

## **EL SHELL: COMANDOS BÁSICOS DE *LINUX***

Cualquier usuario de Linux acabará antes o después relacionándose con el sistema empleando el modo texto, y no los entornos gráficos. Este modo se basa en la ejecución de una serie de comandos, que son interpretados por un programa o ***shell***. *Linux* dispone de varios de estos programas pero el más habitual es conocido como ***bash*** o *Bourne Shell*. Si Linux se ha arrancado en modo texto el sistema arranca de forma directa el *shell* y queda a la espera de introducción de nuevos comandos. Si se ha arrancado en modo gráfico se puede acceder al *shell* de dos formas:

Se puede acceder al *shell* del sistema presionando alguna de las siguientes combinaciones de teclas:

➤<ctrl>+<alt>+<F1>

➤<ctrl>+<alt>+<F2>

➤<ctrl>+<alt>+<F3>

➤<ctrl>+<alt>+<F4>

➤<ctrl>+<alt>+<F5>

➤<ctrl>+<alt>+<F6>

Esto hace que el sistema salga del modo gráfico y acceda a alguna de las seis consolas virtuales de *Linux*, a las cuales también se puede acceder cuando se arranca en modo de texto. Para volver al modo gráfico hay que presionar

➤<ctrl>+<alt>+<F7>

La segunda forma es más cómoda y menos radical permitiendo acceder al shell desde el mismo entorno gráfico. Para esto hay que abrir un programa

llamado terminal o consola, por ejemplo: **kconsole** (en el entorno *KDE*), **xterm**, **gnome-terminal** (en GNOME), etc.

## **Introducción**

Existen una serie de nociones básicas que hay que tener en cuenta a la hora de introducir los comandos. En primer lugar citaremos las siguientes:

- Los comandos hay que teclearlos exactamente.
- Las letras mayúsculas y minúsculas se consideran como diferentes.
- En su forma más habitual (los shells de Bourne o de Korn), el sistema operativo utiliza un signo de \$ como **prompt** para indicar que está preparado para aceptar comandos, aunque este carácter puede ser fácilmente sustituido por otro u otros elegidos por el usuario. En el caso de que el usuario acceda como administrador (root) este signo se sustituye por #.
- Cuando sea necesario introducir el nombre de un fichero o directorio como argumento a un comando, Linux, permite escribir las primeras letras del mismo y realiza un autorrellenado al presionar la tecla del tabulador. Si no puede distinguir entre diversos casos rellenará hasta el punto en el que se diferencien. Por ejemplo, supongamos una carpeta con los siguientes directorios:

Programas

Documentos\_proyecto

Documentos\_privados

Al escribir `cd Pr<tab>` Linux rellenará el resto del contenido hasta escribir `cd Programas`. Por el contrario al escribir `cd D<tab>` escribirá `cd Documentos`

- Sintaxis de los comandos

Los comandos tienen la siguiente sintaxis:

*# programa arg1 arg2 ... argn*

Se observa que, en la ``línea de comandos'', se introduce el programa seguido de uno o varios argumentos. Así, el intérprete ejecutará el programa con las opciones que se hayan escrito.

Cuando se quiere que el comando sea de varias líneas, se separa cada línea con el carácter barra invertida (\). Además, cuando se quiere ejecutar varios comandos en la misma línea, los separa con punto y coma (;). Por ejemplo:

*# make modules ; make modules\_install*

- En los comandos, también se puede utilizar los comodines:

El asterisco (\*) es equivalente a uno o más caracteres en el nombre de un archivo. Ejm: `ls *.c` lista todos los archivos con extensión c.

El signo de interrogación (?) es equivalente a un único carácter. Ej.: `ls curso.te?` lista el archivo `curso.tex` completando el último carácter.

Un conjunto de caracteres entre corchetes es equivalente a cualquier carácter del conjunto. Ej.: `ls curso_linux.t[aeiou]x` lista `curso_linux.tex` seleccionando la e del conjunto.

## **Trabajando en un terminal.**

Es difícil que hagamos de GNU/Linux nuestro entorno de trabajo habitual sin utilizar nunca la línea de comandos (el verdadero alma de Linux) Vamos a efectuar un primer acercamiento.

Para entrar en un terminal podemos hacerlo en **Aplicaciones → Configuración→Terminal** (o haciendo clic con el botón derecho sobre el escritorio de Guadalínex)



Una vez en el terminal conviene familiarizarse con la línea de comandos. Lo que vemos es un mensaje del sistema (prompt de la shell). Nos informa de que usuario soy (amfabra) como se llama mi ordenador (coordinador) y en que carpeta o directorio me encuentro (:~ que significa el directorio del usuario /home/amfabra) y un separador \$. El símbolo del separador es importante: Se utiliza el símbolo de dólar (\$) para los usuarios normales y el símbolo # para significar que es el root el que está dando ordenes.

Para cambiar a otro usuario escribiremos :

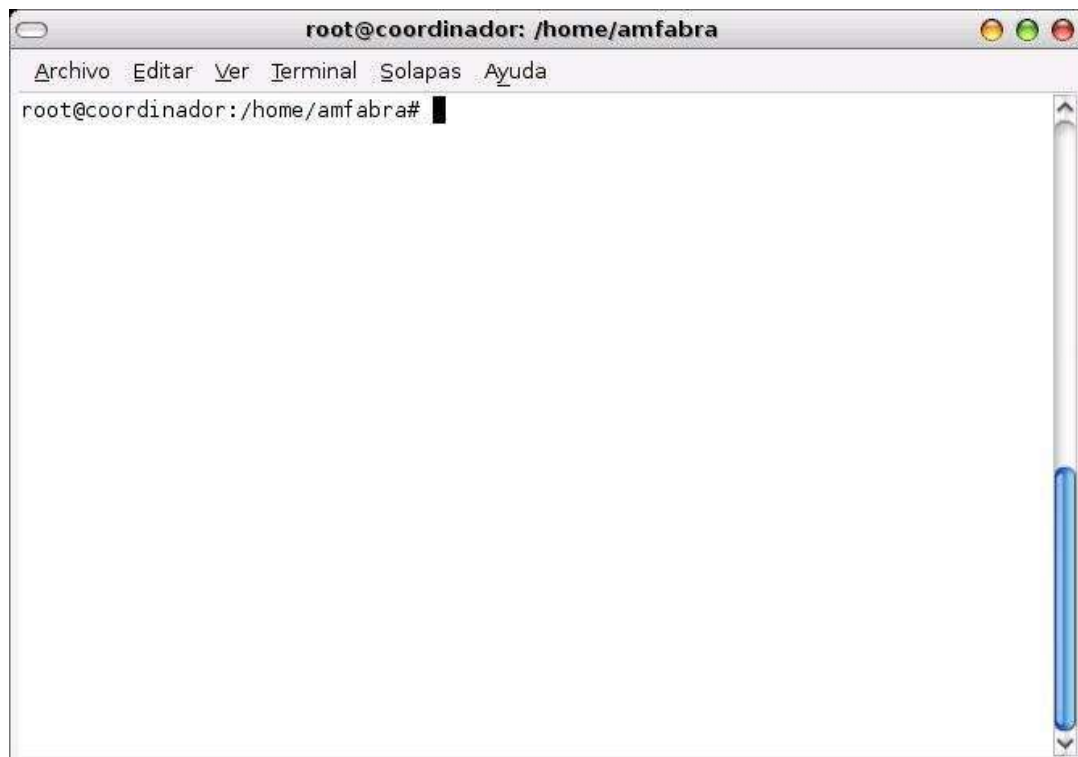
`$su nombre_de_usuario`

lógicamente ese usuario debe estar dado de alta en el sistema, a continuación se pedirá su clave.

Para pasar a ser superusuario escribiremos:

`$su`

Tras lo cual el sistema nos pedirá la contraseña de root y, si es correcta, cambiará a modo de root. Observa como cambia el prompt:



- Advertencia: Trabajar como root puede dejar al sistema inservible si no se sabe lo que se está haciendo. Aunque a veces no hay más remedio.

## **Moviéndonos por los directorios**

La orden para movernos por la estructura de la red es '**cd**' y su formato es el siguiente:

**cd** [directorio]

Donde 'cd' es el comando y lo que está entre corchetes es el argumento, al estar entre corchetes es opcional.

Para movernos de un directorio a otro usamos este comando, tenemos 2 formas de nombrar un directorio: la primera es por su ruta absoluta y la segunda es por su ruta relativa. La absoluta se refiere al nombre completo empezando desde '/', un ejemplo sería ir a /etc (yo estoy en /home/amfabra).

```
amfabra@coordinador:~$ cd /etc
amfabra@coordinador:/etc]$
```

Esto tiene un problema, ya que siempre tenemos que escribir la ruta completa a donde queramos ir. Para no tener que hacerlo, usamos las rutas relativas: ponemos el nombre del directorio que queremos entrar pero a partir desde el que estamos. Para ver esto volveremos a /home/amfabra (en vuestro caso el directorio de trabajo vuestro) y de allí iremos a /home/amfabra/Documentos.

```
amfabra@coordinador:/etc]$ cd /home/amfabra
amfabra@coordinador:~$ cd Documentos
amfabra@coordinador:~/Documentos$
```

En el primer caso hemos usado de nuevo una ruta absoluta y en el segundo como ya estábamos en /home/amfabra hemos puesto solo Documentos para entrar en /home/amfabra/Documentos, que sería su ruta

absoluta.

Dentro de todo directorio existen dos directorios especiales que son '.' y '..'. El primero hace referencia al directorio actual, es decir, si haces 'cd .' te quedas donde estás (el directorio especial '.' ya veréis más adelante lo útil que es), y el segundo hace referencia al directorio padre, o sea, si estamos en /home/amfabra y hacemos 'cd ..' terminaremos en /home.

```
amfabra@coordinador:~$ cd .  
amfabra@coordinador:~$ cd ..  
amfabra@coordinador:/home]$
```

Estos dos directorios especiales los usaremos en la rutas relativas, es lo mismo hacer 'cd ./Documentos' que 'cd Documentos' que para este comando en concreto nos da lo mismo pero en otros puede que necesitéis usar './'.

Imaginaros que desde /home/amfabra queréis entrar (si tuvierais otro usuario llamado así) en /home/kronem, para no escribir la ruta absoluta recurriremos a 'cd ../kronem' y conseguimos mediante una ruta relativa dar un salto atrás en el árbol y entrar en 'kronem'

```
amfabra@coordinador:~$ cd Documentos  
amfabra@coordinador:~/Documentos]$ cd ..  
amfabra@coordinador:~$ cd ./Documentos  
amfabra@coordinador:~/Documentos]$ cd ..  
amfabra@coordinador:~$ cd ../kronem  
amfabra@coordinador:/home/kronem$
```

Bueno pues no se vayan todavía, aun hay más. Hay una referencia especial, estemos donde estemos si hacemos 'cd ~' volvemos a nuestro directorio de trabajo ya que ~ (bajo consola lo conseguimos pulsando la tecla Alt Gr +4) es una referencia a nuestro directorio de trabajo.

```
amfabra@coordinador:~$ cd /usr/bin
amfabra@coordinador:/usr/bin]$ cd ~
amfabra@coordinador:~$
```

Con esto ya hemos visto este comando tan simple pero útil. Se consigue el mismo resultado escribiendo 'cd'.

### **Me he perdido ¿Dónde estoy?**

Si nos perdemos, bien por que hemos enredado mucho moviéndonos por el árbol de directorios o bien porque nuestro prompt no nos indica en cada momento dónde estamos, tenemos un comando bien sencillo:

pwd

```
amfabra@coordinador:~$ pwd
/home/amfabra
amfabra@coordinador:~$
```

### **Antes de continuar...**

Antes de continuar hay que comentar una cosa, para evitar confusiones explicaré un poco los 'alias'. El intérprete de comandos nos da la opción de definirnó lo que denominaremos 'alias', por ejemplo, puedes hacer que la palabra 'casa' ejecute 'cd~', parece una bonita forma de crear comandos a nuestro gusto. Algunos de los comandos que aquí explicaré tienen definidos alias en muchas distribuciones por lo que el resultado puede que no sea el mismo. Si esto os ocurre teclead:

'alias COMANDO\_QUE\_FALLA=COMANDO\_QUE\_FALLA'

(ojo, sin espacios en el igual), por ejemplo, si cuando hicierais 'cd' os mandara a /usr/bin (es un ejemplo raro, pero bueno) bastaría con ejecutar

'alias cd=cd'.

Para ver qué alias tenéis definidos ejecutad 'alias -p'



```
amfabra@coordinador:~$ alias -p
alias ls='ls --color=auto'
amfabra@coordinador:~$
```

Lo que veremos a continuación es muy probable que tengas que hacerlo para que salga lo mismo que te estoy explicando. El uso de alias sobre todo se utiliza para ejecutar por defecto opciones muy útiles sin tener que escribirlas siempre, incluso si vienes del mundo de MS-DOS puedes definir todos tus comandos de siempre con los análogos en GNU/Linux, útil, ¿verdad? ;) (aunque esto último no lo recomiendo, no es bueno coger malos vicios XD ).

*Nota:* los alias que definamos se esfumarán cuando cerremos nuestra sesión o reiniciemos la máquina, pero no tengáis miedo. Ya explicaremos más adelante cómo definir las cosas permanentemente.

### Pues a mí me pica la curiosidad

Bueno, ya sabemos movernos, pero esto no nos sirve de mucho si no sabemos qué hay en nuestros directorios. Para eso tenemos un buen comando:

`ls [opciones][directorio]`

Ahora ejecutaremos 'ls' tal cual y nos encontraremos con la siguiente salida:

```
amfabra@coordinador:~/Documentos$ alias ls=ls
Ejemplos
amfabra@coordinador:~/Documentos$ cd Ejemplos
amfabra@coordinador:~/Documentos/Ejemplos$ ls
Boja Decreto Sociedad del Conocimiento.pdf      tux.dxf
amfabra@coordinador:~/Documentos/Ejemplos$
```

Nos encontramos con una salida algo pobre, porque en algunos casos no sabremos ni diferenciar qué es cada cosa: ¿un fichero? ¿un directorio? . Con la

opción '-F' obtendremos más información:

```
amfabra@coordinador:~/Documentos/Ejemplos$ ls -F
Boja Decreto Sociedad del Conocimiento.pdf*  tux.dxf*
amfabra@coordinador:~/Documentos/Ejemplos$
```

Ahora podemos apreciar que a los directorios al final le añaden '/'. A ls también le podemos indicar el directorio que queremos mirar:

```
amfabra@coordinador:~$ ls -F /usr/local
bin/  games/  include/  lib/  man@  sbin/  share/  src/
amfabra@coordinador:~/amfabra$

amfabra@coordinador:~/Documentos/Ejemplos$ cd
amfabra@coordinador:~$ ls -F
Audio/          Compartido@  Fotos/       Tipos de Letra@
Bases de Datos/ Descargas/   P2P/         Videos/
bienvda.gif     Desktop/    prefs.js     wireless/
bookmarks.html  Documentos/ Templates/
amfabra@coordinador:~/amfabra$
```

En este caso además vemos un \* al final de estos ficheros, lo que nos indica que son ejecutables. También podemos encontrarnos con una @ indicando que es un enlace. Otra opción interesante y que nos será útil es '-a' que nos muestra los ficheros (acuérdate de que en GNU/Linux todo son ficheros) ocultos.

```

amfabra@coordinador:~$ ls -aF
./      Documentos/          .gtkrc-1.2-gnome2    .recently-used
../     .first_login            .ICEauthority        .scribus/
.alias  .fluxbox/              .icons/              .sversionrc
.aMule/ .fonts/                .inkscape/           .synce/
Audio/  .fonts.cache-1         .kde/                Templates/
Bases de Datos/ Fotos/                .kderc               .themes/
.bash_history .gconf/              .liferea/ .thumbnails/
.....
amfabra@coordinador:~$

```

Podemos observar que aparecen más ficheros que estaban ocultos. En GNU/Linux, cualquier fichero que empiece por '.' es un fichero oculto contando los directorios especiales '.' y '..' que ya os comenté antes. Una opción realmente útil y que os interesará es 'l' :

```

amfabra@coordinador:~$ ls -aFl
total 416
drwxr-xr-x  42 amfabra users  4096 2006-01-06 01:03 ./
drwxrwsr-x   4 root   staff  4096 2006-01-04 13:22 ../
-rw-r--r--   1 amfabra users    266 2006-01-04 13:22
.alias
drwxr-xr-x   2 amfabra users  4096 2006-01-04 13:22
.aMule/
drwxr-xr-x   3 amfabra users  4096 2006-01-04 12:24
Audio/
drwxr-xr-x   2 amfabra users  4096 2006-01-04 12:24 Bases
de Datos/
-rw-----   1 amfabra users    96 2006-01-06 01:03
.bash_history
-rw-r--r--   1 amfabra users   704 2006-01-04 13:22
.bash_profile
.....
-rw-r--r--   1 amfabra users  2682 2006-01-04 13:22
.bashrc-rw-r--r--   1 amfabra users 32281 2006-01-06
01:03 .xsession-errors
amfabra@coordinador:~$

```

Para empezar veréis que se pueden mezclar opciones y que '-l' nos da una gran cantidad de información. La primera indica el tipo de fichero y los permisos (por ejemplo, drwxrwxr-x), los permisos no los explicaré ahora por que no toca, pero sí comentar que la primera letra indica qué tipo de fichero es ('d' nos indica que es directorio, '-' que es un fichero normal, 'c' que es un dispositivo orientado a carácter, 'b' dispositivo orientado a bloque y 'l' indica que es un enlace). Lo siguiente nos indica: el número de enlaces que existen a él, el propietario, el grupo propietario, el tamaño en bytes, la fecha y el nombre.

Por último, ya sólo enseñaros otra opción que os gustará muchísimo '--color':

```
amfabra@coordinador:~$ ls --color
Audio                bienvda.gif          Compartido  Desktop
Fotos  prefs.js  Tipos de Letra  wireless
Bases de Datos  bookmarks.html  Descargas  Documentos
P2P    Templates  Videos
amfabra@coordinador:~$
```

La verdad se explica por ella sola, ¿no? No seguiremos con más opciones de 'ls' porque hay casi tantas como letras del abecedario, tanto en mayúsculas como en minúsculas, así que puedes incluso probar 'ls' con tu nombre y demostrar lo triste que es tu vida perdiendo el tiempo en semejantes tonterías.

Ya para finalizar sólo indicaros que, si estáis en modo consola, para cerrar la sesión hay que teclear 'exit' o 'logout', para parar la máquina 'halt' (solo se puede hacer como root) y para reiniciarla pulsar 'ctrl+alt+sup'.

## **Creando directorios**

Para empezar, después de haber aprendido a movernos por el sistema lo primero que veremos es cómo crear un directorio, para ello tenemos la siguiente instrucción:

```
mkdir [opciones] directorio [directorio2 ... directorio 3]
```

Donde como podéis apreciar podemos crear de uno a varios directorios a la vez, así que ahora toca probarlo.

```
amfabra@coordinador:~$ mkdir curso
amfabra@coordinador:~$ cd curso
amfabra@coordinador:~/curso$ mkdir dir1 dir2
amfabra@coordinador:~/curso$ ls --color -l
total 8
drwxrwxr-x 2 matados2k matados2k 4096 may 18 16:15 dir1
drwxrwxr-x 2 matados2k matados2k 4096 may 18 16:15 dir2
```

Como podéis ver el uso es bien sencillo, como opciones vamos a destacar '-p' que crea los directorios padres que no existan, como vemos en el siguiente ejemplo.

```
amfabra@coordinador:~/curso$ mkdir -p ./dir1/NOEXISTO/ops
amfabra@coordinador:~/curso$ cd dir1
amfabra@coordinador:~/curso/dir1$ ls
NOEXISTO
amfabra@coordinador:~/curso/dir1$ cd NOEXISTO/
amfabra@coordinador:~/curso/dir1/NOEXISTO$ ls
ops
amfabra@coordinador:~/curso/dir1/NOEXISTO$
```

Y por último la opción '-m', que nos da la posibilidad de indicarle los permisos que tendrán nuestros directorios, pero... como aún no lo hemos visto en el curso queda pendiente de que lo probéis vosotros mismos cuando veamos cómo son los permisos. La opción -m va seguido del modo antes de poner los directorios.

### **No me gustan los directorios**

Ya que hemos aprendido a crearlos hay que aprender la opción contraria, y ya que mkdir venía de make directory ahora necesitamos un remove directory:

`rmdir [opciones] directorio [directorio2 ... directorio 3]`

Así que lo que nos queda es comenzar a probarlo:

```
amfabra@coordinador:~/curso/dir1/NOEXISTO]$ rmdir ops
amfabra@coordinador:~/curso/dir1/NOEXISTO]$ cd ..
amfabra@coordinador:~/curso/dir1]$ cd ..
amfabra@coordinador:~/curso$ rmdir ./dir1/NOEXISTO/
amfabra@coordinador:~/curso$ ls
dir1 dir2
amfabra@coordinador:~/curso$ rmdir dir1 dir2
amfabra@coordinador:~/curso$ ls
amfabra@coordinador:~/curso$
```

Como podéis ver se pueden borrar varios a la vez, igual que con mkdir creábamos varios a la vez. En este caso para rmdir solo comentaré una opción interesante y es '-p'. Esta opción lo que hace es borrar recursivamente la ruta completa que le indiquemos, por ejemplo, si tenemos /a/b/c un 'rmdir -p /a/b/c' es equivalente a 'rmdir /a/b/c' + 'rmdir /a/b' + rmdir 'a'. Vayamos al ejemplo:

```
amfabra@coordinador:~/curso$ mkdir -p ./dir1/uno/dos
amfabra@coordinador:~/curso$ ls
dir1
amfabra@coordinador:~/curso$ ls dir1
uno
amfabra@coordinador:~/curso$ ls ./dir1/uno
dos
amfabra@coordinador:~/curso$ rmdir -p dir1/uno/dos
amfabra@coordinador:~/curso$
```

### **Tampoco me gusta teclear tanto**

Esto es algo que quizás debí contar anteriormente, pero me acabo de acordar ahora mismo. El intérprete de comandos, para facilitarnos nuestro trabajo, rellena por nosotros con lo más lógico los comandos y los archivos, simplemente empezando a escribirlos y pulsando tabulador. Por ejemplo, no nos acordamos de cómo se escribía el comando 'mkdir' y solo nos acordamos de 'mkd' pulsamos el tabulador y nos lo rellena automáticamente o nos indicará qué comandos empiezan por 'mkd'. En mi caso tabulo y me pita, no me sale nada, vuelvo a tabular y ahora sí me da los comandos que empiezan por mkd:

```
amfabra@coordinador:~$ mkd (Una tabulación y nada, dos y
tachan ...)
mkdep mkdir mkdirhier
amfabra@coordinador:~$ mkd
```

Ahora si añado una 'i' me completa con 'mkdir', ya que 'mkdir' está contenido dentro del nombre de 'mkdirhier'. Ahora tenemos lo que buscábamos, pero si volvemos a tabular nos escribirá 'mkdirhier'. Con lo que

llegamos a la conclusión de que rellenará cuando: a) No hay otro comando que coincida con la cadena que hayamos escrito, y b) Cuando el nombre de un comando esté contenido en otro pudiendo seguir tabulando para llegar al otro.

Cuando existen muchas posibilidades, tantas que no caben en pantalla, antes nos preguntará si mostrarlo o no. Por ejemplo, ejecutar:

```
amfabra@coordinador:~$ m (tabulo dos veces)
Display all 190 possibilities? (y or n)
amfabra@coordinador:~$ m
```

En mi caso he contestado con 'n' ya que no quiero ver la lista completa. Por ejemplo, si quisiéramos ver todos los ejecutables accesibles no tendríamos más que tabular dos veces sin escribir nada.

```
amfabra@coordinador:~$
Display all 2809 possibilities? (y or n)
amfabra@coordinador:~$
```

Todo esto es completamente aplicable a los ficheros que indicamos a nuestros comandos, probad a escribir 'cd /e' y tabular.

## **Copiando voy y copiando vengo**

Y una vez visto esto llegamos al momento en el que vamos a crear copias de los archivos, para ello tenemos el siguiente comando:

cp [Opciones] Origen Destino

El uso es bien sencillo, sólo hay que indicar el origen y el destino de lo que queremos copiar:



```
amfabra@coordinador:~/curso$ cp /etc/fstab  
amfabra@coordinador:~/curso$
```

Como vemos no se nos pide confirmación para copiar, y pudiera ocurrir que nos hubiésemos equivocado y sobreescribiéramos un archivo que ya tuviera sin que nos pregunte, ya que 'cp' también renombra incluyendo el nombre en el destino. Veamos un ejemplo:

```
amfabra@coordinador:~/curso$ cp /etc/shells ./fstab  
amfabra@coordinador:~/curso$ ls  
dir1 fstab  
amfabra@coordinador:~/curso$
```

Como vemos hemos copiado el archivo 'shells' que está en '/etc' a nuestro directorio actual y con el nombre 'fstab' con lo cual hemos sobreescrito nuestro fichero original y eso puede ser algo que no deseemos. Para hacer que nos pregunte antes de copiar usamos la opción '-i' como vemos en el ejemplo:

```
amfabra@coordinador:~/curso$ cp -i /etc/fstab .  
cp: ¿sobreescribir './fstab'? (s/n) s  
amfabra@coordinador:~/curso$
```

Bueno, ya hemos recuperado nuestro fichero original, menos mal ;) , pero 'cp' es aún más versátil, ya que con la opción '-r' podemos copiar directorios enteros aunque es preferible usar la opción '-R' porque '-r' no tiene un comportamiento definido (o sea que puede pasar cualquier cosa) si se copian ficheros que no son normales como pueden ser un dispositivo. Veamos un ejemplo:

```
amfabra@coordinador:~/curso$ mkdir copia_de_bin
amfabra@coordinador:~/curso$ cp -r /bin ./copia_de_bin
amfabra@coordinador:~/curso$ cd copia_de_bin/
amfabra@coordinador:~/curso/copia_de_bin]$ ls
bin
amfabra@coordinador:~/curso/copia_de_bin$ cd bin
amfabra@coordinador:~/curso/copia_de_bin/bin$ ls
arch dmesg ipcalc ping tcsh
..... (omito archivos para que no ocupe
tanto).....
egrep          lsmod          pidof          tar
znew
amfabra@coordinador:~/curso/copia_de_bin/bin]$
```

Y para rematar sólo queda comentar que 'cp' también tiene la opción '-p', que es igual que en 'mkdir' y 'rmdir' así que sobran las explicaciones. Por cierto, 'cp' admite más de un origen, así que puedes copiar varios archivos en una sola línea a un mismo destino.

### **Y borrando por el camino yo me entretengo**

Si 'cp' viene de copy entonces algunos ya habréis adivinado qué orden es la de borrar:

```
rm [opciones] lista_de_ficheros_a_borrar
```

Veamos el uso más simple de rm:

```
amfabra@coordinador:~/curso$ cp fstab teborrare
amfabra@coordinador:~/curso$ ls
copia_de_bin  dir1  teborrare  fstab
amfabra@coordinador:~/curso$ rm teborrare
amfabra@coordinador:~/curso$ ls
copia_de_bin  dir1  fstab
amfabra@coordinador:~/curso$
```

Igual que 'cp' , 'rm' también tiene la opción '-i' y también puede borrar directorios enteros con '-r' y '-R' (en este caso ambos son iguales), aunque también se borran con '-d', pero '-d' no tiene un carácter recursivo y deja desenlazados los ficheros que contiene (o sea que es una cagada, así que mejor no usarla) lo que quiere decir que se quedan como diríamos sin ningún directorio que los contenga, por lo que hay que tener mucho cuidado con esta opción.

```
amfabra@coordinador:~/curso$ rm -ri copia_de_bin dir1
rm: ¿descender al directorio `copia_de_bin'? (s/n) s
rm: ¿descender al directorio `copia_de_bin/bin'? (s/n) s
.....
rm: ¿borrar el enlace simbólico `copia_de_bin/bin/csh'?
(s/n) s
rm: ¿borrar el fichero regular `copia_de_bin/bin/tcsh'?
(s/n) s
.....
rm: ¿borrar el directorio `dir1/uno'? (s/n) s
rm: ¿borrar el directorio `dir1'? (s/n) s
amfabra@coordinador:~/curso$
```

Solo queda apuntar que, tanto 'cp' como 'rm', si '-i' hace que pregunte la opción contraria es '-f' que no preguntará nada de nada.

## **Estamos en movimiento**

Venga, que seguro que algunos ya se han imaginado que para mover es:

`mv [Opciones] origen destino`

Bueno, 'mv' equivale a copiar y borrar, y al igual que 'cp' admite varios orígenes y un directorio destino. Así que con los ejemplos vistos antes con 'cp' sobran los ejemplos, salvo para el caso en que el origen y el destino son el mismo, pero en el destino se indica un nombre de fichero con lo cual lo que hacemos es renombrar:

```
amfabra@coordinador:~/curso$ ls
fstab
amfabra@coordinador:~/curso$ mv fstab perro
amfabra@coordinador:~/curso$ ls
perro
amfabra@coordinador:~/curso$
```

Como veis el directorio origen y destino es el mismo, por lo cual en el destino hemos indicado un cambio de nombre.

Como 'mv' no tiene opción que digamos interesantes salvo quizás '-i' y '-f' (que es igual que en 'cp' y 'rm') pues damos por finalizada la explicación.

## **Un asterisco para dominarlos a todos.**

Vamos a ver el uso de los comodines, que no es ni más ni menos que una característica del intérprete de comandos que nos permite referirnos a un conjunto de ficheros a la vez.

Empezaremos viendo primero el comodín '\*': el comodín '\*' hace referencia a cualquier carácter o cadena de caracteres (es decir, sustituye a uno, ninguno o muchos caracteres). Para entenderlo bien, ya que la explicación puede ser un poco confusa, veamos como siempre un ejemplo simple:

```
amfabra@coordinador:~$ cd /dev
amfabra@coordinador:/dev]$ ls l*

lmscd    loop0    loop2    loop4    loop6    lp0    lp2
logibm   loop1    loop3    loop5    loop7    lp1
amfabra@coordinador:~/dev]$
```

Lo que acabamos de indicarle a ls con el uso de '\*' es que nos liste todos los ficheros que empiecen por 'l' seguido de cualquier cosa, incluso nos lista el contenido de un directorio que empieza por 'l'. Otro ejemplo, para que nos quede definitivamente claro, puede ser este:

```
amfabra@coordinador:~/dev]$ ls *rr*

fb0current  fb2current  fb4current  fb6current  stderr
fb1current  fb3current  fb5current  fb7current
amfabra@coordinador:~/dev]$
```

En este caso lo que hemos hecho es decirle que nos liste todos los ficheros que contengan la cadena 'rr' (incluso los que empiecen o terminen

por 'rr' ya que '\*' incluso puede ser ningún carácter).

Si sois observadores y/o curiosos veréis que no funciona con los ficheros ocultos (acordaros que empiezan por '.' ), porque afectaría entre otras cosas a los directorios especiales '.' y '..' que ya expliqué en entregas anteriores, y según con qué comandos lo utilicemos puede provocar verdaderos problemas (como por ejemplo rm).

Otra cosa importante es que en los ejemplos anteriores no es el comando 'ls' el que recibe por ejemplo el argumento 'l\*', sino que es el intérprete de comandos (en mi caso bash) el que se encarga de buscar las coincidencias y lo que haría es ejecutar algo como 'ls lirc logimouse loop10 loop13 loop2 ....' . Lo que quiero decir es que no es el comando el que procesa los comodines, sino el propio intérprete de comandos, que se encarga luego de llamar al comando con los comodines ya procesados.

### **Y una interrogación para cada señor de la tierra GNU.**

El siguiente y último comodín es '?' que hace referencia a cualquier carácter, pero en este caso sólo a uno (no puede ser ninguno como el caso de '\*'). Para verlo claro veamos un ejemplo (para variar):

```
amfabra@coordinador:~/dev]$ ls ?t??  
  
ptmx  st0m  st1m  tty2  tty5  tty8  vtx1  vtx4  vtx7  
st0a  st1a  tty0  tty3  tty6  tty9  vtx2  vtx5  vtx8  
st0l  st1l  tty1  tty4  tty7  vtx0  vtx3  vtx6  vtx9  
amfabra@coordinador:~/dev]$
```

En este caso le preguntamos a ls por todos aquellos comandos que tienen longitud 4 y el segundo carácter es una 't', ¿sencillo, verdad?

Puedes combinar tanto '\*' como '?' para conseguir el resultado deseado:

```
amfabra@coordinador:~/dev]$ ls ?l*1  
  
aloadC1  
  
amfabra@coordinador:~/dev]$
```

Lo que hemos hecho es decirle a 'ls' que nos muestre todos aquellos que de segundo carácter tienen una 'l' seguido de cualquier número de caracteres y acabe en 1. Quizás con 'ls' no le veáis mucha utilidad a estos comodines, pero imaginaos con cp, rm y mv por ejemplo. Podéis hacer tantas combinaciones como se os ocurran, así que a probar vosotros mismos.

### **Consultando información.**

Muchas veces necesitamos conocer más acerca de un comando determinado o del uso del propio intérprete y nos gustaría tener un manual, en el caso de GNU/Linux disponemos de un manual en línea muy útil denominado 'man'. Su sintaxis básica es la siguiente:

man [sección] comando

Por ejemplo para saber todo de 'cp' no hay mas que consultar 'man cp' nos moveremos con los cursores arriba o abajo y para salir pulsamos la letra 'q'. La barra espaciadora pasa página a página y podemos usar Re Pag y Av Pag de nuestro teclado para movernos.

El manual en línea está dividido en secciones, concretamente de la 1 a la 9, cada una referidas a una cosa distinta. Por ejemplo, la sección 2 es la de programación de C. Por defecto no pondremos sección y encontraremos lo que buscamos ya que busca en todas, pero en ocasiones (como por ejemplo cuando se programa) hay funciones que se llaman igual que algún comando de Linux y por defecto nos dará el comando en vez de la función. En ese caso, por ejemplo, le especificamos que queremos la sección 2. Para ver por vosotros mismos cómo se usa más a fondo 'man' probad con 'man man'.

Otra fuente útil de ayuda en línea es sin duda el comando 'info', que tiene un manejo para visualizar la información igual que 'man'.

### **Siguiendo tus propios pasos.**

Para no tener que escribir una y otra vez los mismos comandos, el intérprete de comando mantiene un histórico de las órdenes que introducimos. Para mostrarlas usad las teclas de cursor arriba y abajo.

Para ver todo el historial de golpe tenemos el comando 'history', que nos mostrará numeradas todas y cada una de las órdenes que le hemos dado al intérprete de comandos. Esto es especialmente útil para ejecutar directamente un comando del historial usando '!' seguido del número de orden visto con 'history'. Por ejemplo, para la orden 5 del historial ejecutad '!5'.

En el caso de bash el historial es guardado en nuestro propio directorio de trabajo, en el fichero '.bash\_history', en el cual podremos visualizar y manipular su contenido a nuestro gusto. (Recordad que es un fichero oculto).

### **¿Cuánto ocupa todo? 'du'.**

du [opciones] [ficheros...]

Este comando contabiliza el espacio que ocupa en disco un fichero o un directorio con todos sus subdirectorios. Como opciones sólo comentaremos 3 de ellas: la primera '-a' muestra además el espacio que ocupa cada uno de los ficheros que hay en los directorios que le indiquemos. La siguiente opción es '-b', que en vez de mostrarnos el tamaño en kB lo mostrará en bytes (1kB=1024bytes). Y por último la opción '-s', que informa solamente del directorio que le hayamos indicado sin contar sus subdirectorios.



```
amfabra@coordinador:~/curso]$ cd ..
amfabra@coordinador:~$ du curso
740  curso
amfabra@coordinador:~matados2k]$ du -a curso
4  curso/hola_holita
728  curso/perro
0  curso/soy_nuevo
4  curso/nueva_lista
0  curso/lista2
740  curso
amfabra@coordinador:~$ du -b curso
744412  curso
amfabra@coordinador:~$ du -s curso
740  curso
amfabra@coordinador:~$
```

Como observáis en el ejemplo, tanto 'du' a secas como 'du -s' dan el mismo resultado, y no es ni más ni menos porque el directorio 'curso' no contiene subdirectorios.

## **Creo que voy a mutar, 'su'.**

su [user]

Este comando lo que hace es convertir tu usuario actual al usuario que le indiques, por defecto con 'su' a secas lo que haremos es pedir que nos haga 'root'. Esto es de gran utilidad ya que no es necesario estar haciendo login/logout todo el rato, simplemente hay que darle la password del usuario al que queremos usar y para volver al anterior simplemente tecleamos exit o pulsamos Ctrl+D.

```
amfabra@coordinador:~/curso$ su
Password:
root@coordinador:~/curso# exit
exit
amfabra@coordinador:~/curso$ su triguelch
Password:
[triguelch@coordinador:/home/amfabra/curso$ su
Password:
root@coordinador:/home/amfabra/curso# exit
exit
[triguelch@coordinador:home/amfabra/curso$ exit
exit
amfabra@coordinador:~/curso$
```

Evidentemente no saldrá nada en pantalla cuando tecleamos los Password, pero sí que los está recibiendo. También podéis observar cómo puedo cambiar más de una vez seguida sin hacer exit.

### **El compresor de archivos File Roller.**

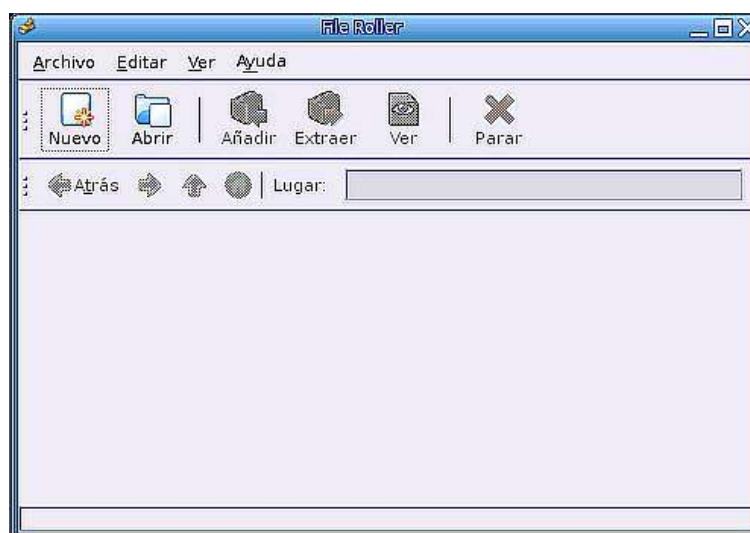
Un concepto muy frecuente en el mundo Linux es el concepto de paquete. Un paquete es un archivo cuyo contenido son otros archivos: los archivos que se han empaquetado. Los paquetes en Linux tienen extensión tar. Así, el archivo prueba.tar, será un paquete en cuyo interior hay otros archivos. Estos paquetes no están comprimidos, pero pueden estarlo. Los compresores más utilizados en linux son gzip y bzip2. Al comprimir un archivo con gzip este toma la extensión gz y al hacerlo con bzip2 la extensión bz2. Esto puede despistar un poco a los usuarios que proceden de Windows pues allí las extensiones de los archivos comprimidos son distintas: zip, rar, ace,...

Lo "normal" en Linux es que cuando queremos distribuir los archivos hagamos las dos cosas: empaquetarlo y comprimirlo, esto da lugar a nombres de archivos como prueba.tar.gz o prueba.tgz (paquete comprimido con gzip) o prueba.tar.bz2 (paquete comprimido con bzip2).

File Roller es un gestor de paquetes, es decir se encarga de empaquetar o

desempaquetar paquetes, comprimidos o no. Además soporta los formatos usuales de Windows como zip o rar. Veamos como funciona.

Podemos entrar en Fille Roller desde **Aplicaciones >> Accesorios >> File Roller**.

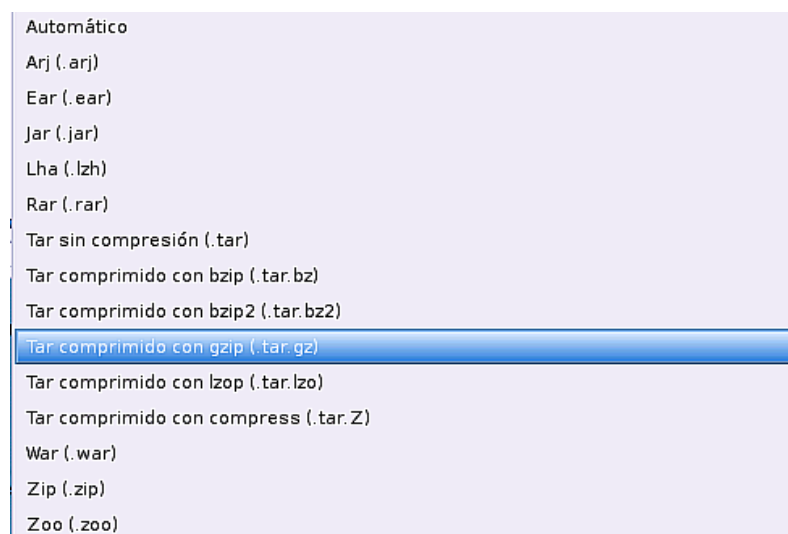


## Creando un paquete

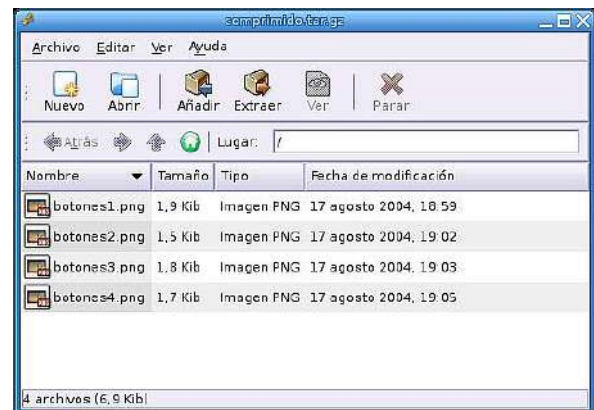
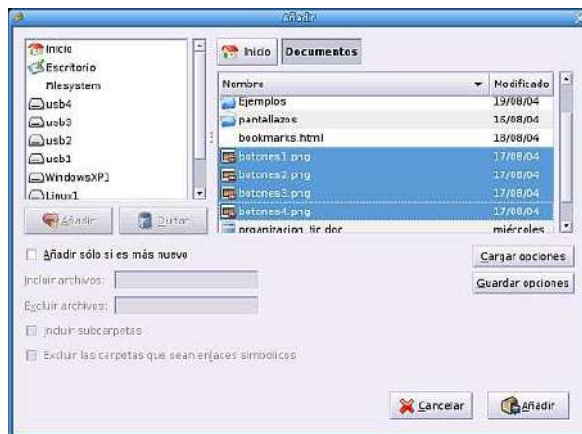
- 1.- Para crear un paquete seleccionamos Nuevo en la barra de botones.
- 2.- En la ventana de creación de un nuevo paquete debemos ponerle un nombre, seleccionar dónde queremos guardarlo y el tipo de paquete que deseamos crear.



- 3.- Si al nombre del fichero le hubiésemos puesto una extensión de un tipo conocido, File Roller crearía un paquete de ese tipo automáticamente. Ejemplo, si hubiésemos puesto '**comprimida.tar.gz**' crearía un paquete tar comprimido con gzip. Los tipos de paquete que podemos crear se muestran al hacer clic en la lista desplegable de Tipo de Paquete.



- 4.- Tras crear el paquete volveremos a la pantalla principal de File Roller donde, ahora, tendremos que pulsar sobre el icono **Añadir** para añadir archivos al paquete y se abrirá la ventana correspondiente. Podemos seleccionar los archivos que deseemos y, cuando finalicemos, pulsamos en **Añadir**.
- 5.- Volvemos de nuevo a la ventana principal del programa donde veremos los archivos que forman el paquete. Ya podemos cerrar File Roller, nuestro paquete ha sido creado.
- 6.- Si ahora vamos a la carpeta donde lo guardamos podremos verlo.



## Abriendo un paquete

Para ver el contenido de un paquete sólo tenemos que hacer doble clic sobre él, con lo cual File Roller lo abrirá. Si deseamos extraer todos o alguno de los ficheros que componen el paquete solo debemos seleccionarlo y hacer clic en el botón Extraer de la barra de botones.

Se abrirá la ventana Extraer desde la que podremos especificar dónde queremos guardar los archivos seleccionados.



Si tenemos activada la opción **Recrear carpetas**, se creará, en el directorio donde estemos extrayendo los archivos, todas las carpetas necesarias para mantener la estructura con que ha sido creado el paquete. Es decir, si por ejemplo, extraemos un archivo que, dentro del paquete, se halla en la carpeta exámenes; File Roller creará la carpeta exámenes, si no existe, y en ella extraerá el archivo. Si no quieres que esto ocurra, desmarca la opción.

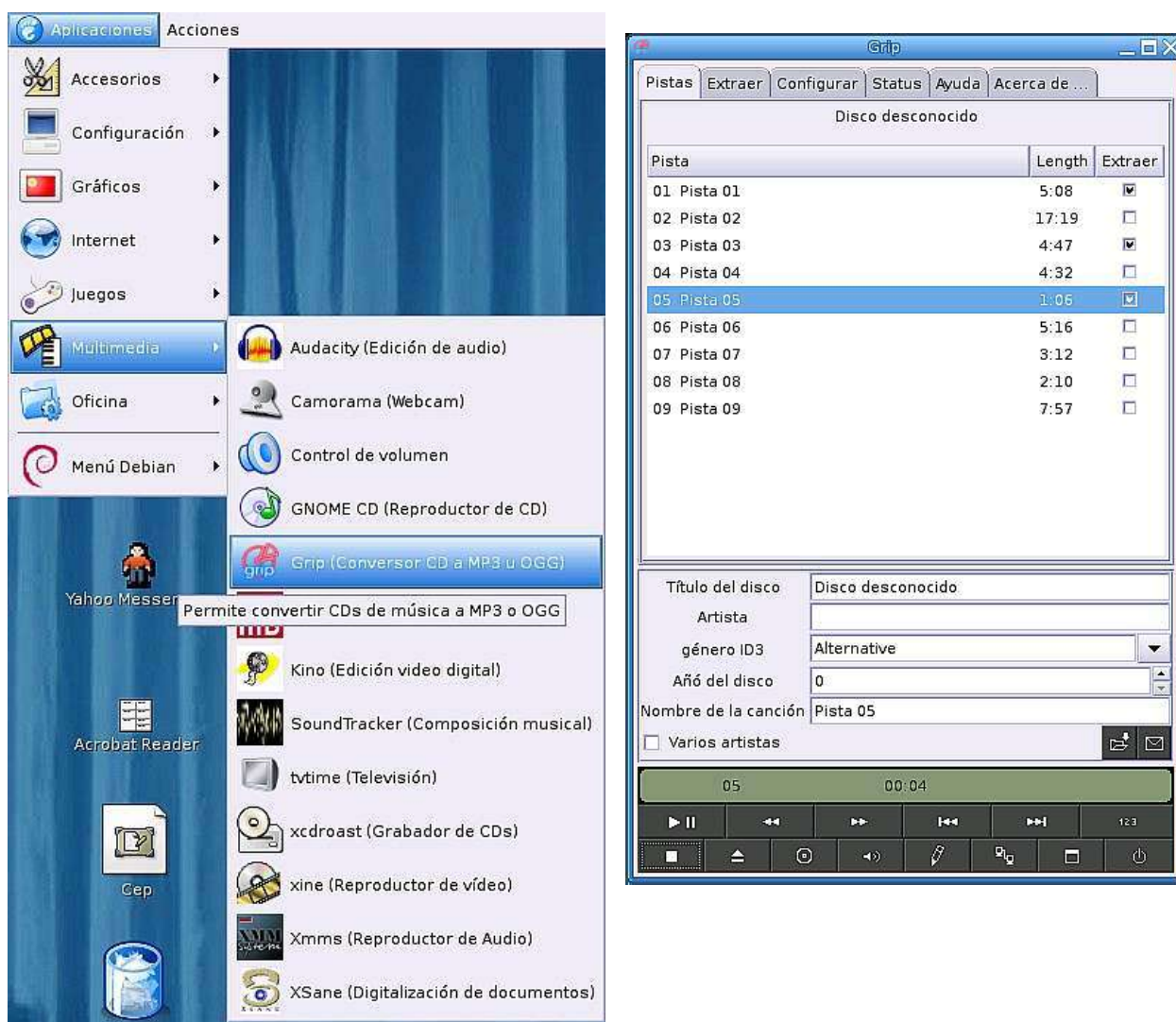
Si no puedes ver los archivos extraídos, refresca la pantalla de Nautilus. NOTA: Existe una posibilidad aún más cómoda de crear un paquete, para ello seleccionamos los archivos que compondrán el paquete, situándonos sobre la selección, hacemos clic sobre el botón derecho del ratón, seleccionando la opción **Crear paquete**.

## **El programa de extracción de audio Grip**

Guadalínex incluye un programa de extracción de audio llamado Grip. La utilidad de este programa es la posibilidad que ofrece de extraer las pistas de audio (ripear) de un CD de música al disco duro de nuestro ordenador. Las pistas podemos extraerlas sin comprimir, en formato wav, o bien extraerlas comprimidas (en formato mp3, ogg, ...). Posteriormente podremos grabar (con kb3) un CD a partir de estas pistas extraídas si así lo deseamos.

Para entrar en el programa lo hacemos en:

**Aplicaciones>>Multimedia>>Grip.**



El programa detecta el CD de audio que tenemos en el CD-ROM y nos muestra las pistas. Aquí seleccionamos las pistas que deseamos extraer

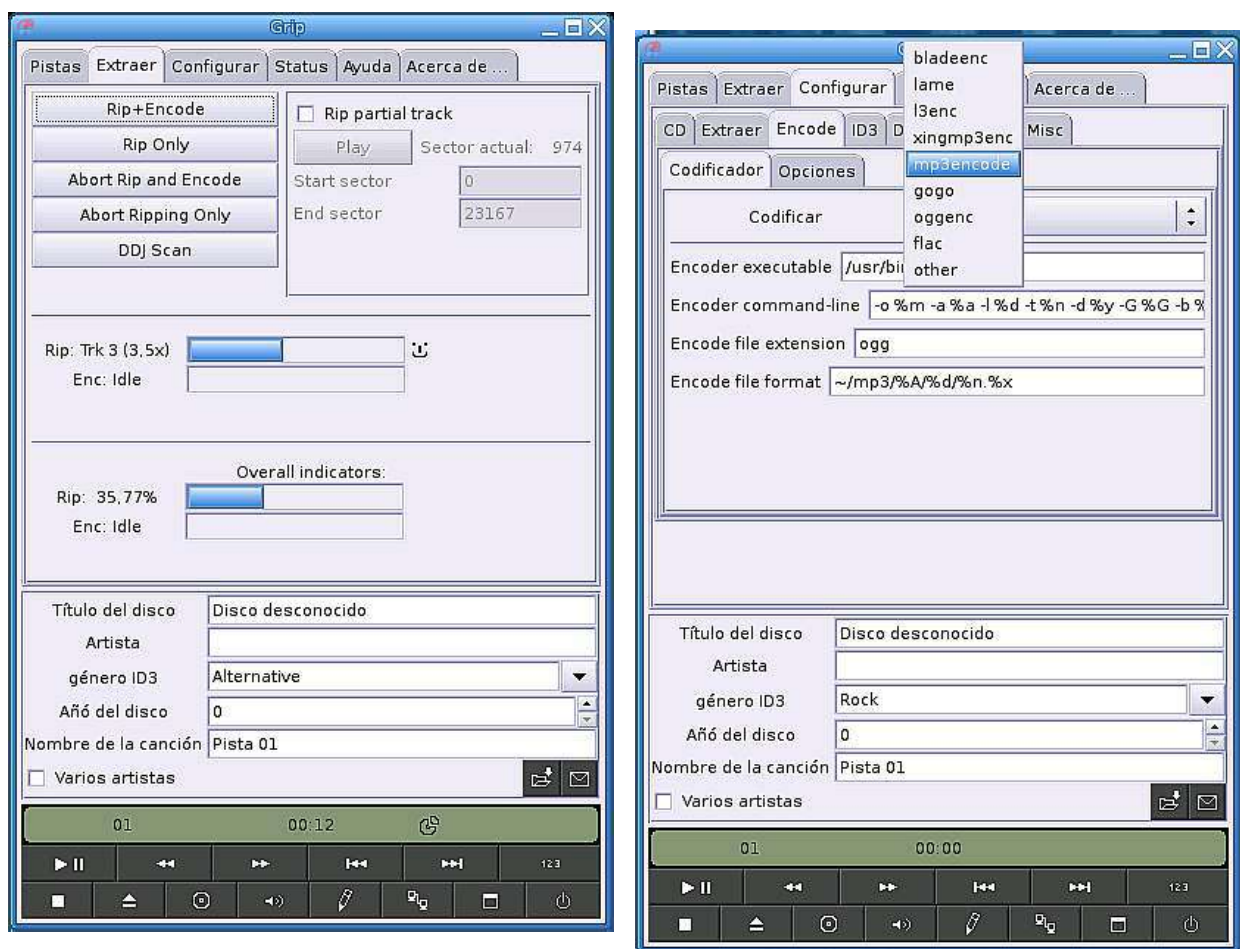


marcando la casilla correspondiente. También es posible modificar el nombre del disco y del artista, esto tiene su importancia porque por defecto las pistas se almacenarán en la carpeta

**`/home/usuario/mp3/nombre_artista/nombre_disco/`**.

Cuando hayamos completado la información que deseemos seleccionamos la pestaña **Extraer**.

En la pestaña **Extraer**, si seleccionamos **Rip Only** empezará la extracción de las pistas sin codificar, en formato wav; si seleccionamos **Rip+Encode** se extraerán y se codificarán. Por defecto la codificación se efectúa en formato “ogg”. Un nuevo formato libre mejor que el mp3.



Si deseas cambiar el tipo de codificación, selecciona la pestaña Configuración y en ella la pestaña Encode. Si seleccionas la lista desplegable Codificar verás las posibilidades de codificación que ofrece el programa.



Cuando hayas terminado con el "ripen" dirígete a la ventana correspondiente y veras las pistas extraídas. Si has extraído pistas sin comprimir y comprimidas puedes ver la diferencia de tamaño viendo los archivos como lista. Ahora escuchálos para ver si notas diferencia.

En el caso de que queramos almacenar la música en nuestro computador, el formato ideal es Ogg Vorbis. Pero si luego queremos poder utilizar en dispositivos portátiles, es muy probable que nos veamos obligados a utilizar el formato MP3.

Tanto uno como otro se pueden utilizar desde grip, una vez que hayamos instalado las herramientas auxiliares que necesita grip, que son en el caso de Ogg Vorbis las vorbis-tools que incluyen "oggenc" y en el caso de mp3, el codificado mp3enc, bladenc o lame. En nuestro caso vamos a utilizar "lame".

## Actualización. Instalación de Software

### **i.- Instalando paquetes deb.**

Un paquete deb es un archivo que tiene la extensión deb (extensión propia de los paquetes de Debian, otras distribuciones usan otros formatos para empaquetar sus archivos, como RED HAT que usa rpm). Instalar un paquete deb es muy sencillo, en primer lugar debemos conseguir el archivo (desde internet, un CD, ...). Vamos a ejemplificar este proceso instalando el messenger de yahoo en nuestro ordenador. Nos dirigimos a la página de yahoo desde la que podemos realizar la descarga:

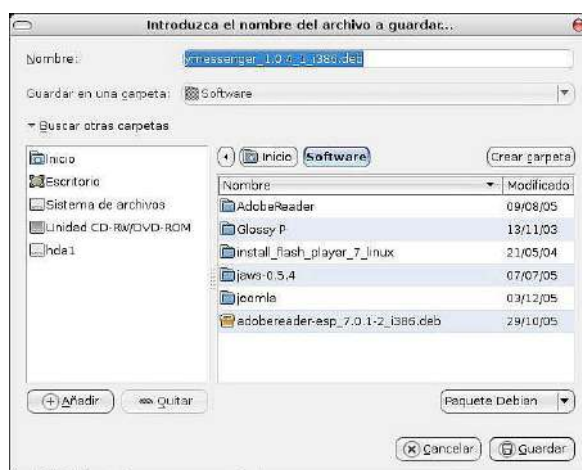
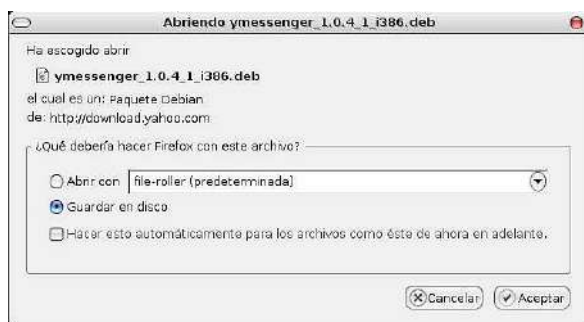
<http://messenger.yahoo.com/unix.php>

En la página podemos ver que el paquete está preparado para varias distribuciones, seleccionamos el de Debian.

En la propia página nos dicen como instalarlo. Pulsamos en **Save the file to your machine** para bajar el paquete a nuestro equipo:

#### **Debian Linux**

1. **Save the file to your machine.**
2. Log in as root and type: **dpkg -i ymessenger\_1.0.4\_1\_i386.deb** to install the application.
3. Run **/usr/bin/ymessenger** from X Window to launch the application.



Una vez el programa en nuestro equipo abrimos una terminal de root y nos dirigimos al directorio donde hemos almacenado el paquete (en este caso

/home/amfabra/Software). Para ello ejecutamos

```
# cd /home/amfabra/Software
# ls
```

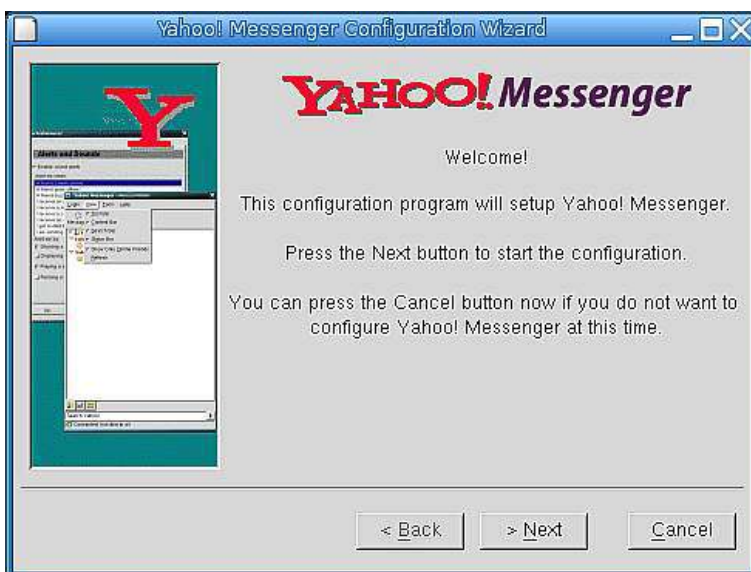
La primera de las instrucciones nos dirige al directorio en cuestión y la segunda muestra los archivos de dicho directorio para comprobar que el archivo está ahí.

Una vez situados en el directorio del paquete procedemos a instalarlo con la instrucción

```
# dpkg -i ymessenger_1.0.4_1_i386.deb
```

(dpkg -i es la orden para instalar paquetes)

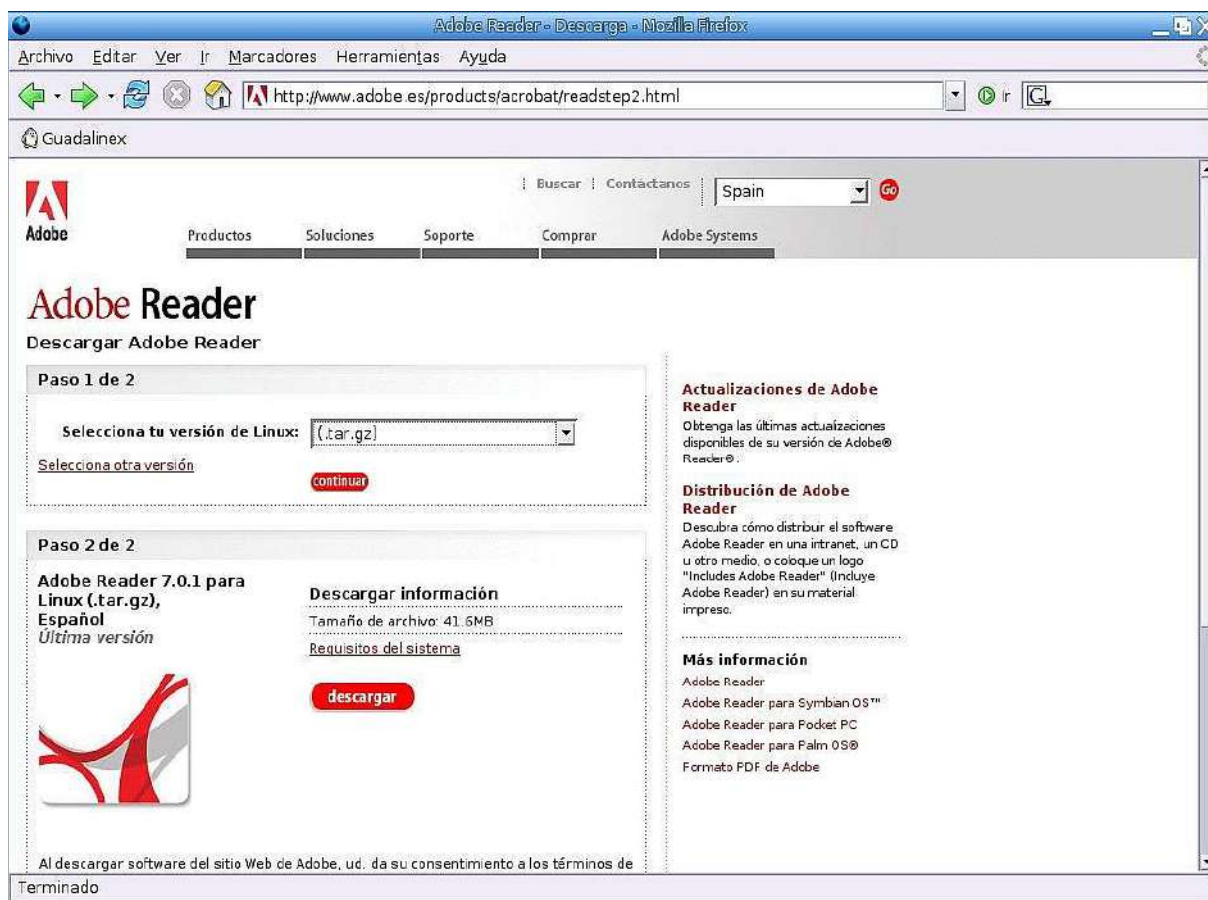
Ya podemos cerrar el terminal, ya está instalado. Lo único que hay que hacer es configurarlo, hay que darse de alta en un correo en yahoo.com si queremos usarlo, él solo crea un lanzador en nuestro escritorio al terminar de configurarlo.



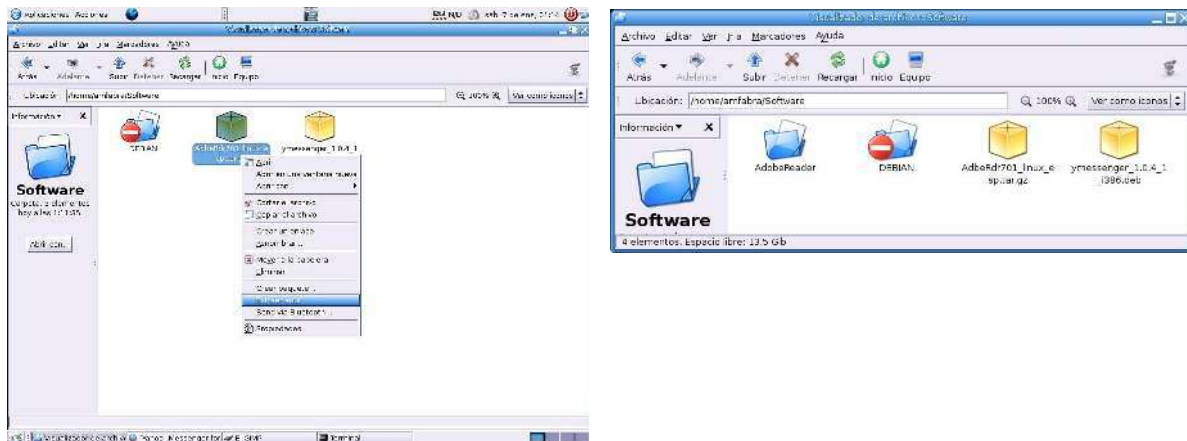
## ii.-Utilizando los instaladores que traen algunos paquetes.

A veces las aplicaciones no vienen como paquetes Debian (deb) ni están en los repositorios, si no que vienen como un paquete tar o tgz (recuerda la sección en la que hablamos de File Roller) que trae su propio script de instalación. Este es el caso, por ejemplo, del programa Acrobat Reader para Linux. Vamos a utilizar este programa para ejemplificar la instalación.

En primer lugar descargamos el programa desde la página de Adobe.



Una vez descargado (se ha descargado la versión 7 ya que en la 5 no funciona la rueda del ratón) abrimos el explorador de archivos Nautilus y nos dirigimos al directorio en que se encuentra el programa. Al hacer doble clic sobre él, se abre el programa File Roller, pues se trata de un paquete. Seleccionamos Extraer, creamos una nueva carpeta llamada Acrobat y extraemos el contenido a dicha carpeta.

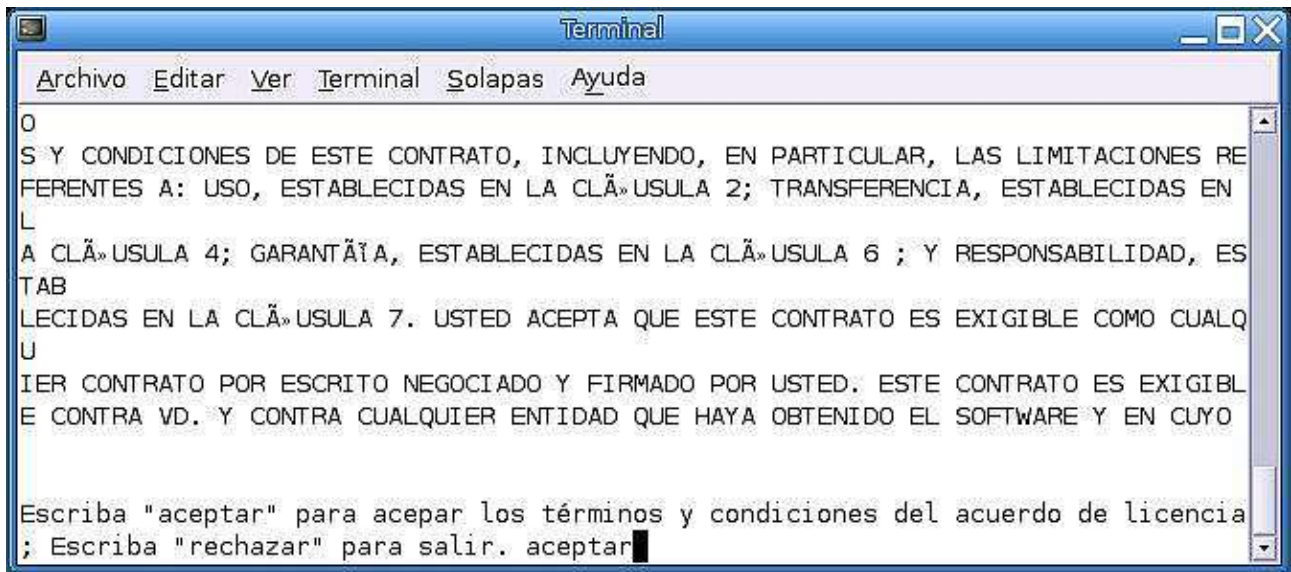


Ahora debemos abrir una terminal de root y situarnos en el directorio en el que hemos descomprimido el paquete (como en el caso del messenger de yahoo) y ejecutamos el script de instalación tecleando `./INSTALL` (¡¡Cuidado!! en este caso el nombre es con mayúsculas y así hay que introducirlo (recuerda que Linux diferencia las mayúsculas de las minúsculas en los nombres de archivos)).

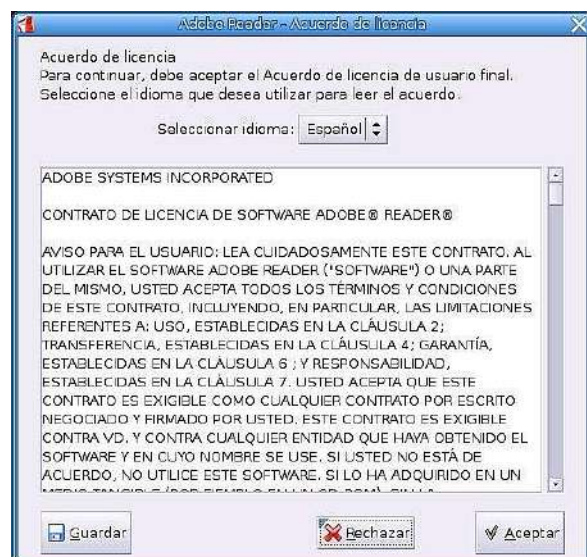
```
Terminal
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
root@coordinador:/home/amfabra/Software/AdobeReader# cd ..
root@coordinador:/home/amfabra/Software# cd ..
root@coordinador:/home/amfabra# cd Software/
root@coordinador:/home/amfabra/Software# ls
AdbeRdr701_linux_esp.tar.gz  AdobeReader  DEBIAN  ymessenger_1.0.4_1_i386.deb
root@coordinador:/home/amfabra/Software# cd AdobeReader/
root@coordinador:/home/amfabra/Software/AdobeReader# ls
COMMON.TAR  ILINXR.TAR  INSTALL  LICREAD.TXT  README
root@coordinador:/home/amfabra/Software/AdobeReader# ./INSTALL
```

En primer lugar debemos aceptar la licencia (pulsas `q` e irás directamente al final donde debes escribir `accept` y pulsar Intro).





Ya tenemos instalado el programa, pero seguro que aún no tenemos muy claro donde se encuentra. No importa, vamos a crear un lanzador al programa y en ese proceso veremos dónde está situado.



### iii.-Instalar desde un repositorio de paquetes.

Un repositorio de paquetes es una dirección de internet (o CD-ROM) en la que se encuentra una gran cantidad de aplicaciones para los sistemas Linux. Estas direcciones de internet desde las que podemos instalar software se encuentran almacenadas en el archivo `/etc/apt/sources.list`.

Una advertencia obvia, si deseamos realizar una instalación desde el repositorio de paquetes, ya sea con apt o con Synaptic, debemos estar conectados a Internet para que nuestro ordenador pueda ponerse en contacto con el repositorio.

Para efectuar la instalación del programa nombre-programa desde un terminal debemos identificarnos como root y ejecutar el comando

```
# apt-get install nombre-programa
```

Guadalínex se encargará de conectarse a internet y hacer todo lo necesario.

O utilizando el entorno gráfico que proporciona la aplicación Synaptic. Esta aplicación viene instalada por defecto en Guadalínex 2004. Para acceder a Synaptic en Guadalínex 2004 nos dirigimos a **Aplicaciones>>Configuración>>Sistema>>Sinaptic**

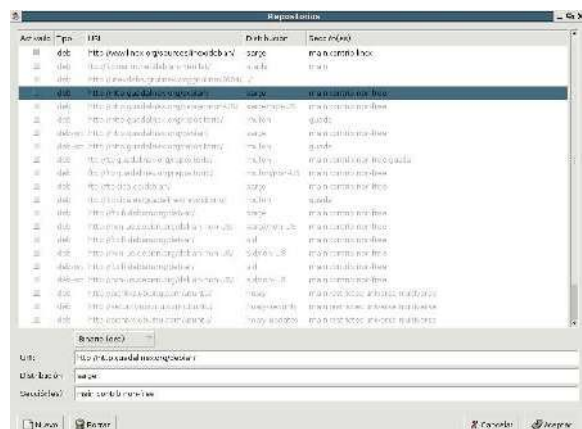
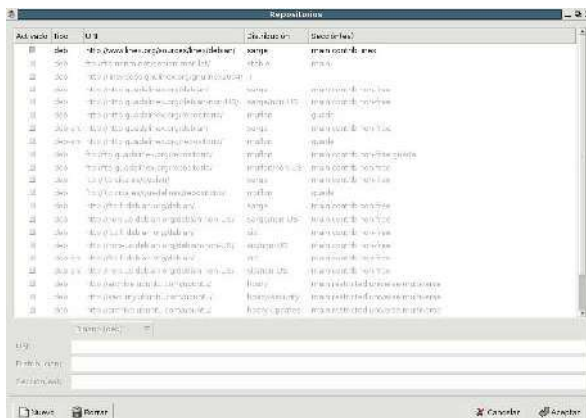


Deberemos introducir la contraseña de root. Tras lo cual entraremos en el programa.



## Manejando las fuentes.

Lo primero que vamos a aprender es cómo manejar las fuentes (acordaos de /etc/apt/sources.list, que es lo que realmente vamos a tocar pero de forma gráfica). Nos dirigimos al menú 'Configuración' y dentro de éste elegimos 'Repositorios', con lo que obtenemos la siguiente ventana:



Si recordamos cuando lo hacíamos manualmente, para comentar una



línea y que no tuviera efecto usábamos un '#'. Ahora simplemente desmarcamos una línea y listo, con lo que las fuentes efectivas son las que tenemos en negrita.

Para modificar una línea la seleccionamos y podremos editar su contenido en la parte inferior:

Para eliminarlas simplemente seleccionamos la que queremos y pulsamos sobre 'borrar', así que si quieres probar puedes pulsar sobre 'Nuevo' para crear una nueva entrada, probar a editarla y luego borrarla.

## Ya tengo a mi gusto las fuentes. ¿Y ahora qué?

Una vez terminado el paso anterior tendremos que hacer el equivalente a 'apt-get update' para sincronizar el índice de paquetes respecto a las fuentes, así que sin más pulsamos en el botón 'Recargar lista de paquetes' de la pantalla principal del programa, con lo que se nos abre un cuadro de diálogo de progreso tal que así:

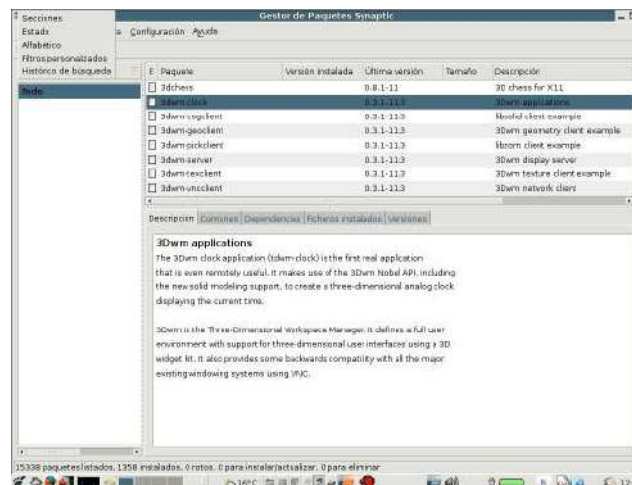


## Instalar, desinstalar, reinstalar y actualizar. Todo al alcance de un simple click.

Lo primero es saber qué queremos instalar, porque en la pantalla principal aparecen todos los paquetes disponibles y la verdad, así no hay quien se aclare (imaginaos mi caso con una lista de 16.302 paquetes entre Debian Sarge y Debian Sid). Lo primero que vamos a hacer es que se ordenen de alguna forma más lógica.

Para empezar nos fijamos arriba a la izquierda donde pone "Histórico de

búsqueda" (o cualquier otra cosa del siguiente recuadro), y pulsamos:

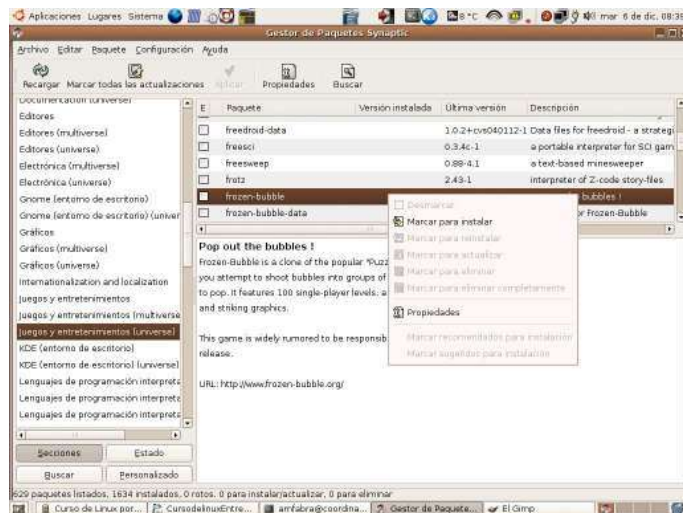
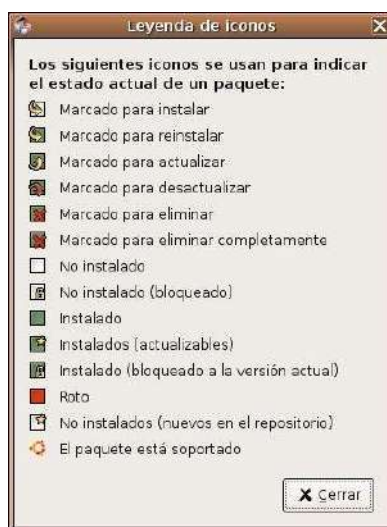


Vemos las posibilidades de ordenación que se nos dan, de las cuales la más interesante a mi modo de ver es por Secciones y quizás por orden Alfabético, la cosa es que investigues por tu cuenta. De momento, para la explicación elegiremos Secciones.

Estupendo, así si que da gusto, vemos en la parte izquierda muchas secciones muy ordenadas por temas, así que vamos a picar por ejemplo en... juegos y entretenimiento (según el sistema que tengáis tendréis uno u otros, o unos muy parecidos). Ahora vamos a buscar el frozen-bubble en la lista superior derecha y lo seleccionamos (no marcarlo), y vemos que nos aparece en la parte inferior izquierda la descripción del paquete, estupendo :) no sólo buscamos visualmente sino que podemos saber de qué se trata.



En la columna 'E' podéis ver un cuadradito que nos indica en qué estado están estos paquetes (instalado, sin instalar ...), los posibles estados los podemos ver en el menú ayuda y son los siguientes:



Pues bien, nos lo ponen todo en bandeja. Sabiendo esto sólo tenemos que marcar el paquete seleccionándolo y pulsando el botón secundario del ratón:

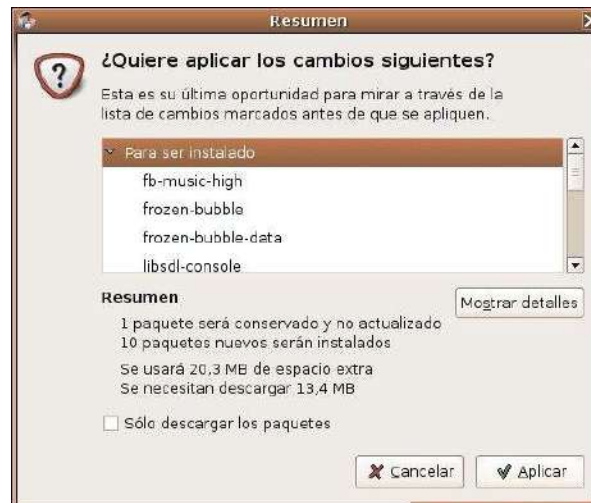
Estupendo, tenemos un todo en 1. Desde aquí, según el estado del paquete, aparecerán activas las opciones posibles a realizar: desmarcar (si estaba marcada) para no hacer nada, reinstalar un paquete, actualizarlo, eliminarlo y eliminarlo completamente (acordaros de que los paquetes 'deb' contenían ficheros de configuración que sólo se eliminaban con la opción `--purge` del 'apt-get').

Pues ya está, lo marco para instalar y automáticamente me marcará todas sus dependencias para instalarse :)



Ya sabemos cómo realizar todas las órdenes, pero pulsamos y no lo hace sino que 'lo marca', esto nos da la posibilidad de indicar todo tipo de acciones para luego pulsar en 'Aplicar' y ver cómo nuestros deseos se hacen realidad:

Vemos la información de lo que se realizará y nos da la opción de 'Sólo descargar los paquetes' como el '-d' de 'apt-get', 1,2,3.. YA



## ¿Y si tenemos un problema, qué?

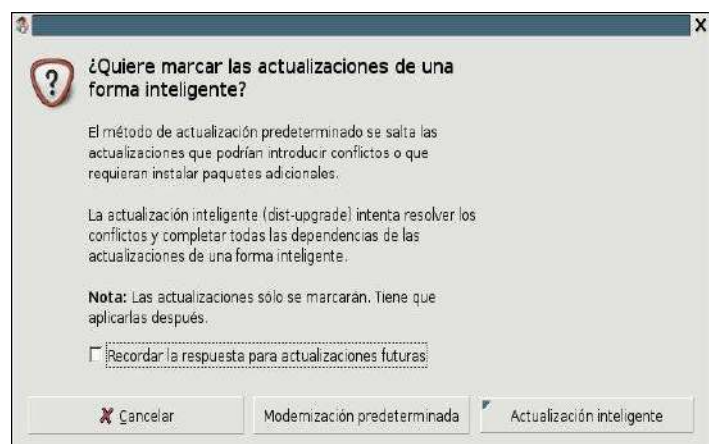
Pues la forma de solucionarlo es pulsando primero en 'Recargar' y luego ir al menú 'Editar' y seleccionar

'Reparar paquetes rotos' (equivalente del 'apt-get -f install').

## Y ahora queremos actualizar nuestro sistema.

Actualizar el sistema es también muy sencillo, pulsamos en 'Marcar todas las actualizaciones':

Y nos aparecen los 2 modos elegid la que os convenga sabiendo que 'Modernización predeterminada' equivale al



'apt-get upgrade' y 'Actualización inteligente' equivale a 'apt-get dist-upgrade'.  
Luego pulsáis 'Aplicar' y todo resuelto.

## ¿Y la caché? ¿Pero qué pasa con la caché?

Pues lo de la caché está más escondido, nos dirigimos al menú 'Configuración', elegimos 'Preferencias' y vamos a la pestaña de 'Ficheros temporales':

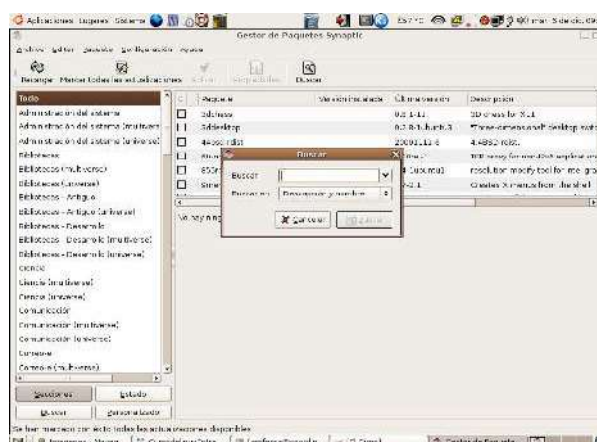


Simplemente dejamos marcada la opción que deseamos, la segunda corresponde a 'apt-get clean' y la tercera a 'apt-get autoclean'.

**Estupendo, ya sólo me queda realizar búsquedas.**

Esto es fácil, y seguro que si no os lo dijera también veríais ese enorme botón que pone 'Buscar':

Ya solo os queda seleccionar dónde buscar y... todos nuestros problemas solucionados.





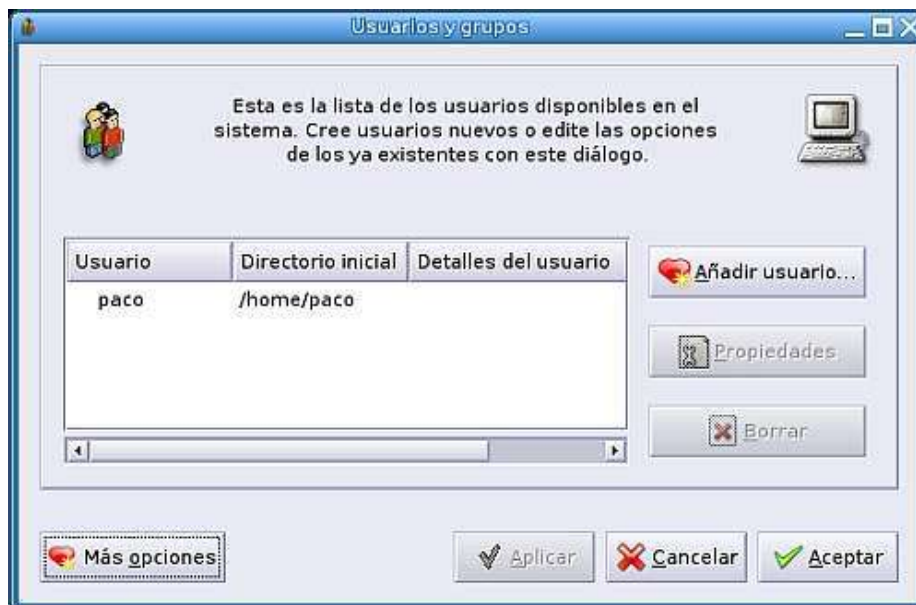
## **Añadir nuevos usuarios al sistema.**

Una de las características de Guadalínex es que es un sistema multiusuario, es decir, que varios usuarios pueden trabajar en el equipo sin que sus documentos puedan ser consultados por el resto de usuarios, cada uno tendrá un directorio de trabajo personal y protegido, en /home/usuario.

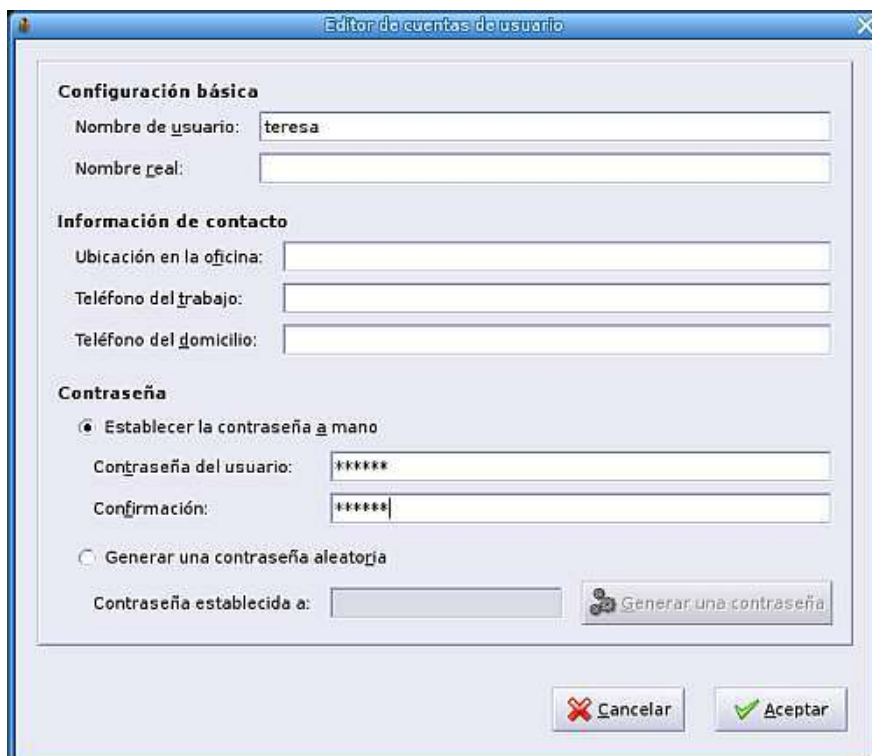
1. Para añadir un nuevo usuario al sistema nos dirigimos a **Aplicaciones>>Configuración>>Sistema>>Usuarios y Grupos.**



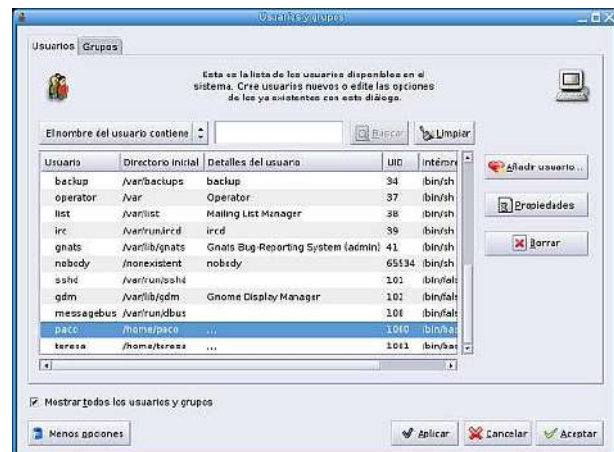
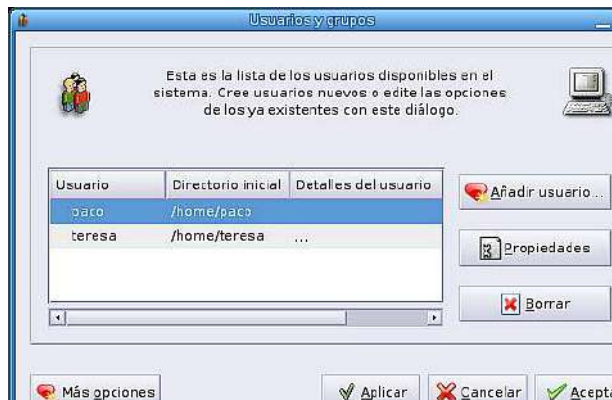
2. Una vez identificados como root entraremos a la aplicación. Como puedes ver, en este momento, sólo hay un usuario en el sistema, el usuario paco que creamos durante la instalación de Guadalínex2004. Para añadir un nuevo usuario al sistema pulsamos sobre **Añadir Usuario.**



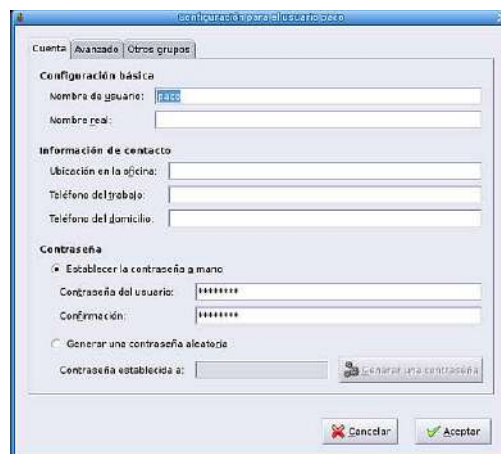
3. Tendremos que completar una ficha con los datos del usuario, basta con rellenar dos de ellos, el **nombre de usuario** (que debe ser corto, sin espacios en blanco ni símbolos especiales) y la **contraseña**.



4. Podemos ver ya al nuevo usuario del sistema.

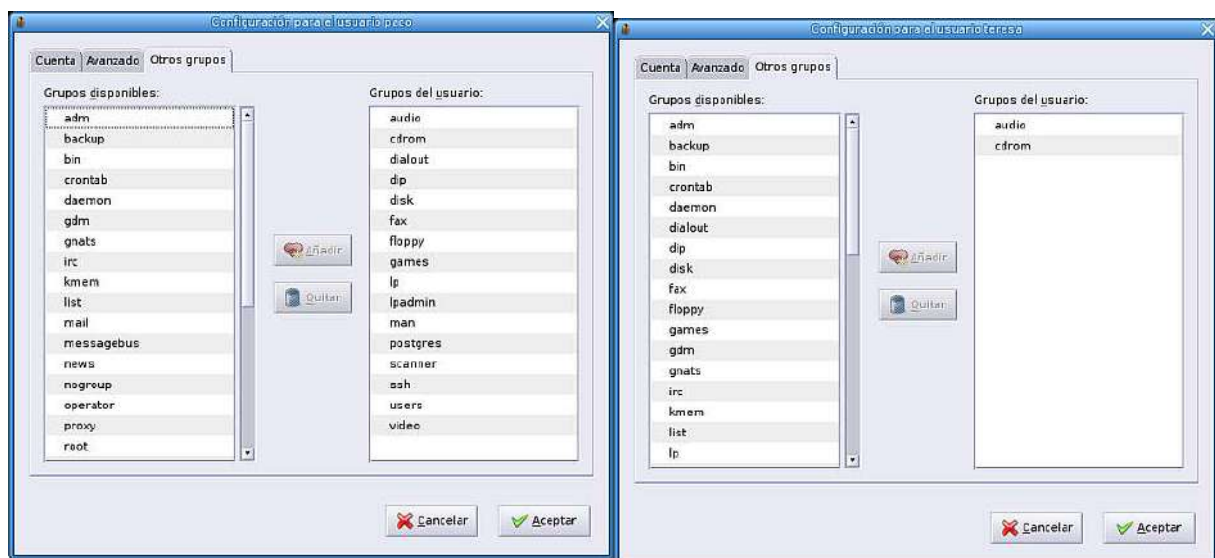


5. Pero, actualmente, este usuario, en Guadalínex 2004 tendrá problemas para hacer uso del sistema debido a la **política de grupos** de los sistemas Linux. No quiero entrar en detalles demasiado técnicos. Sólo decir que este usuario debe pertenecer a los mismos grupos que el usuario creado en la instalación para que pueda acceder a los mismos dispositivos (sonido, cdrom, ...). Por ello vamos a ver a los grupos que pertenece dicho usuario, en este caso 'paco', para incluir al nuevo usuario, 'teresa', en cada uno de ellos. Para ello pulsamos sobre **Más opciones**.
6. Como ves han aparecido muchos nuevos usuarios (entre ellos root), pero son usuarios que el sistema usa de *forma interna*, y que a nosotros, ahora, no nos interesan. Seleccionamos el usuario 'paco' y pulsamos en **Propiedades**.





7. La ventana contiene tres fichas, nos interesa la tercera, la que hace referencia a **Otros grupos** a los que pertenece el usuario
8. Toma buena nota de los grupos que son. Tendremos que dar de alta al nuevo usuario, teresa, en cada uno de ellos. Para ello volvemos a la ventana principal pulsando en **Cancelar** (o **Aceptar**). Una vez en la ventana principal seleccionamos al nuevo usuario, teresa, y seleccionamos **Propiedades**. Nos dirigimos a la pestaña de **Otros grupos** y añadimos todos los grupos a los que pertenecía paco seleccionándolos en la ventana de la izquierda y pulsando sobre el botón **Añadir**. Cuando hayamos finalizado pulsamos en **Aceptar** y en la ventana principal de nuevo en **Aceptar**.



9. Ya disponemos de dos usuario completamente independientes y con las mismas posibilidades de uso de Guadalínex.

*Apuntes completados de los cursos de Thales, realizados por Rafael Bracho López, Carlos Luque y Paco España Pérez y entregas de Luis García Galván*