tiene un sistema operativo instalado. Esto deriva en que tendremos que modificar la estructura de nuestro disco rígido para que ambos sistemas convivan sin problemas. **Modificar la estructura** es modificar lo que se llama la **tabla de particiones** (lo veremos más adelante), y si bien en la actualidad es un proceso simple, se trata de una tarea crítica ya que está relacionada con el

## III GESTIÓN DE PARTICIONES

GParted (<http://gparted.sourceforge.net/livecd.php>) es un Live CD que incluye un gestor de particiones muy práctico. Podemos utilizarlo para cambiar el tamaño de las particiones y para crear nuevo espacio para el sistema operativo Linux que vayamos a instalar.

nivel más bajo de control del disco rígido, que es nuestra principal unidad de almacenamiento. ¿Toda esta explicación para qué? Para que hagamos **copias de seguridad de todos los datos** importantes antes de instalar GNU/Linux. Hace algunos años, ésta era una tarea tediosa, pero hoy en día gracias a los nuevos medios de almacenamiento como el CD-R y el DVD-R es una cuestión que nos puede llevar unos pocos minutos. Por eso, conviene hacer una copia de seguridad de todo lo que consideremos de importancia, sólo por precaución. Esta recomendación no es para que nos asustemos, ya que no suele haber problemas con las particiones a la hora de instalar GNU/Linux, pero es una cuestión de precaución.

También es bueno tener a mano todos los **manuales de los dispositivos** más importantes de nuestro equipo. Du-

Figura 8. Ubuntu es la distribución más popular de los últimos años.

## III LAS DISTRIBUCIONES MÁS POPULARES

En 2007, las distribuciones más populares reportadas por el sitio <http://distrowatch.com> son (por orden de popularidad): Ubuntu, PCLinuxOS, OpenSuSE, Fedora, MEPIS, Debian, Sabayon, Mandriva, Mint y Damn Small Linux.

rante el proceso de instalación existe un punto de detección y configuración del hardware. Si el sistema no puede detectar nuestra placa de video o nuestro monitor, si tenemos los manuales cerca podremos definir los parámetros a mano. Si no tenemos los manuales, al menos tenemos que averiguar la marca y el modelo de cada uno de los dispositivos que tengamos en el equipo, para luego poder encontrarlos más rápidamente en las listas de hardware del sistema de instalación.

Otra de las recomendaciones es que busquemos el disco de instalación del sistema operativo que tengamos actualmente instalado o que consigamos un disco de inicio de éste. Además, necesitaremos un floppy vacío porque durante el proceso de instalación tendremos la posibilidad de realizar un disco de inicio de Linux que nos servirá de **disco-rescate** si algún día nuestro sistema se rompe y no inicia.

Como última recomendación, conviene que busquemos en Internet experiencias de otros usuarios que hayan instalado la misma distribución que estamos por instalar nosotros.

Estas experiencias las podemos buscar directamente en **Google** utilizando palabras clave como **instalación, Man-**



Los sitios oficiales de los LUGs (Linux Users Group) de nuestra localidad seguramente tendrán listas de correo o foros donde podremos encontrar experiencias de otros usuarios en la utilización de Linux. Además, así podremos conocer usuarios que se encuentran cerca nuestro y armar encuentros o eventos con gente que comparta nuestro gusto por este sistema operativo.

**driva y problemas.** Miles de experiencias, preguntas y soluciones aparecerán gracias a este fabuloso buscador.

Informarse sobre lo que le sucedió a otras personas cuando intentaron instalar la misma distribución nos ayudará a estar preparados para todo, e incluso para decidir cambiar de distribución antes de siquiera perder tiempo intentando instalar la que habíamos elegido porque, por ejemplo, descubrimos que es incompatible con alguno de los dispositivos que tenemos.

Habiendo hecho las recomendaciones correspondientes, vamos a analizar en qué consiste el proceso de instalación, cómo funciona de manera interna y qué es exactamente lo que le sucederá a nuestro sistema durante los 45 minutos que estaremos frente al programa de instalación de GNU/Linux.

## Guía universal de instalaciones domésticas de Linux

El proceso de instalación comienza con el inicio del programa de instalación. Aunque parezca raro, el programa de instalación de GNU/Linux funciona bajo GNU/Linux. Entonces, no podemos iniciar el sistema de instalación desde Windows ya que no se trata de un programa de Windows sino de Linux. La manera más fácil de iniciar el sistema de instalación consiste en ingresar el CD y configurar la computadora para que bootee automáticamente desde esa unidad. En general, todas las computadoras permiten bootear desde una unidad de CDROM y para esto, si el equipo no lo hace automáticamente, lo único que hay que hacer es ir hasta la configuración del BIOS del sistema y definir como pri-

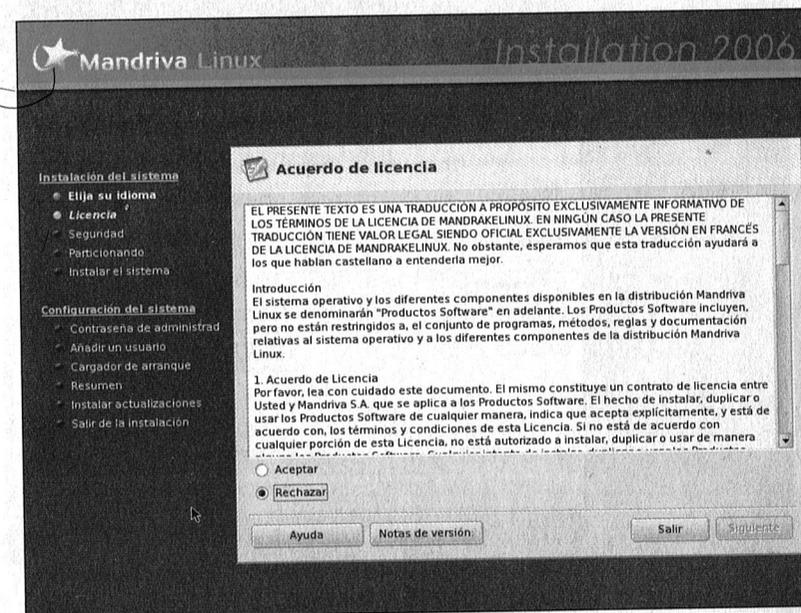
mera unidad de inicio el CDROM. Luego, sólo deberemos ingresar el CD en la lectora, presionar el botón de reinicio de la computadora y listo.

Una vez iniciado el sistema, lo primero que se llevará a cabo es una detección del hardware que tenemos. Dependiendo de la distribución, éste será el momento de configurar el hardware que tenemos, o no. Algunas distribuciones, cuando apenas inician el sistema de instalación, nos preguntan datos básicos como el idioma de la instalación, el tipo de teclado que tenemos, nos presentan la **Licencia Pública General (Figura 9)**, etcétera.

Uno de los puntos más importantes de todo proceso de instalación de GNU/Linux es el momento de las particiones. Para explicar qué es una partición, podemos imaginar el disco rígido de nuestra computadora como una gran torta, que debemos partir en porciones que se

### III SI NUESTRO EQUIPO NO BOOTEA DEL CD...

Podemos utilizar una herramienta llamada Smart Boot Manager (<http://btmgr.sourceforge.net/download.html>). Ésta se graba en un disco floppy para iniciar el equipo con él. Una vez que booteamos con ese floppy se nos presenta un menú en pantalla que nos permitirá bootear desde la unidad de CD o desde cualquier otra unidad que posea nuestro equipo.



**Figura 9.** Casi todas las distribuciones presentan la **Licencia Pública General** antes de comenzar con el proceso de instalación.

les asignarán a los diferentes sistemas operativos que queramos instalar.

En general, si tenemos un solo sistema operativo, la torta será toda para Windows. Ahora, lo que vamos a hacer es crear una nueva partición. Esto es, cortar una porción de la torta para asignársela a GNU/Linux. Lógicamente, esto quiere decir que Windows dispondrá de menos espacio en el disco rígido.

En realidad, GNU/Linux necesita dos particiones para funcionar correctamente. Una de ellas se llama **partición raíz** y será la partición en donde instalaremos todo el sistema operativo y las aplicaciones. Esta partición tiene que tener un mínimo de 2 GB (para tener un sistema completo) y un máximo que el usuario puede definir según sus necesidades. La segunda partición que debemos crear se llama **Linux Swap** (intercambio), y es

un espacio del disco rígido que se utiliza cuando la memoria RAM del sistema se acaba (por ejemplo, cuando cargamos muchos programas en simultáneo). El tamaño de esta segunda partición en general es del doble de la memoria RAM que esté instalada en el sistema. Entonces, si tenemos 256 MB de memoria RAM, una partición de intercambio de 512 MB es lo recomendado.

Hoy en día, crear particiones es un proceso muy sencillo. Muchas distribuciones lo hacen de manera totalmente automática. Sólo debemos definir qué cantidad de megas le vamos a quitar a Windows para asignárselos a GNU/Linux y el sistema se encargará de crear la partición raíz y la partición de intercambio. De todas formas, la mayor parte de los sistemas de instalación permite acceder a complejos sistemas de particionamiento en donde,

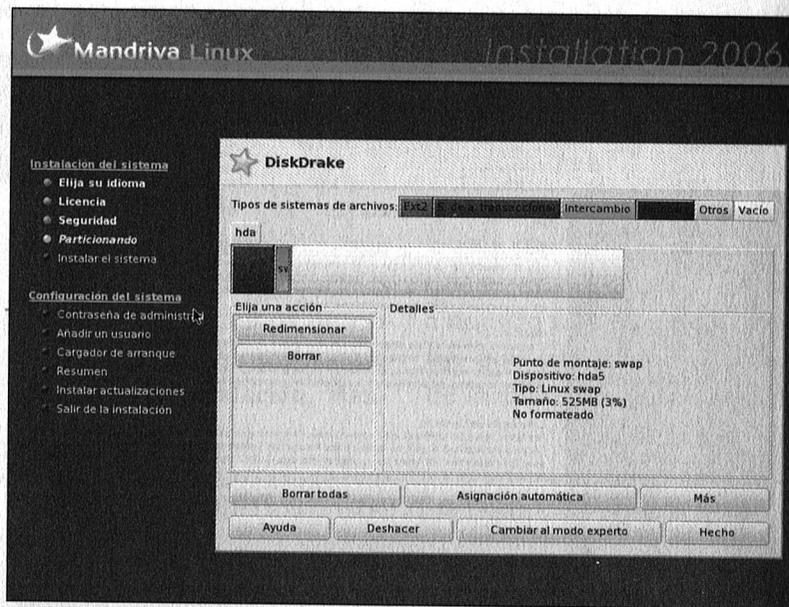
si tenemos las agallas y el conocimiento necesarios, podemos realizar un control más minucioso de la creación y el redimensionamiento de las particiones.

Una vez que el espacio para el nuevo sistema operativo se ha creado, el próximo paso consiste en seleccionar qué paquetes de aplicaciones van a instalarse. Es recomendable no escatimar: si hay recursos (esto es, espacio en el disco rígido), conviene instalar todo lo que podamos. Esto nos ahorrará trabajo más adelante y nos permitirá conocer más aplicaciones del mundo del código abierto. En cambio, si no hay espacio suficiente, tenemos que instalar sólo lo que necesitamos pero sin olvidarnos de instalar los **paquetes de desarrollo de aplicaciones** y la **documentación**. Los primeros serán necesarios a la hora de instalar programas que requieren ser compilados (esto es, cuando descargamos un programa en formato de código fuente). La segunda nunca estará de más para aprender nuevos comandos o sacarnos dudas sobre la utilización de algún parámetro. El proceso de copia de los paquetes a nuestro equipo puede tomar entre media hora y dos horas, dependiendo de la cantidad de aplicaciones que hayamos elegido para instalar y de la velocidad de nuestro equipo.



Existe una versión de Linux llamada ZipSlack que se instala en un directorio cualquiera de una partición DOS/Windows y se puede ejecutar con tan sólo tipear un comando de DOS. Para obtener más información sobre ella, podemos visitar su sitio en [www.slackware.com/zipslack/](http://www.slackware.com/zipslack/). Lo bueno de ZipSlack es que nos permite disfrutar de Linux sin tener que formatear el disco.

Una vez que todo el sistema está instalado, algunas distribuciones nos permiten controlar la configuración del hardware y nos informan si no pudieron instalar algún dispositivo de forma correcta. Éste es el momento de consultar los manuales de nuestros dispositivos y configurar cuanto podamos. El último paso del proceso de instalación del sistema operativo consiste en la instalación de lo que se denomina **gestor de arranque**. Este programa es el que nos permitirá iniciar el sistema operativo y se instala en la **MBR** (sector de arranque del disco rígido). El gestor de arranque nos permitirá, además, seleccionar el sistema operativo que queramos iniciar por medio de un menú que se presentará en nuestra computadora justo después de presionar el botón de encendido.



**Figura 10.** Las herramientas de instalación de las distribuciones más populares nos permiten reparticionar el disco rígido mediante la modificación de un gráfico que representa su estructura.

## III FUTURAS MODIFICACIONES A LAS PARTICIONES

Si más adelante necesitamos modificar las particiones, cambiar el tamaño de la partición de Windows o de Linux o borrar alguna, podremos utilizar una herramienta libre para GNU/Linux llamada **QtParted** (<http://qtparted.sourceforge.net>). Ésta es una aplicación gráfica que nos permitirá gestionar nuestras particiones de manera rápida y sencilla.

El proceso de instalación del gestor de arranque es automático en la mayor parte de las distribuciones. Y si se nos consulta por la creación de un disco de inicio de Linux, conviene que lo hagamos. Después de todo esto, sólo es cuestión de reiniciar el equipo e ingresar a la nueva distribución desde el menú de inicio de sistemas operativos que aparece en pantalla.

## Cómo convencer al jefe de migrar a Linux

Convencer al CEO, al gerente de sistemas o al administrador de turno es, en realidad, explicar los beneficios de la manera más simple y concisa posible. Por eso, el siguiente discurso será de gran utilidad a la hora de presentar un proyecto de migración a Linux.

Cuando las compañías se plantean cuáles son los beneficios de utilizar productos de código abierto, se encuentran a primera vista con el famoso tema de que no hay que pagar licencias. Eso es verdad y es una ventaja que en algunos casos no puede hacer ahorrar miles de dólares. Pero, el principal beneficio no está ahí. De hecho, muchas compañías tienen la suerte de no seleccionar sus sistemas por el valor de las licencias sino por el valor del producto como un todo (y esto incluye el sistema, los servicios relacionados, el soporte, etcétera). El principal beneficio de utilizar herramientas de código abierto en nuestra empresa es la independencia tecnológica. Al utilizar software de código abierto, no estamos atados a ninguna compañía en particular. Y para explicar esto, es bueno recordar el ejemplo que da Jon Hall (presidente de Linux International) en sus charlas (Figura 12).

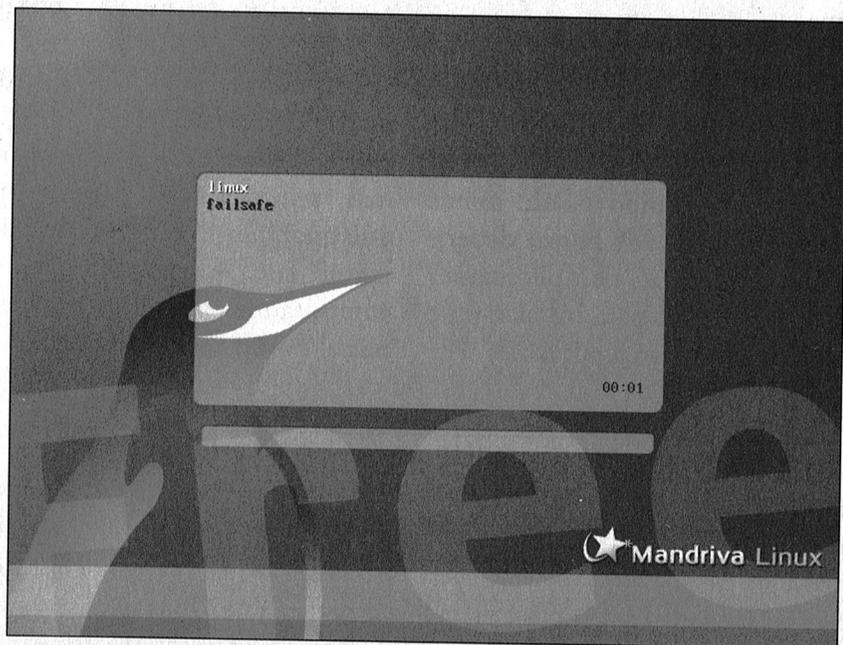


Figura 11. Menú de inicio del sistema operativo Mandriva Linux.

**LINUX INTERNATIONAL** Linux® International

User login  
 Username:   
 Password:

Linux International, also known as LI, is a world-wide non-profit association of end users who are dedicated to furthering the acceptance and use of Free and Open Source Software (FOSS). An end-user is defined as a person that uses FOSS in their daily life. Linux International is not a vendor organization, so while it may have sponsoring bodies who are corporations or companies, voting membership is limited to individuals, and individual representatives of international chapters and international affiliates.

**Linux International Announces New Direction**  
 Submitted by admin on Mon, 2006-01-30 20:01.  
 Linux International, a vendor organization since 1994 which has helped to promote the Linux Operating System, announces their new charter of being an end-user organization for Free and Open Source Software (FOSS).  
 This transformation, which has been underway for six months, is almost complete, and this web site will be changing and expanding rapidly over the next Month (February) to embrace this new mission. Therefore potential users should be warned that accounts created before or during this time period might disappear.  
 (categories: )

Events  
 May 2007  

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

 Upcoming events  
 Who's new  
 admin  
 Who's online  
 There are currently 0 users and 135 guests online.

The registered trademark Linux® is used pursuant to a license from Linus Torvalds, owner of the mark in the U.S. and other countries.  
 All content on this site © 2006 by each individual author, All Rights Reserved.  
 This site is powered by CivicSpace.

Figura 12. Linux International es una organización que se dedica a difundir y evangelizar sobre las ventajas de Linux para usuarios finales y negocios.

Hace algunos años se detectó una vulnerabilidad importante que afectaba a la mayor parte de los navegadores de Internet. La vulnerabilidad era muy seria y, entre otras cosas, permitía que cualquier persona que tenga algunos conocimientos avanzados de seguridad pudiera acceder a sitios seguros con encriptación sin tener siquiera usuario y password. Bien, dado que la vulnerabilidad debía ser resuelta lo más rápido posible, el parche para el navegador Konqueror (el navegador oficial del proyecto KDE) apareció algunas horas después de que se detectara la vulnerabilidad. El proyecto Mozilla por su parte, lanzó el parche algunas horas más tarde. Finalmente, el parche para el navegador de Internet propietario más famoso, apareció casi una semana después.

En resumen, los productos de código abierto se corrigieron el mismo día en el que se detectó la vulnerabilidad, mientras que los usuarios de productos propietarios tuvieron que vivir en la oscuridad durante varios días.

Problemas como el de este ejemplo ocurren con bastante frecuencia y son una muestra del dinamismo y la independencia de los que gozan los proyectos de código abierto. Supongamos que en un departamento de una compañía se utiliza un software que cierto día comienza a fallar. Si el programa es propietario, entonces habrá que contactarse con la empresa que lo desarrolla y pedirle que desarrolle un parche para corregirlo. Eso puede llevar un día, dos o diez meses. Y tendremos que esperar. Imaginemos que, si la compañía es medianamente



En otro de los campos en los que se puede ver cómo Linux ha avanzado mucho más en comparación con otros sistemas operativos es en el de los entornos 3D. Mientras que OS X y Windows Vista han incorporado algunas características 3D muy sutiles a su interfaz, el proyecto Compiz Fusion ya brinda al mundo de Linux un entorno totalmente 3D que es muy atractivo.

Es importante que, mientras se entrevista a los empleados para realizar el proceso de auditoria, se les comunique claramente cuál es el objetivo de la migración, cuáles son las ventajas, los desafíos y los problemas con los que se encontrarán. La transparencia de la comunicación es esencial para lograr que los empleados colaboren y se comprometan con el proceso de migración.

- Necesidad de expandir la capacidad del hardware de las computadoras.
- Aplicaciones abiertas que se pueden utilizar en ese departamento.
- Disponibilidad local de programadores que puedan brindar servicios de soporte y extensión de funcionalidades del sistema de código abierto.
- Actualizar los sistemas de almacenamiento.
- Necesidad de brindarles seminarios a los empleados para convencerlos de que el cambio será para mejor.
- Necesidad y tipo de capacitación que necesitarán los empleados para acostumbrarse al nuevo sistema.
- Estructura de soporte que se les brindará a los empleados.
- Tiempos que llevarán las modificaciones como la actualización del hardware, la instalación de los nuevos sistemas y la capacitación del personal.
- Desarrollo de manuales de uso y solución de problemas para los usuarios.

En cuanto al último punto, esto nos da el pie para hablar de un paso muy importante en el proceso de migración. Todo

ción al cliente, área de ventas, etcétera. Luego debe realizarse un proceso de auditoria en cada uno de esos departamentos. En el proceso de auditoria, el auditor debe prestar especial atención al relevar los siguientes datos:

- Cantidad de computadoras del departamento.
- Cantidad de aplicaciones promedio que utilizan los empleados.
- Tipo de aplicaciones que se utilizan en el departamento.
- Sistema de almacenamiento que se utiliza en el departamento.
- Sistema de copias de seguridad que se utiliza en el departamento.
- Perfil de los empleados.
- Potencial de adaptación al nuevo sistema de cada empleado.
- Voluntad de adaptación al nuevo sistema de cada empleado.
- Problemas que los empleados denuncian que tienen con el sistema actual.
- Capacidad del hardware de funcionar con el nuevo sistema.

Y el listado puede extenderse a cualquier campo que el auditor considere necesario. Éstos son sólo algunos de los puntos importantes del relevamiento de información para generar un plan de migración acorde a cada departamento. El plan de migración total de la compañía es, en realidad, la suma de micro planes de migración de los diferentes departamentos.

Luego de obtener todos los datos que mencionamos anteriormente, el auditor encargado de generar el plan de migración puede deducir otros aspectos como los siguientes:

programador capacitado para que modifique y adapte el programa al funcionamiento de la empresa.

La independencia tecnológica es el principal beneficio de utilizar código abierto. El hecho de poder disponer del programa como queramos para adaptarlo, modificarlo, mejorarlo y actualizarlo es, simplemente, el valor más importante de los productos de código abierto. Y por eso, las grandes compañías ya entendieron que el código abierto es el futuro y los productos libres aparecen en los primeros puestos de predilección en los listados de implementación de nuevos sistemas.

### **Cómo llevar adelante un proceso de auditoria exitoso**

En una compañía, lo primero que debe hacerse cuando se plantea la migración de sus sistemas a productos de código abierto es dividir el parque de sistemas en departamentos, aun cuando esto no esté hecho realmente en la empresa. Es bueno visualizar los departamentos en un mapa y entonces ponerle un nombre a cada uno como área. Por ejemplo, área de contabilidad, área de documentación, área de sistemas, área de informa-

grande, tendrá tiempos mayores ya que la situación que le reportamos nosotros con seguridad se habrá repetido en otras compañías, quizá con otros sistemas. En cambio, si la herramienta que utilizamos es de código abierto, podemos llamar a la compañía que nos brindó el servicio de instalación y pedirle que solucione el problema. Si la compañía nos ofrece un servicio que no nos satisface (como un plazo de treinta días para solucionar el problema), entonces tenemos toda la libertad de llamar a otra compañía para tener otra propuesta de solución diferente.

Lo mismo sucede con las funcionalidades de un programa. Si compramos un programa cerrado, estamos comprando un grupo de funcionalidades que muy difícilmente puedan extenderse para adecuarse aún más al funcionamiento de nuestra compañía. Podemos llamar a la compañía que desarrolla el programa propietario y decir que necesitamos tal o cual funcionalidad, pero seguramente recibiremos un no como respuesta ya que, si la compañía tiene 10.000 clientes, es imposible que agregue miles de funcionalidades pedidas por todos esos clientes. En cambio, si utilizamos programas de código abierto, podemos contratar a cualquier

### **EL PORTAL DE LINUX CORPORATIVO**

Uno de los sitios de referencia de la aplicación de Linux y otras herramientas de software libre en el ámbito corporativo es Linux.com ([www.linux.com](http://www.linux.com)). Allí encontraremos noticias, revisiones de software y casos de éxito que nos servirán de ejemplo a la hora de llevar a cabo la tarea de evangelización.

movimiento, desde que se tiene la primera reunión con una compañía proveedora de servicios de sistemas hasta que se instala el último programa del proceso de migración, debe ser documentado. Todo debe quedar archivado con fecha, hora, nombres de los responsables, firmas, etcétera. Por eso, hasta ahora generamos dos informes por departamento: el primero de ellos se llama Informe de auditoría y es, básicamente, el resultado del relevamiento que se hizo en el departamento. Allí se especifican todos los puntos que detallamos más arriba y se suman conclusiones personales del auditor. Con este informe se genera el Informe de implementación, que es un detalle de los puntos que se tratan en el plan de migración. Una vez que estos dos informes estén realizados y pulidos podremos pasar al próximo paso: el desarrollo del paso a paso de la migración.

## Plan modelo para migrar de Windows a Linux

Cambiar el sistema operativo implica un cambio en los paradigmas operativos de gran parte de nuestra compañía. Si bien es importante y peligroso, si lo hacemos de manera estructurada (y sin apuros) todo saldrá según lo planeado.



Nuevamente, buscar experiencias de migración en otras compañías nos será de gran ayuda para evitar problemas inesperados. Una simple búsqueda en Google nos ofrecerá un sinfín de sitios web que documentan las experiencias de migración de cientos de compañías en todo el mundo. Así, evitaremos pasar por los mismos problemas o al menos podremos preverlos y tener la solución.

Aquí, el auditor tendrá que tener en cuenta diferentes puntos aislados tales como la capacidad del hardware de adoptar el nuevo sistema operativo, los medios de backup de la información actual a utilizar, los períodos de prueba y más. Veamos una guía paso a paso.

## Relevamiento de información del hardware

Lo primero que se debe hacer para poder seleccionar una distribución es analizar las ofertas del mercado en cuanto a las versiones de Linux posibles para instalar en los diferentes equipos de los distintos departamentos. Seleccionar la distribución según el hardware es un error por que seguramente terminaremos eligiendo una distribución que quizá no sea óptima para la operativa humana. De todas

formas, el hardware es un punto muy importante y debemos tratar de equilibrar nuestra decisión entre las distribuciones que mejor funcionen en nuestro hardware y las que nos ofrezcan mejores herramientas operativas.

## Análisis de compatibilidad de la distribución elegida

Como mencionamos antes, una vez que tenemos elegida la mejor distribución que se adecúa a la operativa del departamento de nuestra compañía, podemos proceder a analizar si es compatible con el hardware que tenemos. En caso de que no lo sea, existe la posibilidad de actualizar el hardware o de buscar otra distribución.

## Actualización del hardware (opcional)

Las actualizaciones de hardware nunca vienen mal. De hecho, debemos tomar-

lo como una inversión en la compañía. En el 90% de los casos, las actualizaciones de hardware optimizan el desempeño de nuestros empleados. Por eso, si la distribución que queremos usar necesita más recursos, podemos analizar la opción de actualizar el hardware como una buena oportunidad de obtener mejores resultados con todo el proceso.

## Backup de los datos importantes

Antes de realizar cualquier tipo de modificación en el sistema informático, es absolutamente necesario realizar backups de los datos de los equipos. En muchos casos, con copiar documentos y planillas a un CD es suficiente. En otros, también es necesario hacer réplicas de las bases de datos. Todo este proceso puede ser muy sencillo o muy complejo si hablamos de cientos de máquinas. Es bueno tener en cuenta la posibilidad de generar imágenes de los discos

## BASE DE DATOS DE COMPATIBILIDAD DE HARDWARE

Phoronix ([www.phoronix.com](http://www.phoronix.com)) es un sitio en el que la gente escribe revisiones de hardware en Linux. Allí encontraremos reportes, enlaces a drivers y soluciones a problemas de todo tipo de dispositivo de hardware.

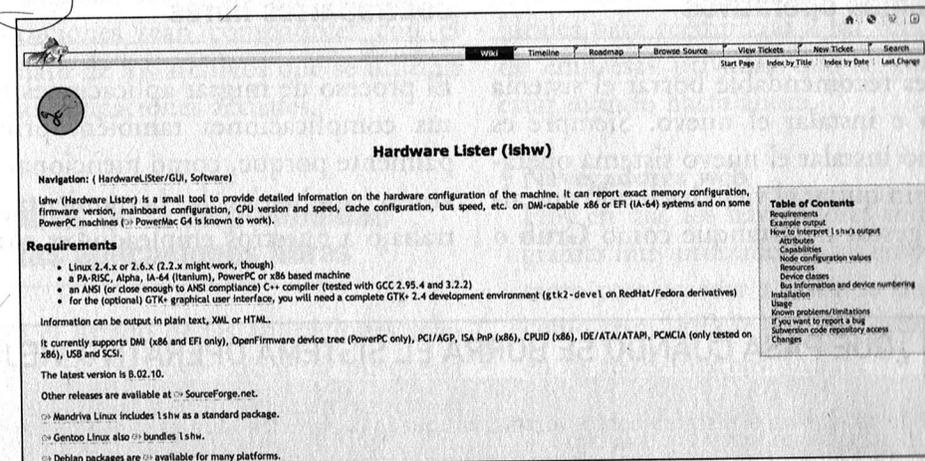


Figura 13. Hardware Lister (<http://ezix.org/project/wiki/HardwareLlster>) es una herramienta que genera un informe del hardware de fácil navegación.

rígidos de los sistemas. Éstas son muy útiles ya que, en caso de tener problemas, se puede tener el sistema original restablecido en cuestión de minutos.

## Instalación del nuevo sistema

Una vez que se realizó el backup de todo lo importante, el hardware está listo y la distribución está en nuestras manos, podemos proceder a instalar el sistema operativo. Si tenemos muchos equipos, podemos montar un servidor NFS que contenga los CDs de la distribución e instalar varias computadoras al mismo tiempo. Sería bueno que todas las distribuciones detectaran todo nuestro hardware sin problemas. Lamentablemente, esto no es así, por lo que será necesario tener en cuenta los tiempos extras que tome la configuración del hardware, la búsqueda de los drivers, etcétera.

## Período de prueba con los dos sistemas operativos

No es recomendable borrar el sistema viejo e instalar el nuevo. Siempre es bueno instalar el nuevo sistema operativo sin quitar el viejo haciendo que algún gestor de arranque como **Grub** o

**LILO** nos permitan seleccionar cuál de los dos sistemas operativos bootear cuando se inicia la computadora. Luego de un tiempo prudencial, y cuando ya tenemos certeza de que el departamento funciona sin problemas con el nuevo sistema operativo, podemos desinstalar el viejo.

## Remover el sistema operativo anterior

Remover el sistema operativo anterior es un proceso relativamente simple, pero no es bueno borrar todos los sistemas del parque. La eliminación debe ser paulatina y siempre teniendo en cuenta la posibilidad de que las cosas puedan salir mal o que se necesite restablecer el sistema anterior para alguna actividad en particular.

## Cómo reemplazar aplicaciones propietarias por sus contrapartes libres

El proceso de migrar aplicaciones trae sus complicaciones también, principalmente porque, como mencionamos antes, cambiarles la herramienta de trabajo a nuestros empleados es como

cambiarles los métodos y las formas de trabajo. Veamos en qué pasos podemos estructurar el proceso para que sea lo más suave posible.

## Relevamiento de la información del software

Lo primero que debemos hacer es un informe detallado de todas las aplicaciones que utilizan frecuentemente los empleados de los departamentos que vamos a migrar. Durante el relevamiento, es necesario que el auditor tenga la pericia suficiente para discriminar cuáles de las herramientas que se utilizan son necesarias y cuáles no tanto. Los usuarios suelen decir que necesitan ciertos tipos de aplicaciones como sistemas de chat, mensajería instantánea y descarga de archivos, cuando en realidad no los necesitan.

Es importante analizar también qué versiones de las aplicaciones se están utilizando y en qué formato de archivos trabajan estas aplicaciones. Es absolutamente necesario que las nuevas aplicaciones sean compatibles con el formato de los archivos que se utilizan en las aplicaciones actuales.

## Análisis de la disponibilidad de las aplicaciones libres

Con el listado de las aplicaciones relevadas, podremos realizar la búsqueda de las alternativas disponibles en el mundo del código abierto. Para eso, debemos tener en cuenta los requerimientos de las aplicaciones que esta-



En la actualidad, existen muchas alternativas de software libre por cada aplicación propietaria. Por eso es muy importante que, antes de decidirnos por una, hayamos probado a fondo todas las aplicaciones similares para asegurarnos de estar eligiendo la que mejor funciona en nuestro sistema y la que más se adecua a nuestras necesidades, preferencias y forma de trabajo.

mos seleccionando, el tipo de licencia, el historial de desarrollo que tienen (con qué frecuencia se generan parches para errores, qué actividad hay en el grupo de desarrolladores, etcétera). Otro factor importante para tener en cuenta es la difusión local que tiene la aplicación. Es importante darles prioridad a las aplicaciones que sabemos que tienen un grupo nutrido de usuarios o desarrolladores en la zona en la que estamos para así tener mayores probabilidades de tener soporte a la hora de solucionar problemas. Veamos un resumen de las aplicaciones de uso cotidiano que pueden ser ideales para reemplazar a las versiones de empresas privadas que se pueden estar usando hasta ahora.

### • Navegadores web

Hoy en día, los navegadores son programas muy utilizados en las empresas, tanto para acceder a sitios de Internet como para utilizar portales exclusivos en las redes internas (intranet).

– **Firefox** ([www.mozilla.org/products/firefox](http://www.mozilla.org/products/firefox)): Firefox es el navegador web de código abierto más popu-

## III ¿QUÉ PASA CUANDO SE BORRA EL SISTEMA OPERATIVO VIEJO?

Cuando se elimina el sistema operativo anterior, queda la partición libre para que podamos utilizarla. Entonces se puede usar alguna herramienta como **GParted** o **QTParted** para asignarle ese espacio libre a la partición de uso cotidiano en Linux.

lar del momento. Es altamente compatible y su desarrollo se lleva a cabo con gran velocidad.

– **Konqueror (www.konqueror.org)**: se trata del navegador oficial del proyecto KDE. Es muy veloz y estable, aunque no es tan compatible como Firefox.

– **Opera (www.opera.com)**: no es código abierto pero es gratuito. Es veloz y tiene características que ningún otro navegador ofrece, como un cliente de BitTorrent integrado (Figura 14).

### Cientes de correo electrónico

Muchas de las operaciones de comunicación de las empresas se realizan por medio de e-mails, por lo que una aplicación de este tipo es de suma importancia para el correcto funcionamiento de la compañía.

– **Evolution (www.gnome.org/**

**projects/evolution)**: es considerado por muchos la copia más exacta de Microsoft Outlook.

– **Thunderbird (www.mozilla.org/products/thunderbird/)**: es el cliente de correo electrónico oficial del proyecto Mozilla. Ofrece muchas prestaciones y es muy estable.

– **Kmail (http://kmail.kde.org/)**: es el cliente oficial del proyecto KDE. No ofrece tantas funcionalidades como Thunderbird y Evolution, pero es muy veloz.

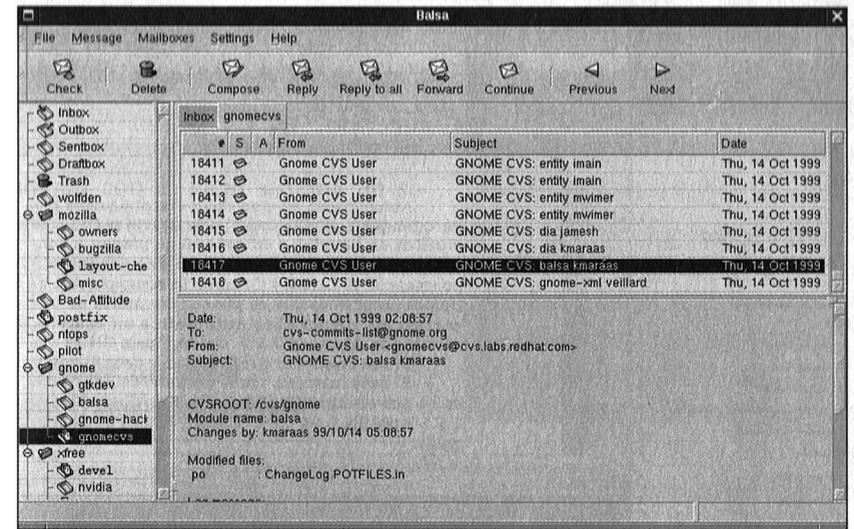
– **Spruce (http://spruce.sourceforge.net)**: es una versión reducida de Kmail, que es mucho más liviano.

– **Balsa (http://balsa.gnome.org)**: se trata de un cliente de correo electrónico del proyecto GNOME que tiene un look muy similar al del clásico Eudora (Figura 15).

### Herramientas de oficina



**Figura 14.** Opera es un excelente navegador, sumamente compatible con las actuales aplicaciones web 2.0



**Figura 15.** Balsa es un cliente de correo electrónico sumamente liviano, ideal para los sistemas con bajos recursos.

Para llevar adelante una empresa, se utilizan documentos de texto, planillas de cálculo y otro tipo de elementos. Por eso, es indispensable conseguir aplicaciones que permitan crearlos y también compartirlos con los usuarios que utilicen los productos más comunes del mercado.

– **OpenOffice (www.openoffice.org)**: la suite de oficina por excelencia. Ofrece procesador de texto,

planilla de cálculo, sistema de diseño vectorial, presentaciones, editor web, base de datos y más. Todo compatible con Microsoft Office.

– **AbiWord (www.abiword.org)**: excelente procesador de texto compatible con Microsoft Word. Es muy liviano, lo que lo hace ideal para los sistemas de bajos recursos en los cuales OpenOffice funciona lento (Figura 16).

– **Gnumeric (www.gnumeric.org)**: ésta es una excelente planilla de cálculo compatible con el formato de archivos de Microsoft Excel.



Recordemos que también podemos probar las herramientas de oficina Web 2.0 (si tenemos una conexión a Internet de Banda ancha) como Google Docs & Spreadsheets (http://docs.google.com). Éstas ofrecen compatibilidad con archivos de MS Office y funcionan muy bien en las computadoras con bajos recursos, ya que todo el procesamiento se lleva a cabo en los servidores.

### Aplicaciones de servidor

En las empresas se ahorran recursos mediante la centralización de éstos en servidores que permiten que todos los usuarios de la red compartan y utilicen, por ejemplo, la misma impresora. Veamos aplicaciones típicas para implementar en una red con servidor.

– **CUPS - Servidor de Impresión**

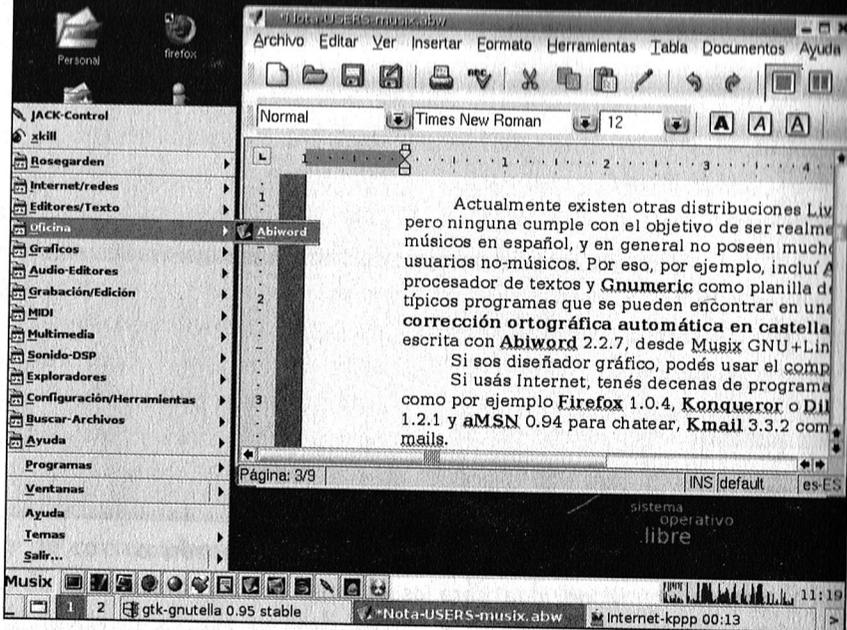


Figura 16. AbiWord es uno de los mejores procesadores de texto de código abierto.

([www.cups.org](http://www.cups.org)): es el mejor servidor de impresión hasta el momento. Soporta una gran cantidad de impresoras y es compatible con los sistemas Linux, Mac y Windows.

- **Samba – Servidor de Archivos** ([www.samba.org](http://www.samba.org)): Samba permite que los recursos alojados en un equipo Linux sean visibles desde un equipo Windows y viceversa.
- **Postfix – Servidor de Correo Electrónico** ([www.postfix.org](http://www.postfix.org)): servidor de correo electrónico estable, robusto y sobre todo, muy fácil de instalar.

Como vemos, hay opciones para todo y mucho más. Éstas han sido sólo algunas recomendaciones. Para más información podemos visitar Linux Migration ([www.linuxmigration.com/quickref/](http://www.linuxmigration.com/quickref/)), un sitio que contiene un listado de aplicaciones útiles a la hora

de realizar la migración.

## Instalación de la aplicación

En este punto, lo más importante es que, al igual que en la instalación del sistema operativo, la instalación de las aplicaciones no debe ser masiva en todas las computadoras al mismo tiempo. El proceso debe ser paulatino y con constantes controles de compatibilidad entre la nueva aplicación y la vieja, para asegurarnos de que todo vaya a funcionar bien.

## Período de prueba con la nueva aplicación

El período de prueba del desempeño de los empleados con la nueva aplicación puede llevar entre una semana y un par de

meses, según la magnitud de la empresa y la cantidad de empleados. No es mala idea que, durante un período de tiempo prudencial, tengamos algunos empleados que trabajen con el viejo sistema en simultáneo con algunos empleados que ya estén trabajando con el nuevo.

## Cómo armar un plan de capacitación efectivo

El plan de capacitación es muy importante porque nuestros empleados deben acostumbrarse al nuevo entorno de trabajo. Estos planes de capacitación deben contemplar tiempos, material, ejercitación, períodos de adaptación y sistemas de soporte.

## Armado del programa

El programa de capacitación debe contemplar las tareas que realizan los diferentes departamentos de nuestra compañía. Por ejemplo, si el departamento contable utiliza exclusivamente planillas de cálculo, entonces debemos armar una especie de curso sobre la utilización del software de planilla de cálculo que vamos a instalar en Linux. Pero, al mismo tiempo, debemos capacitar a los empleados contables en la operación general del nuevo sistema operativo.

## Dictado de clases

Mucha gente deja este punto para cuando el proceso de migración ha finalizado, pero existe la posibilidad de que el proceso de capacitación se reali-

## WINDOWS / LINUX

En [http://wiki.linuxquestions.org/wiki/Linux\\_software\\_equivalent\\_to\\_Windows\\_software](http://wiki.linuxquestions.org/wiki/Linux_software_equivalent_to_Windows_software) encontrarán un completo listado de equivalencias entre las aplicaciones del mundo de Windows y del mundo de Linux. El listado es actualizado en tiempo real por los visitantes, por lo cual está en constante expansión.

ce en paralelo con el de migración. Podemos enviar a nuestros empleados a un instituto externo si aún no tenemos los equipos preparados con el nuevo software para trabajar.

## Evaluación de los alumnos

Quien gestione el plan de migración, después de capacitar a los empleados, debe evaluarlos y archivar los resultados de las evaluaciones para poder sacar conclusiones en cuanto a la capacidad de adaptación al nuevo sistema. Las evaluaciones pueden ser prácticas, teóricas o una mezcla de ambas.

## Seguimiento de la actividad

Una vez que la migración esté hecha, es absolutamente necesario implementar un plan de seguimiento de la operativa de los empleados con el nuevo sistema durante un período de tiempo que puede oscilar entre 10 y 30 días. El objetivo de hacer esto es detectar los posibles problemas de operativa que pueda generar la migración.

## Generación de documentos le soporte

Finalmente, el punto más importante es que los empleados tengan a quien acudir cuando tengan un problema. Más allá de que haya un administrador de sistemas que pueda ayudarlos, es necesario que haya un **kit de documentación** desarrollado específicamente para cada departamento, en el que se especifique a utilización y el mantenimiento tanto del sistema operativo como del conjunto de herramientas utilizados.

## ¿Cuánto tiempo es recomendable tener los dos sistemas operativos?

A lo largo de todo este capítulo hemos intentado destacar numerosas veces que la migración de Windows a Linux no debe hacerse de manera brusca de un día para el otro. La tecnología actual nos permite tener los dos sistemas operativos instalados en simultáneo, y es

mejor que los equipos estén así por un tiempo prudencial. Ahora bien, ¿por cuánto tiempo? Lo que seguramente sucederá es que los primeros días las personas se encontrarán en problemas para llevar a cabo determinadas tareas. En esos casos, es recomendable que utilicen Windows si y sólo si la tarea requiere solución con extrema urgencia. En caso contrario, es una excelente oportunidad para que el usuario investigue, se aclimate al sistema operativo y descubra el problema por sí mismo (o alguien con un poco más de conocimientos le explique cómo llevar a cabo la tarea). De esta forma, la próxima vez que aparezca un problema similar, ya no tendrá que ingresar a Windows. Por eso, una vez que se hizo la implementación de Linux, hay que pensar en la opción Windows como un salvavidas para cuando las cosas salgan demasiado mal. Si no hay problemas, es recomendable dejarlo instalado durante algunas semanas o incluso meses si hay espacio en disco y las licencias lo permiten. Esto

les dará cierta seguridad a los administradores y a los usuarios. Luego, cuando esté probado y demostrado que todas las tareas se pueden llevar a cabo sobre Linux sin problema, se puede prescindir totalmente de la partición de Windows.

## En resumen...

Si hay algo que queda demostrado en este capítulo es que el proceso de migración a Linux no es una tarea que pueda llevarse a cabo a las apuradas. Lo más importante de todo es informarnos sobre los problemas que puedan llegar a ocurrir. Por eso, en este capítulo hemos brindado algunos trucos que seguramente serán de gran utilidad, pero lo cierto es que la mejor enseñanza es la experiencia propia.

Si tenemos computadoras de marca (como IBM, HP, Acer, Dell, etcétera) corremos con la ventaja de poder encontrar con más facilidad los problemas específicos que pueda llegar a tener la distribución que estamos intentando instala-

lar. Sólo tenemos que entrar a Google y utilizar como cadena de búsqueda una combinación de palabras que incluya la marca y el modelo de la computadora junto con el nombre y el número de la versión que pensamos instalar.

Por último, sería muy importante que, luego de haber llevado a cabo con éxito el proceso de migración (y si seguimos los consejos de este capítulo al pie de la letra seguramente habrá sido así), documentemos todo el proceso y lo pongamos en Internet para que les sirva de experiencia a otros usuarios. Porque esto funciona así, la gente que tiene las soluciones a los problemas de otras personas las brinda de manera desinteresada (si no GNU/Linux no sería lo que es hoy). Finalmente, recordemos que migrar a Linux es un proceso que puede llevar un par de horas, un par de días o un par de semanas. Todo depende de la forma en la que llevemos a cabo los pasos necesarios para evitar todos los posibles problemas. Por eso, conviene que seamos ordenados, que nos informemos y que nos animemos a hacerlo.

The screenshot shows the website for Linux Migration, Inc. At the top, there is a navigation bar with links for Home, Linux Quickref Wiki (New!), Linux Quick Reference, Server Linux, Desktop Linux, Company, and Contact. Below the navigation bar, there is a section titled "Find answers to your Linux questions in the Linux Quickref Wiki". The main content area lists various services and support options, including Linux training, open source customizing, real-time embedded Linux, system installation and configuration, support for Red Hat Linux, Fedora, SuSE, Ubuntu, Debian, and other open source software, network management, SNMP configuration, Linux kernel customization, and security services. At the bottom, there is a "Legal" link.

**Figura 17.** Linux Migration ([www.linuxmigration.com](http://www.linuxmigration.com)) es un excelente recurso web para encontrar información sobre cómo llevar adelante el plan de migración.

## CERTIFICACIONES

La certificación más reconocida en el mundo de Linux es la **LPI**, por lo que puede ser interesante armar el plan de capacitación según esa certificación. Para obtener más información sobre ella, podemos visitar el sitio [www.lpi.org](http://www.lpi.org).

- 1 Antes de proceder con el plan de migración conviene relacionarse con el grupo de usuarios Linux local. Ellos podrán brindar ayuda si surgen problemas.
- 2 Si no hay un grupo de usuarios en nuestra localidad, los foros en línea pueden ser una excelente opción. En [www.linuxforums.org](http://www.linuxforums.org) podemos encontrar muchos foros repletos de usuarios dispuestos a ayudar.
- 3 Siempre hay que tener en cuenta la posibilidad de recurrir a soluciones comerciales de implementación y soporte. Novell y Red Hat son dos de las compañías que ofrecen las soluciones más completas sobre Linux.
- 4 Realizar un sistema de copia de seguridad básico una vez instalado Linux es una excelente idea. Éste puede ser un simple script de Bash que lleve a cabo una copia de seguridad de la configuración del sistema y será provisorio hasta instalar un sistema de backup más completo, como Bacula.
- 5 Ubuntu posee una herramienta de migración automática de Windows a Linux. Traspasa los e-mails, los favoritos, los documentos y hasta el papel tapiz de un sistema operativo al otro.

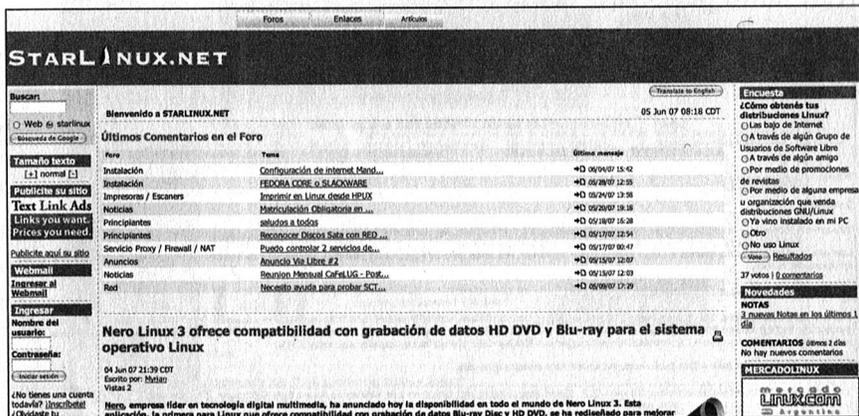
# Distribuciones

Existen muchas distribuciones y conviene conocerlas antes de instalar alguna de ellas. Por eso, en este capítulo veremos las principales características de cada una de las más populares y de algunas menos conocidas, pero igualmente recomendables.

<b>Distribuciones para los que recién empiezan</b>	260
Mandriva Linux 2006	260
Ubuntu 5.10	262
SuSE Linux 10	265
Fedora Core 5	267
RxArt 2.0	269
<b>Distribuciones para servidores</b>	271
Debian 3.1	271
Slackware	274
CentOS	276
<b>Distribuciones Live CD</b>	278
MEPIS 3.4	278
Knoppix 4	281
Tuquito K	281

www.starlinux.net

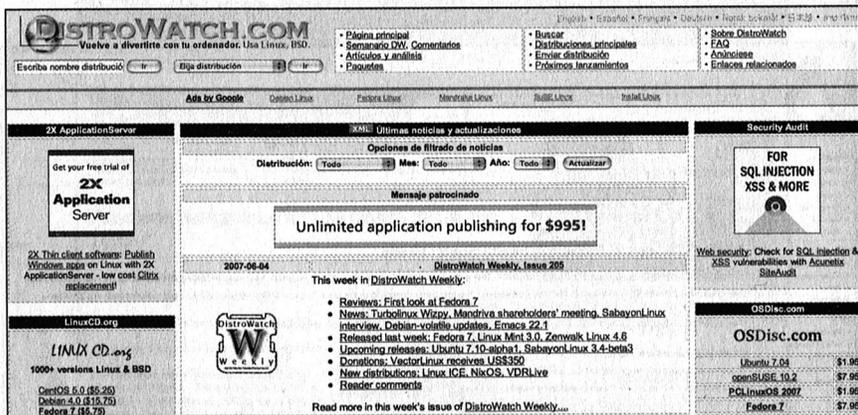
io dedicado a difundir noticias relacionadas con GNU/Linux. Incluye foros de discusión que nos permiten realizar consultas o ayudar a otros usuarios a resolver sus problemas. Se encuentra en idioma español, por lo que es una buena opción para buscar la información que necesitamos.



## distroWatch

http://distrowatch.com

Se trata del gran índice de distribuciones de Linux. Podemos realizar búsquedas por categoría y por plataforma. Aunque muchas cosas sólo se encuentran en idioma inglés, posee un ranking con las distribuciones más populares de Linux que nos puede ser de gran ayuda cuando deseamos evaluar cuál de ellas vamos a instalar.



# 202 Secretos de Linux

# Aplicaciones que no todos conocen

En este apéndice conoceremos algunas aplicaciones que no sólo nos ayudarán a mejorar nuestro trabajo en Linux, sino que también nos permitirán realizar tareas cotidianas de forma más rápida y sencilla. Además, algunas de ellas nos darán la posibilidad de hacer cosas que no creíamos posibles.

Aplicaciones útiles	298
Extracción de audio de CDs con ripperX	298
Accesorios para GNOME con gDesklets	299
Control de reproducción de música desde Firefox	299
Mantenimiento del sistema con KleanSweep	300
Descargar contenidos embebidos con Download Embedded	301
El súper portapapeles Glipper	301
Groove Search Toolbar	302
Alexandria: nuestra biblioteca virtual	303
Gmail Manager	304
Gestión de descargas con Pando	304

# APLICACIONES UTILES

En lugar de hacer una reseña de cuáles son las aplicaciones más populares para Linux, en esta sección incluimos una selección de mini-aplicaciones que, si bien no son muy populares, harán las delicias de aquellos usuarios que quieran mejorar su experiencia de usabilidad de Linux.

## Extracción de audio de CDs con ripperX

Seguramente, tenemos cientos de CDs de audio que nos gustaría pasar a la computadora. Bien, lo haremos con una pequeña aplicación muy útil llamada ripperX (<http://sourceforge.net/projects/ripperx>). Como es común en estos proyectos chicos, tendremos que compilar para instalar. Una vez que tengamos el programa instalado (Figura 1), cuando lo iniciemos nos dará error si aún no hemos coloca-

do el CD de audio en la lectora, y tendremos que presionar **Scan** después de insertarlo. Lo recomendado es primero visitar la opción **Configuración**. Allí podremos indicar en qué directorio se almacenarán nuestros temas **ripeados**, el plug-in de configuración y en qué resolución queremos que comprima (el default de 128K está más que bien) y listo. Cuando ha leído el CD, nos mostrará debajo todos los temas y algo de información como el artista, el nombre del álbum y en qué año se editó. Luego sólo resta marcar los temas que queremos **ripear** (podemos seleccionar todos con el comando alusivo) y luego presionar **Go**. Además del formato MP3, también podemos codificar en **ogg** y **flac**. Si hablamos de las ventajas del programa, nos encontramos con una codificación rápida (tarda aproximadamente un minuto y medio en **ripear** un tema de tres minutos a MP3) que consume pocos recursos. Si nos ponemos preciosos, podríamos pedir una inter-

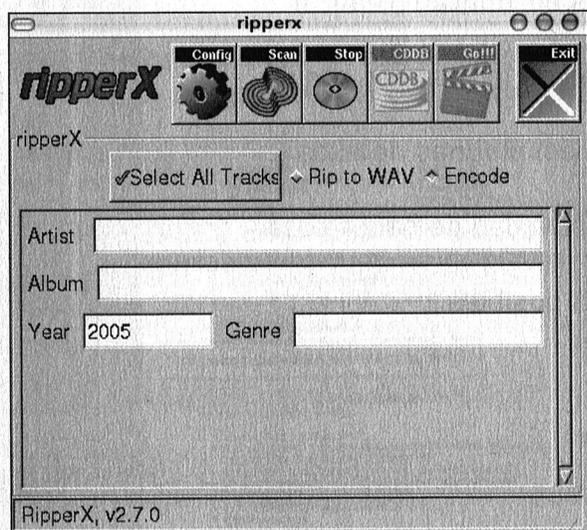


Figura 1. RipperX nos permite comprimir en varios formatos con tan sólo hacer unos clics

faz un poco más agradable, ya que la que tiene se ve algo tosca y fea. Sin embargo, sus funcionalidades y su facilidad de uso hacen que nos olvidemos de este punto en forma inmediata.

## Accesorios para GNOME con gDesklets

**GDesklets** (<http://gdesklets.de>) es una aplicación que incorpora muchísimos **Widgets** (pequeños programas aplicativos), algunos de muy buena calidad, otros muy útiles y otros que dan un renovado aspecto y que mejoran la interfaz visual de nuestro escritorio (Figura 2). Desde hace mucho tiempo existe Karamba y SuperKaramba para el proyecto KDE, por lo que los usuarios de GNOME estaban a la deriva en este apartado de embellecer nuestro escritorio. Pero esto ya está cambiando. Para instalar la aplicación simplemente hay que ir a la consola y escribir el siguiente comando:

`$ sudo aptitude install gdesklets-data.`

Luego de la descarga y la instalación de los paquetes, gDesklets estará listo para que afinemos nuestro escritorio. Sólo lo abriremos desde **Aplicaciones/Accesorios**. Algunas mini-aplicaciones que tendremos a disposición son: relojes analógicos, barras extras transparentes, carpetas con efecto en 3D, monitores de sistema que nos indican la capacidad del disco, la temperatura de nuestra máquina y hasta completos **docks** que le darán a nuestro escritorio una apariencia similar a la de Mac OS X.

## Control de reproducción de música desde Firefox

**FoxyTunes** (Figura 3) es una extensión que agrega controles de reproducción a Firefox para que podamos manejar la colección de música que estamos reproduciendo sin perder de vista la pantalla de

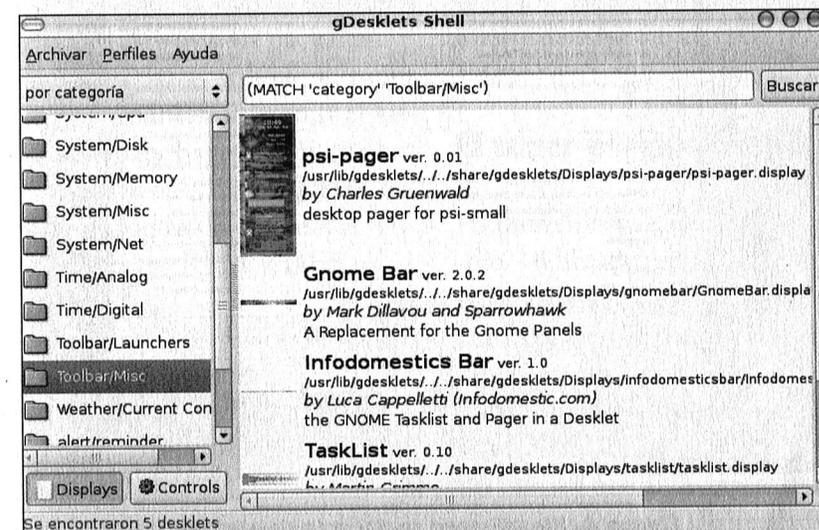
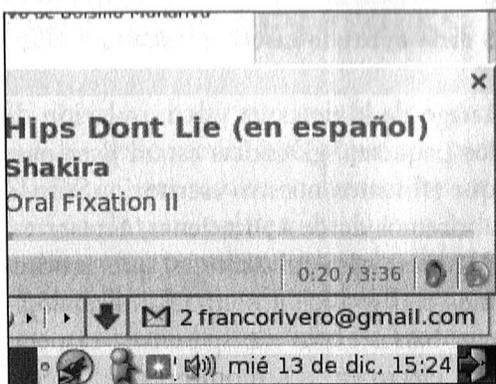


Figura 2. Con Gdesklets podremos agregar muchos widgets al escritorio GNOME.



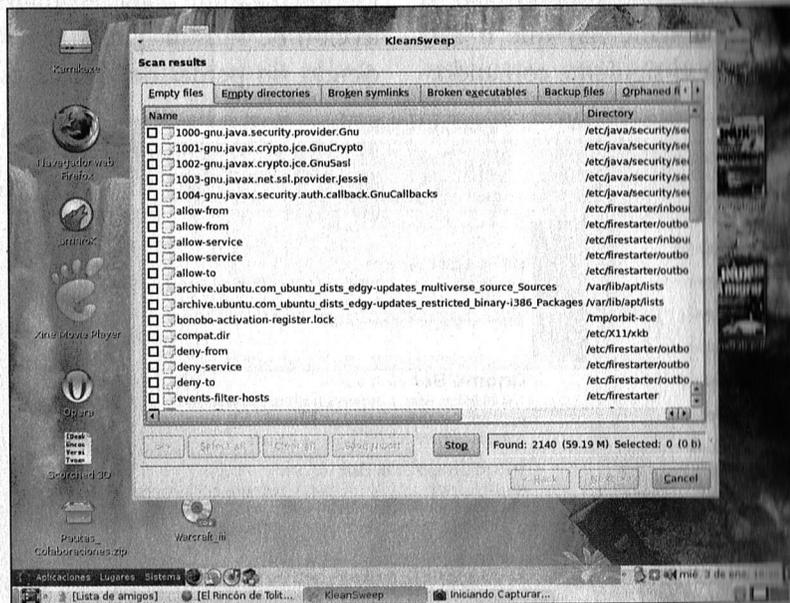
**Figura 3.** Ejecución de la extensión de Firefox FoxyTunes.

nuestro navegador. Lo primero que debemos hacer para instalarla es ir a <https://addons.mozilla.org/firefox/219>. Allí simplemente seleccionaremos **Install Now** y listo (sólo ocupa 500 kb). Cuando reiniciemos Firefox la misión estará cumplida: tendremos disponible la barra de FoxyTunes en su parte inferior. Luego de instalado deberemos configurar algún re-

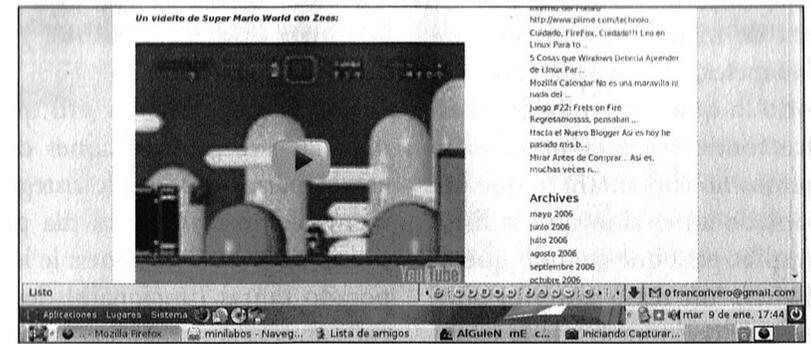
productor, ya que FoxyTunes sólo tiene comandos y no el motor de reproducción. Por ejemplo, se puede utilizar con Amarok, pero soporta muchos más. En el menú principal también es recomendable ver la ayuda de **Shortcuts** para manejar los comandos con el teclado. Luego de que tengamos todo configurado, ya estaremos listos para disfrutar de nuestra colección musical de la mano de Firefox, podremos adelantar, retroceder, poner en pausa, play, stop, obtener información sobre la canción que se está reproduciendo, maximizar y minimizar el reproductor, etcétera.

## Mantenimiento del sistema con KleanSweep

Este programa nos (Figura 4) servirá para hacer un mantenimiento y eliminar



**Figura 4.** KleanSweep se encarga de eliminar toda la basura de nuestro sistema de archivos.



**Figura 5.** Download Embedded permite descargar videos y archivos de audio incrustados en páginas web.

dos los archivos duplicados, en desuso y los archivos fantasmas que no hacen otra cosa que ocupar valioso espacio, además de ralentizar el inicio y el desempeño de nuestro sistema operativo. El problema principal radica en que hacerlo a mano es algo peligroso y poco recomendado si no sabemos con exactitud lo que se debe hacer. Para eso, nos podremos valer de esta aplicación ya que hace toda la búsqueda con un cuidado brillante y nos preguntará si estamos de acuerdo con los cambios que pretende realizar. La primera vez que se utiliza la aplicación tardará un buen rato, por lo que hay que ser pacientes. KleanSweep es una solución súper confiable para mantener limpio el sistema Linux, y para bajarlo hay que ir a <http://freshmeat.net/projects/kleansweep>, donde lo podemos encontrar en formato RPM y DEB.

## Descargar contenidos embebidos con Download Embedded

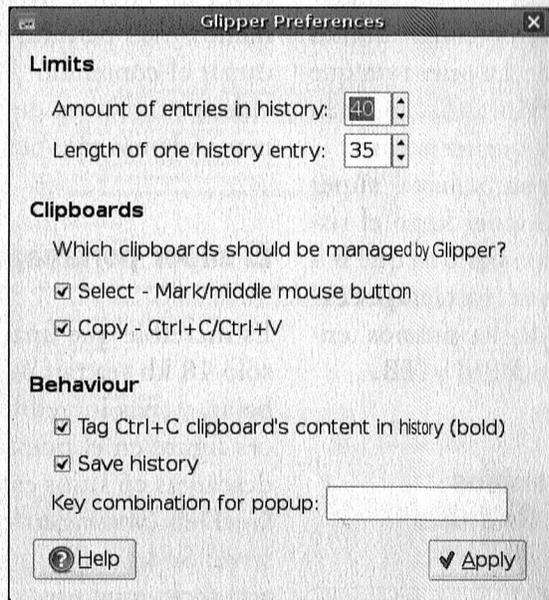
Es agradable mirar y disfrutar en línea los contenidos embebidos, pero ¿qué su-

cede si queremos ver ese material sin estar conectados? Lo más probable es que no lo podamos descargar. La solución viene de la mano de extensión Download Embedded para Firefox, que podemos instalar de <https://addons.mozilla.org/firefox/1993>. ¿Cómo funciona? Muy fácil. Cuando navegamos un sitio que contiene contenido embebido, no aparecerá una flecha roja en la barra inferior de Firefox (Figura 5). Si la presionamos, nos permitirá visualizar o descargar el contenido. Podemos descargar videos y archivos de audio sin problemas. Además sólo ocupa 30 kb.

## El súper portapapeles Glimper

Es increíble que una aplicación de tan sólo 18 kb sea tan útil a la hora de trabajar, y más increíble aún es que a veces figure en el puesto número uno de descargas en sitios especializados como GetDeb ([www.getdeb.net](http://www.getdeb.net)). ¿Para qué sirve? Se trata de un simple pero eficaz accesorio, que una vez instalado e iniciado se acomoda discretamente en nuestra barra de trabajo. Desde al-

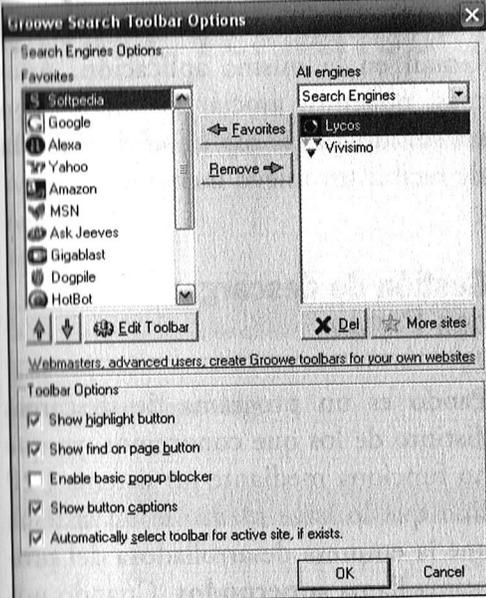
administra de manera excelente nuestro portapapeles, y funciona almacenando todo lo que copiamos de nuestras aplicaciones. De esta manera, lo que copiamos no suplantarán lo que ya hemos copiado antes sino que se listará en el **applet** para que cuando queramos recuperar un texto anterior simplemente lo seleccionemos de la lista. Cabe destacar que el texto actual se resaltará en color azul, y siempre tendremos la opción de limpiar cuando necesitemos vaciar nuestro portapapeles. Lo podemos descargar del sitio <http://glipper.sourceforge.net>. Si nos dirigimos a la opción de **Preferencias** podremos configurar algunas características como la cantidad de entradas que soportará el historial y también algunas teclas rápidas. Si somos usuarios de KDE, el proyecto original (sinónimo de **Glipper**) es **Klipper**.



**Figura 6.** Glipper nos permite establecer la cantidad de ítems que almacenará para que podamos recuperar.

## Groove Search Toolbar

Quizá la facultad más útil que contaban las primeras versiones de Firefox era la barra de Google integrada. La mentablemente, hoy en día pasa desapercibida por dos razones: le han incorporado tantas funcionalidades al navegador que contar con una barra de Google ya no es algo novedoso, y con el crecimiento de la **blogósfera** y otras aplicaciones web 2.0, la inclinación tecnológica marca otro camino. Por suerte, existe una verdadera barra de búsqueda que no es otra cosa que una extensión para nuestro navegador favorito. Estamos hablando de **Groove Search Toolbar** (**Figura 7**), y para instalarla podemos ir a <http://addons.mozilla.org/firefox/534> (sólo ocupa 112 Kb). Cabe destacar que también se encuentra disponible para Internet Explorer.

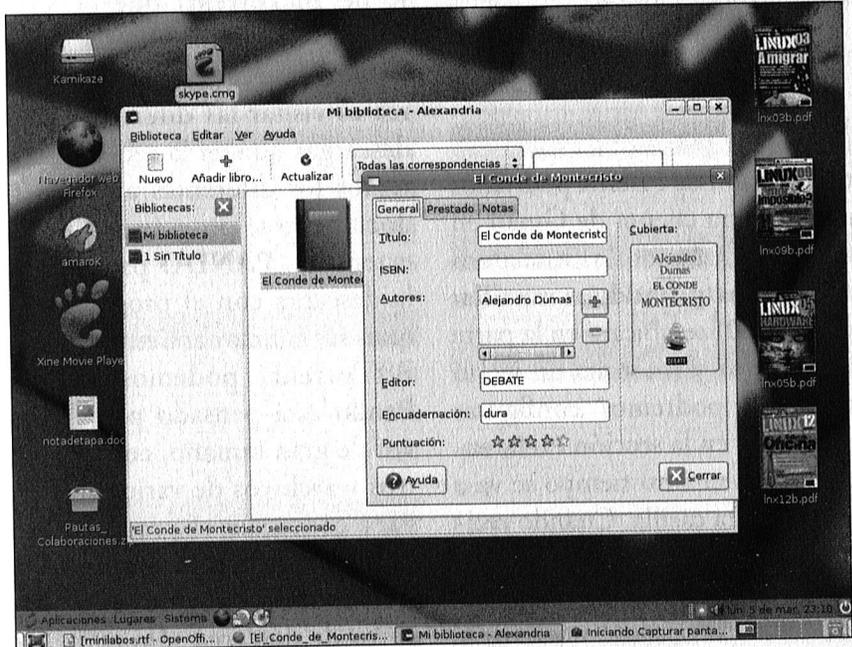


**Figura 7.** La barra de búsquedas nos permitirá acceder a información contenida en varios sitios web populares, que podemos elegir desde las opciones de configuración del programa.

Básicamente, lo que podemos hacer es buscar en sitios muy representativos como Digg, Del.icio.us, Ebay, Google, Yahoo, Amazon, Download y Mamma, entre otros. Además, podemos alterar entre diferentes motores de búsqueda hasta que demos con lo que estamos buscando. Como si fuera poco, podemos definir qué es lo que estamos buscando: noticias, blogs, videos, grupos, directorios, etcétera.

## Alexandria: nuestra biblioteca virtual

Se trata de un completo organizador de libros (**Figura 8**). Antes de empezar a utilizarlo, deberemos descargarlo (<http://alexandria.rubyforge.org>) e instalarlo y para eso, si utilizamos Debian o Ubuntu



**Figura 8.** Aquí vemos cómo Alexandria nos muestra información sobre uno de los libros registrados.

desde una consola, deberemos ingresar la siguiente: **apt-get install alexandria**. También podemos instalarlo en otras distribuciones como Mandriva, de manera sencilla con **urpmi alexandria** desde una terminal.

Una vez instalado, lo encontraremos en la opción **Oficina** de nuestro menú. Con éste podremos incorporar libros añadiendo todos sus datos como ISBN, nombre, editor y cantidad de páginas. Como valor agregado, también podremos aclarar la persona que lo tiene, si es que lo hemos prestado. Identificar los libros en el programa es súper fácil, ya que podremos bajar una imagen de la portada del libro de Internet y anexarla. Así se mostrarán las portadas de nuestros libros en lugar de simplemente su nombre. También podremos realizar búsquedas por cualquiera de las categorías que enumeramos antes.

## Gmail Manager

Con esta extensión de Firefox podremos sincronizar nuestra cuenta de Gmail en pocos segundos. Cuando la instalamos (lo hacemos de <http://addons.mozilla.org/firefox/1320>) se ubicará en la parte inferior de Firefox y mostrará un icono de Gmail. Allí podremos configurar nuestra cuenta y, en la sección **Preferencias**, indicar cada cuánto tiempo se va a comprobar nuestra casilla. Cuando recibamos nuevos mensajes mostrará la cantidad al lado del icono de la extensión. Pero lo bueno de la extensión no termina aquí, sino que también debemos tener en cuenta que nos dará la posibili-

dad de configurar varias cuentas de Gmail en la misma aplicación. Además, podemos asociar los alertas con un sonido que se reproducirá a la hora de recibir un nuevo mensaje.

## Gestión de descargas con Pando

Pando es un programa de descargas distinto de los que conocemos, ya que no funciona mediante un sistema P2P sino que lo hace mediante un sistema que la empresa desarrolladora del programa llama **supernodos**. Cuando nosotros subimos un archivo, éstos se cachean en los servidores de la empresa para quedar disponibles. Por eso podemos decir que, a nivel funcional, Pando está más cerca de un Azureus (cliente de BitTorrent) que de un Amule (cliente de Edonkey).

Para realizar una descarga, nuestra tarea es visitar las diferentes páginas de descargas que en la red. Allí encontraremos los diferentes archivos (juegos, documentos, programas, música) con extensión **.PANDO** para luego iniciar la descarga con el programa. Al saber que su funcionamiento es similar a BitTorrent, podemos deducir que Pando está pensado para bajar archivos de gran tamaño, como ISO de discos y archivos de varios gigas.

Para conseguir el programa podemos ir a la página oficial del proyecto en [www.pando.com/download](http://www.pando.com/download). El archivo de descarga sólo ocupa tres megas y necesita al menos 256 MB de memoria RAM para funcionar.