

# Sistema Operativo GNU/Linux. Ubuntu

---

UD 7. Introducción en GNU/Linux. Tareas  
básicas.



## ÍNDICE

1.	interfaz textual: SHELL .....	2
1.1.	Comandos .....	3
1.2.	Shell-Scripts de Linux .....	5
2.	Administración de usuarios locales .....	10
2.1.	El fichero /etc/passwd .....	11
2.2.	Fichero /etc/shadow .....	11
2.3.	Directorio /etc/skel .....	11
2.4.	Creación de usuarios .....	11
2.5.	Eliminación de usuarios .....	14
2.6.	Cambiar la contraseña de un usuario .....	14
2.7.	Modificar cuentas de usuario .....	15
2.8.	Bloqueo de cuentas de usuario .....	15
3.	Administración de grupos locales .....	16
3.1.	Fichero /etc/group .....	17
3.2.	Creación, eliminación y modificación de grupos .....	17
3.3.	Añadir un usuario a un grupo .....	18
4.	Servicios .....	18
4.2.	En manera gráfica .....	19
4.3.	En manera comando .....	19
5.	Rendimiento y monitorización de sistema .....	19
5.1.	En manera Comando .....	20
5.2.	En manera gráfica: Monitor del sistema .....	20
5.3.	Archivos o registros de sucesos (logs) .....	20
6.	Gestión dispositivos hardware .....	21
6.1.	Ver dispositivo hardware .....	22
6.2.	Controladores privativos .....	22
6.3.	Gestión de discos .....	22
7.	Programación de tareas .....	23
7.1.	Comando At .....	24
7.2.	Comando Cron .....	24
8.	APLICACIONES DE MANTENIMIENTO Y OPTIMIZACIÓN .....	27
8.1.	FSLINT .....	28
8.2.	UBUNTU TWEAK .....	28
8.3.	BLEACHBIT .....	29
9.	Copias de seguridad .....	29
9.1.	Manera comandhueso .....	29
9.2.	Manera gráfica .....	30
10.	Gestor de arranque: grub .....	30
10.1.	Configurar el gestor de arranque .....	31
11.	ANEXOS .....	31
11.1.	Instalar Guest Additions en ubuntu .....	32



## 1. INTERFAZ TEXTUAL: SHELL

---

En Linux la interfaz de línea de comandos se conoce como Shell . La Shell Bash es la predeterminada en la mayoría de los sistemas Linux, como Ubuntu. Sin embargo, aparte de la Shell Bash, existen otras como sh, csh, ksh...

En Ubuntu se accede a la Shell a través del denominado **Terminal**. El terminal consiste en una pantalla en la cual podemos introducir comandos.

Podemos acceder a un Shell de varias formas:

- Menú Aplicaciones del Dock → Terminal
- Pulsando simultáneamente las teclas Ctrl+Alt+T

En general, todas las órdenes de UNIX/Linux son programas que están almacenados en el sistema de ficheros. Su sintaxis es similar a la del entorno comando de Windows y es la siguiente:

**orden [-opciones] [argumentos]**

Antes de empezar a trabajar con los comandos de la Shell, tengas en cuenta los siguientes **consejos**:

- Para trabajar más rápida y cómodamente con el Shell, puedes utilizar los siguientes **atajos**:
  - o Puedes **completar** los comandos usando la tecla **tabulador**
  - o Puedes **reescribir** comandos ya utilizados usando las **teclas** ↑ y ↓.
- Puedes **detener** la ejecución de un comando pulsando **Ctrl + c**
- Mediante el comando **sudo** (que tendrás que escribir antes del comando) puedes **ejecutar comandos** que requieran **privilegios de root**.
  - o El fichero **/etc/sudoers** contiene qué **usuarios o grupos** pueden **usar el comando sudo**
- Mediante la orden **su**, puedes **pasar a ser otro usuario**. Por ejemplo:
  - o `sudo su` → te conviertes en el usuario root
  - o `sudo su mar` → te conviertes en el usuario mar

El **prompt** del Shell bash muestra dos **formas** diferentes según el tipo de usuario que lo utilice:

- Si se trata de un usuario **estándar** → finaliza con el carácter **\$**
- Si se trata de la cuenta de superusuario o root del sistema → finaliza con el carácter **#**

### 1.1. COMANDOS

A continuación, se indica un resumen de los principales comandos:

COMANDO	USO	QUÉ HACE	EJEMPLO
<b>Comandos para manejo de la interfaz</b>			
Man help	man comando comando --help	Información de ayuda de los comandos	man ls muestra el manual de uso del comando ls (para salir de man pulsa q) ls -help
echo	echo texto	Muestra un texto literal por pantalla	echo hola Muestra el texto hola
clear	clear	Borra la pantalla	clear
exit	exit	Salir del shell	exit

<b>Comandos generales de configuración del sistema</b>			
Date	Date	ver/poner la fecha del sistema	date muestra la fecha y hora actual date -s establece la fecha que le indicamos
Cal	Cal	Muestra un calendario	Cal -m 4
shutdown	shutdown [-opciones] tiempos [mensaje]	Apaga el sistema	shutdown 1 apaga el sistema en 1 minuto shutdown -h now apaga el sistema y el equipo shutdown -t seco: Espera t según antes de iniciar el proceso de cierre del sistema shutdown -r: reinicia (reboot) después de shutdown.
reboot	reboot	Reinicia el sistema	reboot reinicia el sistema

<b>Comandos de tratamiento de unidades de disco</b>			
du	du [-opciones][fichero]	Muestra un resumen del uso de disco para cada fichero o directorio. Recursivo para directorios.	Du -h muestra los tamaños de manera legible (p.ej: 1K 234M 2G)
df	df [-opciones][fichero]	Muestra información sobre el sistema de ficheros en el cual reside cada fichero. Por defecto, de todos los sistemas de ficheros.	df -h muestra los tamaños de manera legible (p.ej: 1K 234M 2G)

<b>Comandos de manejo de ficheros y directorios</b>			
touch	touch fichero(s)	Crea un fichero vacío	touch ejercicio1.txt crea el fichero vacío ejercicio1.txt
vi nano	vi ver fichero nano fichero	Crear o modificar un fichero	vi ej1.txt. nano ej1.txt
cat more less	cat fichero(s) more fichero less fichero	Ver el contenido de un archivo	cat ej1.txt muestra por pantalla el contenido del fichero ej1.txt more ej1.txt muestra el contenido del fichero de forma paginada (pulsando intro avanza una línea y pulsando la barra espaciadora avanza una página)
grep	grep [-opciones] patrón [fichero]	Muestra las líneas de un fichero que contienen un determinado patrón (cadena de texto)	grep root /etc/passwd muestra las líneas del fichero /etc/passwd que contienen la palabra root
head	head [-opciones] fichero	Muestra las 10 primeras líneas del fichero	head -n 5 index.php muestra las 5 primeras líneas del fichero
tail	tail [-opciones] fichero	Muestra las 10 últimas líneas del fichero	tail -n 15 messages.txt muestra las ultimas 15 líneas del fichero
wc	wc [-opciones] fichero	Muestra n.º de líneas, palabras o caracteres	wc -l muestra el n.º de líneas del fichero wc -w muestra el n.º de palabras
cut	cut [-opciones] fichero cut -d -f fichero	Muestra solo ciertas columnas del fichero	Suponemos texto.txt contiene el texto “Este es, un ejemplo, del comando cut” cut -d, -f1,2 texto.txt muestra por pantalla “Este es, un ejemplo” Con -d indicamos el delimitador de columnas (la ,) y con -f indicamos las columnas a mostrar.
sort	sort [-opciones] fichero	Ordena las líneas de un fichero	sort fichero ordena el contenido del fichero
pwd	pwd	Muestra la ruta completa del directorio actual	pwd
ls	ls [directorio] ls -l ls -a ls -r ls -i	Ver el contenido de un directorio	ls SI muestra todos los ficheros y directorios del directorio SI ls -l lista los archivos con su información detallada ls -a lista todos los archivos, incluidos los archivos ocultos ls -r SI lista el contenido del directorio SI y de todos sus subdirectorios (de forma recursiva) ls -i muestra el nodo-i
cd	cd [directorio]	Cambiar de directorio	cd /home/mar/SI cambia al directorio SI usando una ruta absoluta cd ./SI cambia al directorio SI usando una ruta relativa cd .. cambia al directorio padre cd ~ cambia al directorio home del usuario

mkdir	mkdir directorio mkdir -p	Crear un directorio	mkdir SI crea un directorio llamado SI mkdir -p SI/PRACTICAS crea todos los directorios que conforman la ruta que le damos, es decir, crea el directorio SI y dentro de este, el directorio PRACTICAS
rm	rm fichero(s) rm -r directorio(s)	Eliminar un archivo	rm ./SI/s borra todos los ficheros del directorio SI que empiezan por el carácter s (borra solo los ficheros, no subdirectorios)
		Eliminar un directorio	rm -r ./SI/SUBDIR1 borra el directorio SUBDIR1 completo (con sus ficheros y directorios)
cp	cp fichero(s) directorio cp -r fichero(s) directorio(s) directorio	Copiar un archivo	cp ejercicio1.doc SI copia el archivo ejercicio1.doc al directorio SI cp ej ./SI copia todos los ficheros del directorio actual que empiezan por "ej" al directorio SI cp ej? ./SI copia todos los ficheros de tres caracteres del directorio actual que empiezan por "ej" al directorio SI
		Copiar un archivo o directorio	cp -r ./SI/practica1 ./SI/PRACTICAS copia la carpeta practica1 dentro de la carpeta PRACTICAS
mv	mv fichero(s) directorio mv fichero1 fitxer2	Cambiar el nombre o mover un archivo o directorio	mv ej1.txt pr1.txt ./tema1 mueve los ficheros ej1.txt y pr1.txt al directorio tema1 mv ej1.txt ej2.txt cambia el nombre del fichero ej1.txt a ej2.txt
ln	ln [-opciones] fichero enlace ln -s	Crea un enlace a un fichero	ln crea un enlace duro
			ln -s crea un enlace simbólico
file	file fichero	Determina el tipo de un fichero	file ej1.txt muestra el tipo del fichero ej1.txt
chmod	Chmod o g o a  = + - r w x fichero(s) directorio(s) Chmod XXX fichero(s) directorio(s)	Establecer permisos en un archivo o directorio	chmod o=rwx, g=rx, o=- ej1.txt
			chmod 750 ej1.txt chmod g-wx ej1.txt
chown	chown [-opciones] propietario:[grupo] fichero	Cambia el amo y/o grupo del fichero	chown mar:profesores examen.doc el nuevo propietario del fichero examen.doc es mar y su nuevo grupo es profesores
tar	tar [-opciones] [fichero] tar -cvf fichero.tar fichero(s) directorio(s) tar -tf fichero.tar tar -rvf fichero.tar fichero(s) directorio(s) tar -xvf fichero.tar tar -tfz fichero.tar	Empaquetar ficheros y directorios (no comprime, solo empaqueta)	-c crea un nuevo fichero
			-x extrae archivos de un fichero -t lista el contenido del fichero -v verbose -z comprime o descomprime -f guarda en el fichero indicado
gzip	gzip [-opciones] fichero gzip fichero(s) directorio(s) gzip -d fichero.gz gzip -r directorio	Comprimir y descomprimir un archivo o directorio (reduce el tamaño de los ficheros)	gzip prac1.doc comprime el archivo prac1.doc que pasa a decirse prac1.doc.gz gzip -d prac1.doc.gz descomprime el archivo prac1.doc.gz gzip -r Comprime los ficheros y subcarpetas de un directorio

### Direccionalientos → >, >>, <, 2>, 2>>, &>

DIRECCIONAMIENTO	QUÉ HACE	EJEMPLO
comando > fichero	crea un fichero nuevo cuyo contenido es el resultado del comando de la izquierda (si ya existía, lo destruye)	echo hola > saludo.txt crea un fichero saludo.txt cuyo contenido es la palabra hola
comando >> fichero	añade al final del fichero (sin borrar nada de lo que había antes) el resultado del comando de la izquierda	echo que tal >> saludo.txt
comando < fichero	Introduce como datos de entrada el fichero	sort < fichero_a_ordenar.doc
comando 2> fichero	Envía la salida de error de comando a fichero	ls -l 2> fichero Si hay un error, lo guarda en fichero (puede dar error si no tenemos permiso de lectura en el dir).
comando &> fichero	Envía la salida estándar y de error a fichero	cat datos1 datos2 &> datos

### Tuberías (pipes) → |

Uso: comando1 | comando2

Permite conectar la salida de un comando con la entrada del siguiente. Ejemplo: cat /etc/passwd | grep mar

### Variables de entorno del shell

Se utilizan para guardar información del sistema y del usuario. Entre las más usadas encontramos:

- \$HOME: Directorio personal.
- \$HOSTNAME: Nombre de la máquina.
- \$USER: Nombre del usuario.
- \$PWD: Directorio de trabajo actual.

De este modo, puedes escribir echo \$HOSTNAME para saber el nombre de tu equipo.

## 1.2. SHELL-SCRIPTS DE LINUX.

Equivalen los archivos .BAT de MSDOS.

Son ficheros de texto que **contienen comandos** que se irán ejecutando por la Shell.

Para crear un fichero Shell script puedes usar: gedit, nano, vi, joe, vim, pico o emacs, entre otros.

**Porque** un script pueda **ejecutarse**, hay que **añadirle** al fichero de texto **permisos de ejecución**.

Cuando basura Shell scripts hay que decirle al sistema con qué Shell queremos ejecutar el script. Por ejemplo, para el Shell Bash escribiríamos en la primera **línea** del script:

**#!/bin/bash**

También es frecuente añadirle la extensión **.sh** para indicar al usuario que se trata de un Shell script.

### A. Operaciones básicas

- **Tabulador (Tab completion)**  
Sirve para completar el comando, sólo hay que escribir una letra y pulsar la tecla tabulador.
- **Tubería (|)**  
El primer comando se convierte en la salida del segundo comando.
- **Comentario (#)**  
Texto que no se ejecuta.
- **Unir varios comandos seguidos (;)**  
Para ejecutar varios comandos seguidos.
- **Ayuda sobre un comando**  
man Comando
- **Listado de todos los comandos**  
Pulsar dos veces la tecla tabulador.
- **Propiedades de un comando**  
man Comando

### B. Variables

- **Inicializar una variable**  
Comando: VARIABLE=Valor

*Ejemplo:*  
MENSAJE="Hola Mundo"  
echo \$MENSAJE

- **Uso de variables especiales:**

Variable	Funció
\$0	Nom del Shell script
\$1, \$2, \$3...	Paràmetres o arguments que s'introdueixen desde la línia de comandos
\$#	Número de paràmetres o arguments
\$*	Valor de tots els arguments
\$?	Valor tornat per l'últim comando executat.
\$\$	PID del Shell script

### C. Arrays

- **Inicializar un array**  
Comando: Valor=(Valor1 Valor2 ValorN)

*Ejemplo:*  
\$Valor=(1 2 3); echo \${Valor[1]}

### D. Consola

- **Escribir en consola**  
Comando: echo "hola"

*Ejemplo:*  
Hola

- **Leer en consola**

Comando:  
read palabra  
echo \$palabra

*Ejemplo:*  
echo "Escribir palabra:"; read palabra;  
echo "Palabra escrita:" \$palabra  
Resultado:  
Escribir palabra:  
Hola  
Palabra escrita: Hola

### E. Funciones

- **Declaración de la función**  
Function Nombre\_de\_funcion()  
{  
CÓDIGO\_A\_EJECUTAR  
}  
  
• **Llamada a la función**  
Nombre\_de\_la\_función Valores

*Ejemplo:*  
Declarar función suma:  
function suma(){let a=2; let b=1; let c=b+a;  
echo \$c}  
Llamar a la function suma:  
suma  
Resultado  
3

### F. Ficheros

- **Escribir en fichero**  
Comando: echo "Texto" > fichero

*Ejemplo:*  
echo "Hola" > fichero

- **Leer de fichero**

Comando:  
while read linea;  
do CÓDIGO\_A\_EJECUTAR;  
done < ruta

*Ejemplo:*  
while read linea; do echo -e "\$linea";  
done < fichero

### Estructuras de control: condicionals y bucles

Les estructuras de control les usarem perquè, depenent d'un determinat valor d'una variable o resultat d'una condició, s'executen uns comandos o altres, o bé perquè s'executen uns comandos un nombre determinat de vegades, segons l'expressió que usem.

#### if-then-fi

```
if [ condició ]  
then  
    comando/s  
fi
```

#### if-then-else-fi

```
if [ condició ]  
then  
    comando/s  
else  
    comando/s  
fi
```

#### if-then-elif-then-else-fi

```
if [ condició ]  
then  
    comando/s  
elif [ condició ]  
then  
    comando/s  
else  
    comando/s  
fi
```

#### case-in-esac

```
case $variable in  
    expr1)comando/s;;  
    expr2)comando/s;;  
    ...  
    *) comando/s;;  
esac
```

#### for-in-do-done

```
for $variable in valores  
do  
    comando/s  
done
```

#### while-do-done

```
while [condició]  
do  
    comando/s  
done
```

### Expressions condicionals

Per a avaluar el valor retornat per la condició, usarem les següents expressions dins de claudàtors [ ].

Condició	Torna vertader (true) si...
-f \$variable	... variable és un fitxer
-d \$variable	... variable és un directori
-r/w/x \$variable	... variable té permís de r/w/x
-e \$variable	... variable és un fitxer que existeix
\$var1 -nt \$var2	... var1 és un fitxer més nou que var2
\$var1 -ot \$var2	... var1 és un fitxer més antic que var2
-z \$variable	... variable és una cadena buida
-n \$variable	... variable és una cadena no buida
\$var1 = \$var2	... var1 és una cadena igual que var2
\$var1 != \$var2	... var1 és una cadena distinta de var2



## Ejemplos de Shell scripts:

- a) Típico script “hola mundo”:

```
#!/bin/bash
echo Hola Mundo
```

En este script, se muestra el mensaje Hola Mundo en la terminal

- b) Script de copia de seguridad muy simple:

```
#!/bin/bash
tar -czf /var/my-co pia-de -seguridad.tgz /home/usuario/
```

En este script, creamos una copia comprimida del directorio home de un usuario.

- c) Script que borra la pantalla y muestra la fecha actual:

```
#!/bin/bash
#borra la pantalla
clear
#Muestra la fecha
Date
```

- d) Script en el cual se hace uso de “if”:

```
#!/bin/bash
echo -e “Escribe el nombre de un fichero: \c”
read fichero
if [ -f $fichero ]
then
    ls -l $fichero
else
    echo “Error: $fichero no es un fichero”
fi
```

El script pregunta el nombre de un fichero. Si el fichero existe, muestra la información sobre él en formato largo. Si no existe o no es un fichero muestra un mensaje de error.

- e) Script en el cual se hace uso de “for”:

```
#!/bin/bash
echo -e “Escribe el nombre de un fichero: \c”
read fichero
for var in
do
    if [ $var -nt $fichero ]
    then
        echo “$var es más reciente”
    fi
done
```

Este script te pregunta por el nombre de un fichero. Si existe, para cada fichero del directorio personal, te dirá si es más reciente o no que el fichero que has escrito.



- f) Script en el cual se hace uso de “for”:

```
#!/bin/bash
echo "Introduce el directorio a analizar:"
read var
líneas='ls $var | wc -l'
echo "El directorio tiene $líneas elementos"
for elemento in `ls $var`
do
    if [ -d $elemento ]
    then
        echo "$elemento es un directorio"
    elif [ -f $elemento ]
    then
        echo "$elemento es un fichero"
    else
        echo "$elemento no es ni un fichero ni un directorio"
    fi
done
```

Este script pide al usuario una ruta de directorio e indica para cada uno de los elementos que tiene este directorio, si se trata de un fichero, un directorio u otra cosa.

- g) Script en el cual se hace uso de “while”:

```
#!/bin/bash
if [ $# -eq 0 ]
then
    echo "Error, no hay parámetros"
else
    while [ $# -ne 0 ]
    do
        if [ -f $1 ]
        then
            echo "$1 es un fichero"
        elif [ -d $1 ]
        then
            echo "$1 es un directorio"
        else
            echo "$1 no es un fichero ni directorio"
        fi
        shift
    done
fi
```

Este script recibe una serie de parámetros por la línea de comandos, los muestra y te dice si son ficheros o directorios.

## Más ejemplos de Shell scripts:

- h) Realiza un script denominado **archi**, al cual haya que suministrarle un parámetro que será la dirección absoluta de un archivo.
- SI el archivo existe
  - ENTONCES lo presentará en pantalla (cat )
  - SINO lo creará y añadirá la frase "Esto es un archivo nuevo"

Lo ejecutaremos así: `./archi /home/usuario/mifichero`

```
#!/bin/bash
if [ -f $1 ]
then
    cat $1
else
    echo "Esto es un archivo nuevo" > $1
fin
```

- i) Realiza un script denominado **dire**, al cual haya que suministrarle un parámetro que será la dirección absoluta de un directorio.
- SI el directorio existe
  - ENTONCES lo listará (ls -ali \$1)
  - SINO indicará que "No existe" y lo creará (mkdir \$1)

Lo ejecutaremos así: `./dire /home/usuario/micarpeta`

```
#!/bin/bash
if [ -d $1 ]
then
    ls -ali $1
else
    echo "No existe"
    mkdir $1
    echo "El directorio ha sido creado"
fin
```

- j) Realiza un script llamado **carné**, al cual haya que suministrarle dos parámetros. Un serán un nombre de persona y el otro la edad.
- SI es mayor o igual (-ge) de 18
  - ENTONCES escribirá el nombre y que es mayor de edad
  - SINO escribirá el nombre y que no es mayor de edad

Lo ejecutaremos así: `./carné juan 23`

En este caso: \$1 es juan y \$2 es 23

```
#!/bin/bash
if [ $2 -ge 18 ]
then
    echo $1 eras mayor de edad, tenso $2
    echo Puedes sacarte el carné
else
    echo $1 eras menor de edad, tenso $2
    echo No puedes tener carné
fin
```

k) Realiza un script denominado **fichor**, que solicite la dirección absoluta de un fichero

- SI el fichero existe
- ENTONCES lo presentará en pantalla (cat )
- SINO presentará la frase "El fichero no existe"

```
#!/bin/bash
echo "Introduce la direccion absoluta de un archivo"
read ruta
if [ -f $ruta ]
then
    cat $ruta
else
    echo "El archivo no existe"
fi
```

l) Realiza un script llamado **día**, el cual solicite que introduzca un día de la semana: lunes, martes, miercoles, jueves, viernes, sabado, domingo. Según el día introducido saldrá una frase.

```
#!/bin/bash
echo "Introduzca un día de la semana"
read día
case $día in
    lunes) echo "El lunes es una lata" ;;
    martes) echo "El martes es el segundo día de la semana" ;;
    miercoles) echo "Hoy es miercoles" ;;
    jueves) echo "Siempre en medio" ;;
    viernes) echo "Ya es finde" ;;
    sabado) echo "A salir" ;;
    domingo) echo "Domingo pingo" ;;
esac
```

m) Realiza un script denominado **comprime**, el cual solicite que escriba la dirección absoluta de un archivo.

- SI el archivo existe
- ENTONCES
  - o SI la carpeta "comprimidos" existe
  - o ENTONCES guarda el archivo comprimido en esa carpeta
  - o SINO crea la carpeta "comprimidos" y guarda el archivo
- SINO sale un aviso que ese archivo no existe

Nota: al comprimirlo no se tiene que eliminar el archivo original.

```
#!/bin/bash
echo Introduce la dirección absoluta de un archivo
read archi
if [ -f $archi ]
then
    if [ -d /home/utele/comprimidos ]
    then
        gzip -c $archi > /home/utele/comprimidos/abc.gz
    else
        mkdir /home/utele/comprimidos
        gzip -c $archi > /home/utele/comprimidos/abc.gz
    fi
else
    echo El ARCHIVO $archi NO EXISTE.
fi
```

## 2. ADMINISTRACIÓN DE USUARIOS LOCALES

En este apartado veremos como crear usuarios y grupos tanto en modo gráfica como mediante comandos y conoceremos los detalles de los ficheros `/etc/passwd` y `/etc/shadow`.

### 2.1. EL FICHERO `/etc/passwd`

El fichero `/etc/passwd` almacena las cuentas de usuario del sistema.

Si ejecutamos el comando  
`cat /etc/passwd`  
podemos ver su contenido:

mar	:	x	:	1002	:	1002	:	Maria M. Soler ,,,	:	/home/mar	:	/bin/bash
											Shell	
											Carpeta personal	Ruta de la carpeta personal.
											Información del usuario	Nombre, ubicación, teléfono del trabajo, de la oficina.
											ID de grupo (GID)	ID del grupo principal del usuario. La información de los grupos está en /etc/groups.
											ID de usuario (UID)	El 0 está reservado para root y 1-99 para cuentas predefinidas. 100-999 para cuentas administrativas del sistema.
											Contraseña	Una x indica que la contraseña se encuentra encriptada en /etc/shadow. Debe tener entre 6 y 8 caracteres como mínimo.
											Nombre de usuario	Nombre que identifica al usuario en el sistema. Debe tener entre 1 y 32 caracteres.

Mediante el comando `grep` podemos filtrar usuarios: `cat /etc/passwd | grep mar`

### 2.2. FICHERO `/etc/shadow`

Para mayor seguridad, en el fichero `/etc/passwd` no aparecen las contraseñas de los usuarios del sistema. Estas se almacenan cifradas en el fichero `/etc/shadow`, propiedad del usuario root, porque ningún usuario pueda ver su contenido.

Si ejecutamos el comando `cat /etc/shadow` podemos ver su contenido:

mar	:	\$1\$NLJJ6\$ow5g1l1NgYITqqQQy5D21:14234:0:99999:7:::	
			Caducidad
			Días a los que se deshabilita la cuenta contados desde el 1 de enero de 1970.
			Inactivo
			Días a los que se deshabilita la cuenta después de que caduque la contraseña.
			Aviso
			Días a los que el usuario será avisado de que debe cambiar la contraseña antes de que ésta caduque.
			Máximo
			Días durante los que la contraseña es válida.
			Al terminar el usuario tiene que cambiar la contraseña.
			Mínimo
			Días que deben pasar como mínimo para que el usuario pueda cambiar la contraseña.
			Último cambio
			Días que han pasado desde la última vez que la contraseña fue cambiada contados desde el 1 de enero de 1970.
			Contraseña
			Contraseña encriptada. La forman entre 13 y 24 caracteres (a-z, A-Z, 0-9, \, /).
			Si comienza por el carácter ! indica que la cuenta está bloqueada
			Nombre de usuario
			Nombre que identifica al usuario en el sistema.
			Debe tener entre 1 y 32 caracteres.

Nuevamente, mediante el comando `grep` podemos filtrar usuarios: `cat /etc/shadow | grep mar`

Ojo! Si encontramos en el espacio de la contraseña como primer carácter el signo de admiración “!” significará que esa cuenta está deshabilitada.

```
GNU nano 2.5.3 Archivo: /etc/shadow
root:!$17045:0:99999:7:::
usuario:$6$MJBeuahn$THf0up8wwwprZ8ik6Ddrvmt3WQXiDbQpmve5hgmCmSw9V7tt47RKgagI0xo$
```

### 2.3. DIRECTORIO `/etc/skel`

En este directorio encontramos los ficheros de perfiles: `.bash_logout`, `.bashrc` y `.profile`. Cuando se crea un nuevo usuario se copian en su directorio home estos tres ficheros. Concretamente, el fichero `.bash_logout` se ejecuta al finalizar la sesión, el fichero `.bashrc` se ejecuta cuando se invoca un nuevo shell y `.profile` se ejecuta cuando el usuario inicia sesión en el sistema.

## 2.4. CREACIÓN DE USUARIOS

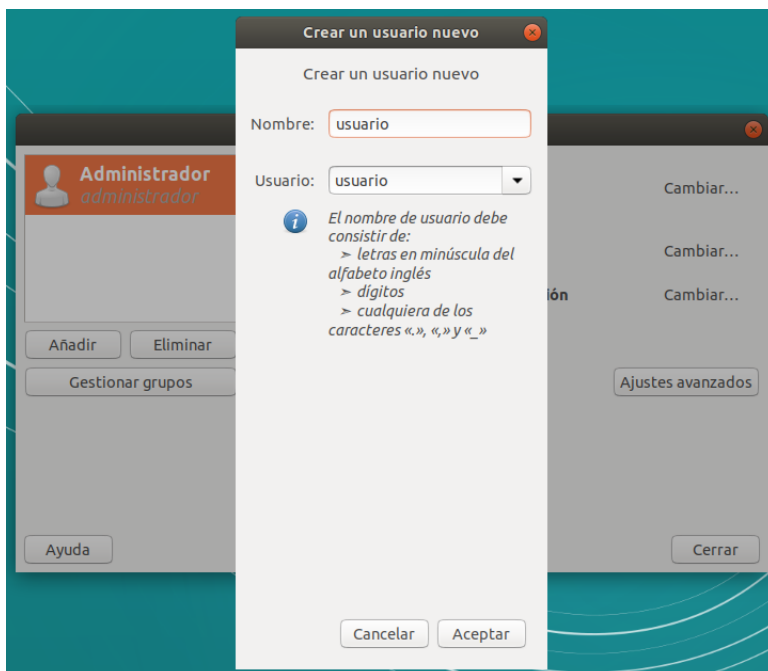
### 2.4.1. En modo gráfica

Buscamos la aplicación “Cuentas de usuario”:

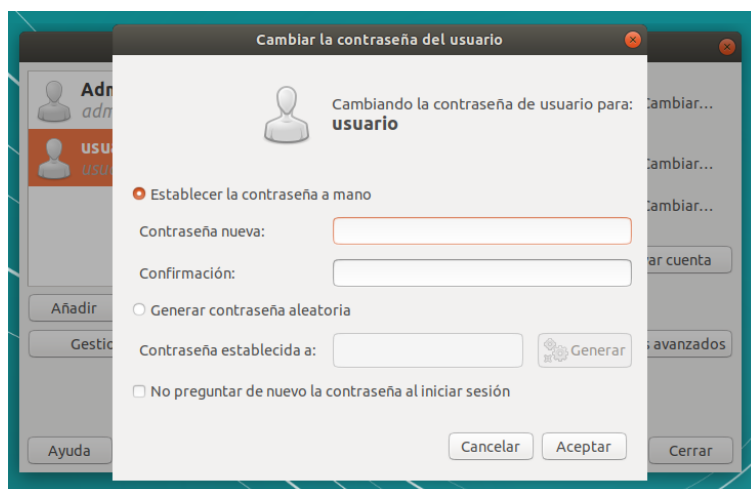


Hagamos clic a “Desbloquear” e introducimos la contraseña.

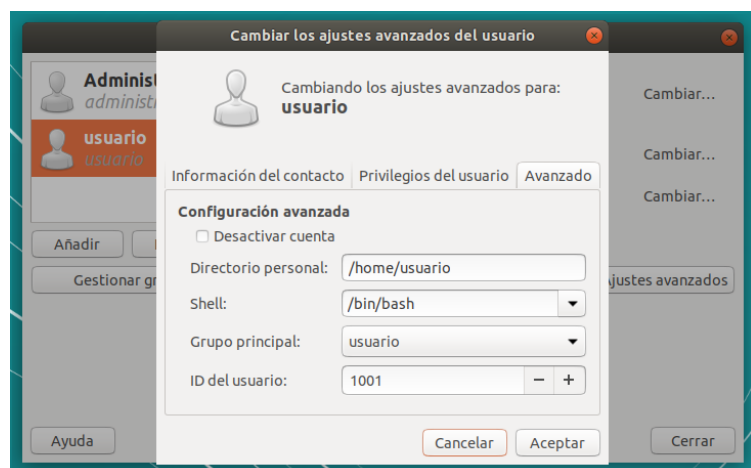
A continuación, hacemos clic en “añadir” y después escribimos el nombre completo de la persona y el nombre de usuario (login).



Finalmente, nos pedirá la contraseña y que indiquemos si tiene que escribirla para iniciar sesión o no.



Si a posteriori seleccionamos el usuario creado y pulsamos en “Ajustes Avanzados” podremos cambiar los permisos del usuario, grupos, etc.



### 2.4.2. En modo comando

Podemos usar los comandos `useradd` o `adduser`.

- **`sudo useradd [-opciones] login`**

Opciones:

- c: indicamos el nombre completo del usuario (entre comillas)
- d: indicamos el directorio home del usuario
- m: indicamos que se cree el directorio home que hemos escrito en -d
- s: indicamos el Shell
- g: indicamos el grupo del usuario

Ejemplo: `sudo useradd -c "Davit Martínez" -d /home/mar -m -s /bin/bash davit`

Ojo! Si no le indicamos el directorio home con `-d` y añadimos la opción `-m`, tendremos que crear el directorio home "a mano".

- **`sudo adduser login`**

Ejemplo: `sudo adduser davit`

El Shell te irá preguntando la contraseña del usuario y el nombre completo del usuario entre otros.

- **`sudo newusers nombredelfichero`**

También podemos usar el comando `newusers` para la creación de más de un usuario al mismo tiempo. Este comando lee un fichero y usa esa información para actualizar un grupo existente de usuarios o crearlos de nuevo.

Los campos que el fichero tendría que contener son los siguientes:

- Username:** login del usuario
- Password:** password del usuario
- UID:** Identificador del usuario
- GID:** Identificador del grupo primario del usuario
- User Info:** Información de usuario como el nombre, contacto,...
- Home Dir:** Directorio home del usuario
- Default Shell:** Shell por defecto del usuario

```
<Username>:<Password>:<UID>:<GID>:<User Info>:<Home Dir>:<Default Shell>
```

A continuación se puede ver un ejemplo de fichero de usuarios:

```
# cat /root/users.txt
tester1:test1@123:600:1530:Test User1,testuser1@abc.com:/home/tester1:/bin/bash
tester2:test2@123:601:1529:::/bin/bash
tester3:test3@123:::::
tester4:test4@123:::::/home/tester4:/bin/tsh
```

El comando se ejecutaría de la siguiente forma:

```
# newusers /root/users.txt
```

Podemos comprobar que los usuarios han sido dados de alta en el fichero `/etc/passwd`

```
# cat /etc/passwd | grep tester
tester1:x:600:1530:Test User1,testuser1@abc.com:/home/tester1:/bin/bash
tester2:x:601:1529:::/bin/bash
tester3:x:65537:65538:::
tester4:x:65538:65539::/home/tester4:/bin/tsh
```

También podemos comprobar que los grupos primarios se han creado en el fichero /etc/group

```
# cat /etc/group | grep tester
devel:x:1529:tester2
tester:x:1530:tester1
tester3:x:65538:tester3
tester4:x:65539:tester4
```

La x en el campo password indica que este campo está encriptado (shadowed) y almacenado en el fichero /etc/shadow

```
# cat /etc/shadow | grep tester
tester1:$1$NK0LH/kL$.gy3tBXHULsapiHP1PKs21:15607:0:99999:7:::
tester2:$1$NK0LH/kL$08Y4Vdi0Y4TTms.UCjjoE1:15607:0:99999:7:::
tester3:$1$NK0LH/kL$0kaRrrsm51tW3j5yheD7q1:15607:0:99999:7:::
tester4:$1$NK0LH/kL$pU2FIZEdGfYcYlMSCN8sI1:15607:0:99999:7:::
```

## 2.5. ELIMINACIÓN DE USUARIOS

### 2.5.1. En modo gráfico

Para lo cual marcaremos el usuario deseado y pulsamos el botón “eliminar”.

Nos preguntará si queremos eliminar también los archivos del usuario que están en /home/usuario.



### 2.5.2. En modo comando

Podemos usar los comandos `userdel` o `deluser`:

- **`sudo userdel [-opciones] login`**

Opciones:

- r: elimina el directorio home del usuario
- f: elimina la cuenta del usuario aunque esté conectado en el sistema

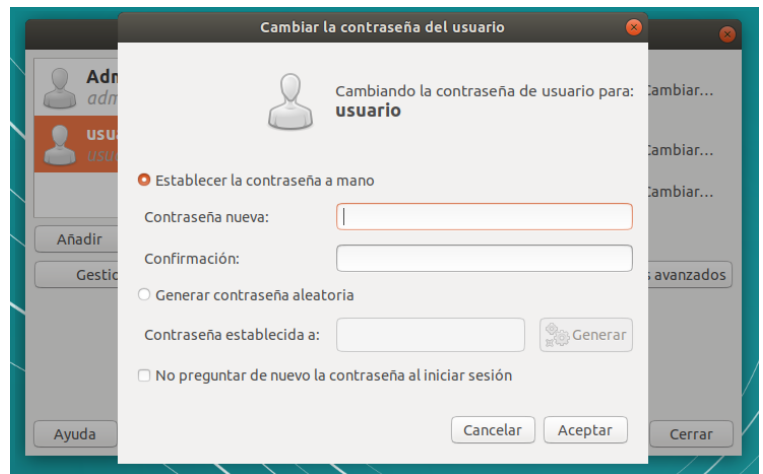
- **`sudo deluser login`**



## 2.6. CAMBIAR LA CONTRASEÑA DE UN USUARIO

### 2.6.1. En modo gráfica

Accedemos a cuentas de usuario, hacemos clic a cambiar contraseña y finalmente tan solo nos queda escribir la nueva contraseña.



### 2.6.2. En modo comando

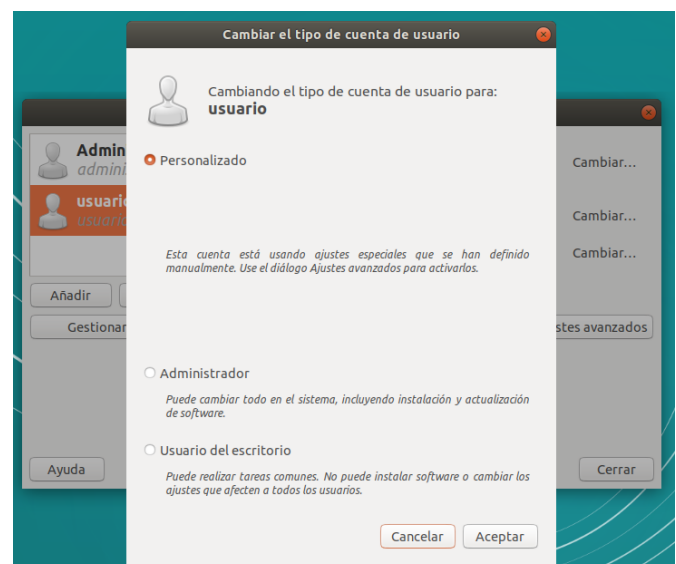
Mediante el comando **sudo passwd login** modificamos la contraseña de un usuario.

Ejemplo: `sudo passwd davit`

## 2.7. MODIFICAR CUENTAS DE USUARIO

### 2.7.1. En modo gráfica

Desde “Cuentas de usuario” podemos cambiar el nombre de la cuenta de usuario, su imagen o el tipo de cuenta, entre otras opciones.



### 2.7.2. En modo comando

Para modificar las características de los usuarios se emplea el comando **usermod [-opciones] login**

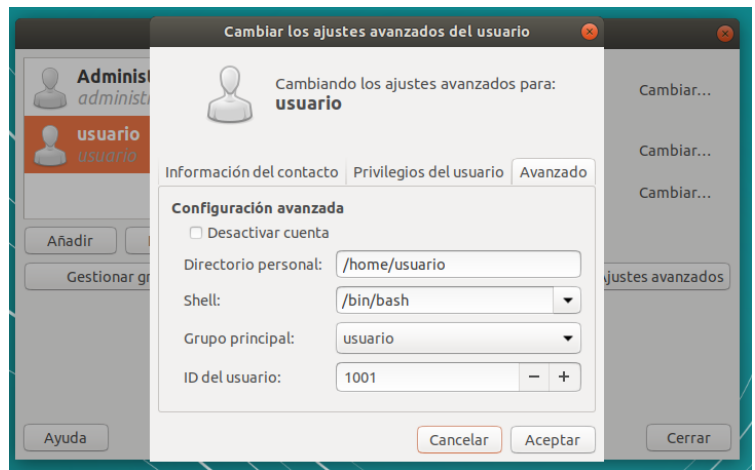
Ejemplo:

Para cambiar el nombre del usuario juan a jorge : `sudo usermod -l jorge juan`

## 2.8. BLOQUEO DE CUENTAS DE USUARIO

### 2.8.1. En modo gráfico

Accedemos a cuentas de usuario hagamos clic “Ajustes avanzados”. A continuación, elegimos la opción “Desactivar esta cuenta”:



### 2.8.2. En modo comando

Podemos bloquear el acceso de un usuario al sistema mediante el comando **usermod -L login**

Podemos comprobar que la cuenta ha sido bloqueada porque en el fichero `/etc/shadow` en el lugar donde tendría que estar la contraseña cifrada vemos un signo de admiración al principio.

Ejemplo:

```
sudo usermod -L frodobolson
sudo cat /etc/shadow | grep frodobolson
```

```
usuario@linuxserver:~$ sudo usermod -L frodobolson
usuario@linuxserver:~$ sudo cat /etc/shadow | grep frodobolson
frodobolson:!!$6$q0Da05CHxIQr6o$dANlIx189z.P02F2teFDN0ItvXWbEKp.wpHusDGrlo.gEQLKNq
hqpviQkiETyCrMZHbW43cjeejLSCUm.CA5a..:17132:0:99999:7:::
```

Podemos volver a habilitar la cuenta mediante **sudo usermod -U login**

Ejemplo:

```
sudo usermod -U frodobolson
sudo cat /etc/shadow | grep frodobolson
```

```
usuario@linuxserver:~$ sudo usermod -U frodobolson
usuario@linuxserver:~$ sudo cat /etc/shadow | grep frodobolson
frodobolson:$6$q0Da05CHxIQr6o$dANlIx189z.P02F2teFDN0ItvXWbEKp.wpHusDGrlo.gEQLKNq
hqpviQkiETyCrMZHbW43cjeejLSCUm.CA5a..:17132:0:99999:7:::
usuario@linuxserver:~$
```

### 3. ADMINISTRACIÓN DE GRUPOS LOCALES

Los grupos que hay en el sistema se almacenan en el archivo de texto `/etc/group`, en el cual podemos ver los diferentes grupos del sistema, así como los usuarios que pertenecen a estos grupos.

En este apartado veremos como crear usuarios y grupos tanto en modo gráfico como mediante comandos y conoceremos los detalles del fichero `/etc/group`.

Cuando creamos un usuario en Linux, se crea también un grupo con el mismo nombre que el usuario. Además, el usuario pertenece al grupo creado con el mismo nombre. Por otro lado, un usuario puede formar parte de varios grupos.

#### 3.1. FICHERO `/etc/group`

El fichero `/etc/group` almacena los grupos del sistema.

Si ejecutamos el comando

```
cat /etc/group
```

podemos ver su contenido:

- Cada línea del fichero es un grupo
- Los servicios del sistema también disponen de su propio grupo
- La contraseña del grupo también se oculta igual que en `/etc/passwd`
- La pertenencia a un grupo de Linux se usa para permitir el acceso de los usuarios a los dispositivos del equipo, por ejemplo: los usuarios miembros de los grupos `cdrom`, `floppy` o `plugdev` podrán usar la unidad CD-ROM, la disquetera y los dispositivos USB extraíbles (pendrives, cámaras digitales, tarjetas SD, discos duros externos) respectivamente.

```
mar@mar-VirtualBox:~$ sudo cat /etc/group | grep smx  
smx:x:1005:mar,eva,david
```

**smx : x : 1005 : mar, eva, david**

Usuarios miembros del grupo

ID de grupo (GID)

Contraseña del grupo

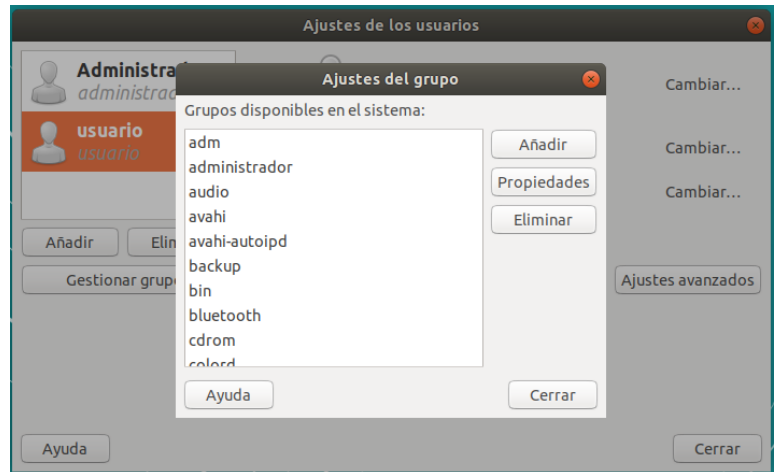
Nombre del grupo

```
mar@mar-VirtualBox:~$ sudo cat /etc/group | grep mar  
adm:x:4:mar  
cdrom:x:24:mar  
sudo:x:27:mar,eva  
dip:x:30:mar  
plugdev:x:46:mar  
lpadmin:x:109:mar  
mar:x:1000:  
sambashare:x:124:mar
```

## 3.2. CREACIÓN, ELIMINACIÓN Y MODIFICACIÓN DE GRUPOS

### 3.2.1. En modo gráfico

Con la aplicación "cuentas de usuario", también es posible crear, modificar o eliminar grupos. Para lo cual tenemos que hacer clic en el botón "Gestionar grupos":



### 3.2.2. En modo comando

- Para añadir un usuario al sistema estableciendo users como su grupo inicial o primario: `sudo adduser --ingroup users usuario`
- Para añadir nuevos grupos: `sudo addgroup grupo`
- Para añadir un usuario (existente o no) a un grupo existente: `sudo adduser usuario grupo`
- Para eliminar un grupo: `sudo groupdel grupo`
- Para cambiar el grupo inicial (primario) del usuario juan porque sea profesores: `sudo usermod -g profesores juan`
- Para cambiar el nombre del grupo profesores a alumnos: `sudo groupmod -n alumnos profesores`

## 3.3. AÑADIR UN USUARIO A UN GRUPO

Para añadir usuarios a un grupo (secundario) usamos el comando **usermod [-opciones] login**

Opciones:

- G: indicamos los grupos a los cuales añadir al usuario (separados por comas y sin espacios en blanco)
- a: así indicamos que no se quite al usuario de los grupos a los cuales ya pertenecía

Ejemplo: `sudo usermod -aG cdrom,plugdev mar`

De este modo estamos añadiendo al usuario mar a los grupos cdrom y plugdev le damos acceso al uso de la unidad de CD-ROM y a los dispositivos USB extraíbles. Además, el usuario Eva continúa conservando los grupos en los cuales ya estaba antes de ejecutar el comando usermod.

Nota: Alternativamente, para sistemas pequeños suele ser mejor "desproteger" los dispositivos adecuados porque todos los usuarios puedan usarlos, evitando tener que recordar añadir usuarios a los grupos adecuados. Por ejemplo, para dar acceso de lectura al CD-ROM (suponiendo que esté en /dev/hdc) y de lectura/escritura a la disquetera a todos los usuarios, haríamos:

```
sudo chmod a+r /dev/hdc
```

```
sudo chmod a+rw /dev/fd0
```

## 4. SERVICIOS

Los **servicios** son procesos que se ejecutan en segundo plan a la espera de ser llamados por el usuario para ofrecerle cierta función.

**Ejemplos de servicios** que a menudo se cargan en el sistema son, entre otros:

- **acpid**: servicio para el control de ahorro de energía. Se usa porque apago el equipo sin ningún problema
- **anacron**: servicio de la aplicación **anacron** para programar tareas
- **atd**: servicio del comando **at**
- **cron**: ejecuta las tareas programadas con **cron**
- **cups**: servicio de impresora
- **dbus**: se encarga de la comunicación entre los procesos
- **dhcpcd**: servidor DHCP
- **httpd**: servidor de páginas web Apache
- **named**: servidor DNS
- **netfs**: monta sistemas de archivos en red
- **networking**: servicio de las conexiones de red
- **network-manager**: herramienta de administración de conexiones de red
- **nfs**: servidor de ficheros en red
- **smb**: comparte archivos e impresoras con Windows
- **sshd**: habilita servicios seguros de red (secure shell)
- **udev**: servicio de control de los dispositivos. Controla los archivos del directorio **/dev**

### 4.1. EN MODO GRÁFICO

Por defecto no es posible administrar los servicios de manera gráfica en Ubuntu 18 (en la versión anterior era posible usar la aplicación de terceros **Boot-up Manager**). Únicamente podemos controlar las aplicaciones que se ejecutan al inicio. Para lo cual podemos hacer uso de la aplicación “Aplicaciones al inicio”



### 4.2. EN MODO COMANDO

Los **scripts** que se encargan de **arrancar y parar los servicios o reiniciarlos** se encuentran en el directorio **/etc/init.d**.

Los parámetros que se le suelen pasar dependen del script, pero normalmente son:

- **start**: arrancar servicio
- **restart**: reiniciar servicio (es un stop+start)
- **stop**: parar servicio
- **status**: ver el estado del servicio (saber si está ejecutándose)

De este modo, disponemos de dos formas para interactuar con servicios en Linux:

- Usando el comando **service**: `service nombro_servicio status/start/stop/restart`
- Usando lo **script init**: `/etc/init.d/nombro_servicio status/start/stop/restart`

*Nota: Desde la versión 9.10 de Ubuntu, empezó a implantarse Upstart como sustituto del servicio **init**, que es quien se encarga de iniciar los servicios durante el inicio del sistema, detenerlos cuando apagamos y controlar su funcionamiento mientras están activos. Por eso, es posible que no funcione para algún servicio usar el script **ini**. En estos casos, o bien usamos **service**, o **sudo start/stop/restart nombro\_servicio** (que es como lo hace Upstart).*

Para ver el estado de todos los servicios usamos: `service -status-all`

**Ejemplos:**

- Saber si está ejecutándose **cron**: `sudo service cron status`
- Arranque del demonio **cron**: `sudo service cron start`
- Parada del demonio **cron**: `sudo service cron stop`

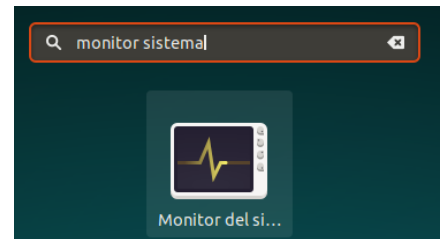
## 5. RENDIMIENTO Y MONITORIZACIÓN DE SISTEMA

### 5.1. EN MODO COMANDO

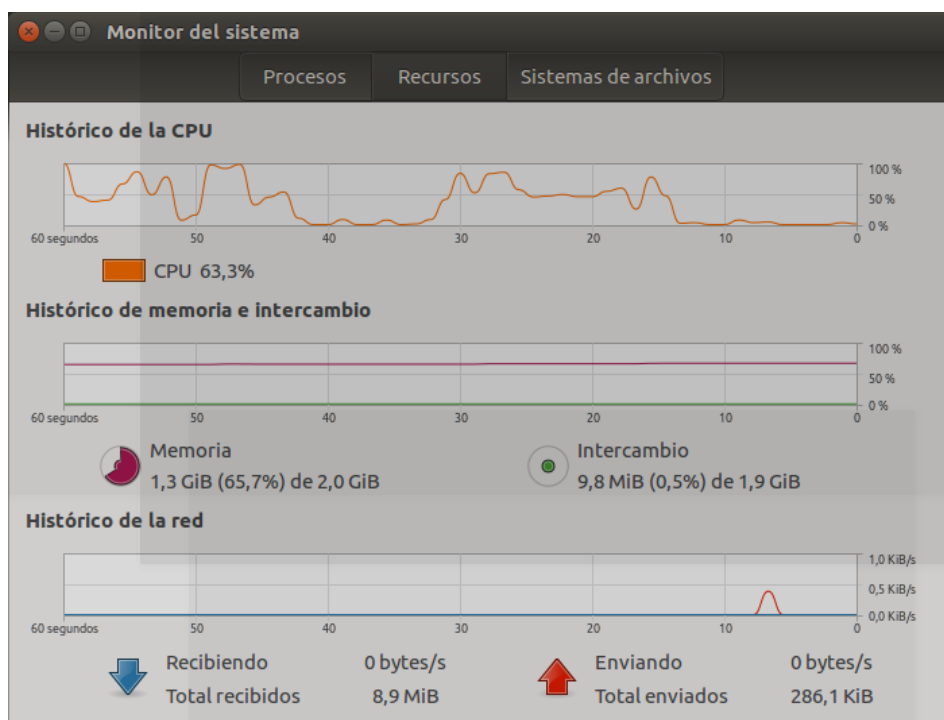
En la hora de monitorizar el sistema, podemos hacer uso de los comandos vistos en los apartados de procesos y servicios: ps, top, htop...

### 5.2. EN MODO GRÁFICO: MONITOR DEL SISTEMA

Podemos utilizar la herramienta anteriormente comentada de “monitor del sistema”:



Monitor del sistema						
Procesos Recursos Sistemas de archivos						
Nombre del proceso	Usuario	% CPU	ID	Memoria	Prioridad	
apport-gtk	usuario	0	10620	21,9 MiB	Normal	
at-spi2-registryd	usuario	0	3600	496,0 KiB	Normal	
at-spi-bus-launcher	usuario	0	3593	780,0 KiB	Normal	
bamfdaemon	usuario	0	3611	9,8 MiB	Normal	
bash	usuario	0	3001	4,0 KiB	Normal	
bash	usuario	0	10748	1,9 MiB	Normal	
blueman-applet	usuario	0	4760	21,7 MiB	Normal	



Monitor del sistema						
Procesos Recursos Sistemas de archivos						
Dispositivo	Carpeta	Tipo	Total	Disponible	Usado	
/dev/sda5	/	ext4	7,9 GB	2,3 GB	5,2 GB	69 %
/dev/sda6	/home	ext4	5,6 GB	5,3 GB	47,1 MB	0 %
/dev/sr0	/media/usuario	iso9660	58,2 MB	0 bytes	58,2 MB	100 %

### 5.3. ARCHIVOS O REGISTROS DE SUCESOS (LOGS)

Los sistemas Linux guardan en diferentes archivos cualquier funcionamiento anómalo o problema que pueda surgir en el sistema. Concretamente, rsyslogd es el encargado de vigilar y guardar estos registros, los cuales se encuentran en el directorio /var/log. Como ejemplos, podemos citar:

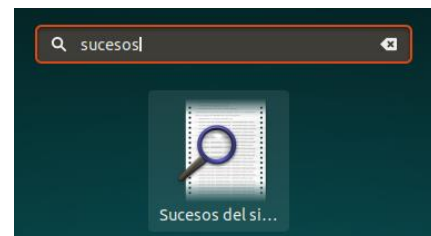
- auth: registro con mensajes relativos a la seguridad y a las autorizaciones
- cron: mensajes sobre demonios periódicos como cron, anacron, at...
- daemon: mensajes sobre otros demonios del sistema
- kern: mensajes relacionados con el núcleo
- lpr: mensajes relativos al subsistema de impresión
- syslog: mensajes relacionados con el demonio de registro
- user: mensajes relacionados con las aplicaciones de los usuarios

En modo texto, podemos consultar estos archivos mediante comandos como **tail** o **grep**. Por ejemplo:

- `tail -f /var/log/auth.log` nos permitirá obtener solo las 10 últimas líneas del archivo auth.log
- `cat /var/log/auth.log | grep "lightdm"` nos permite ver solo las líneas que contienen el texto lightdm.

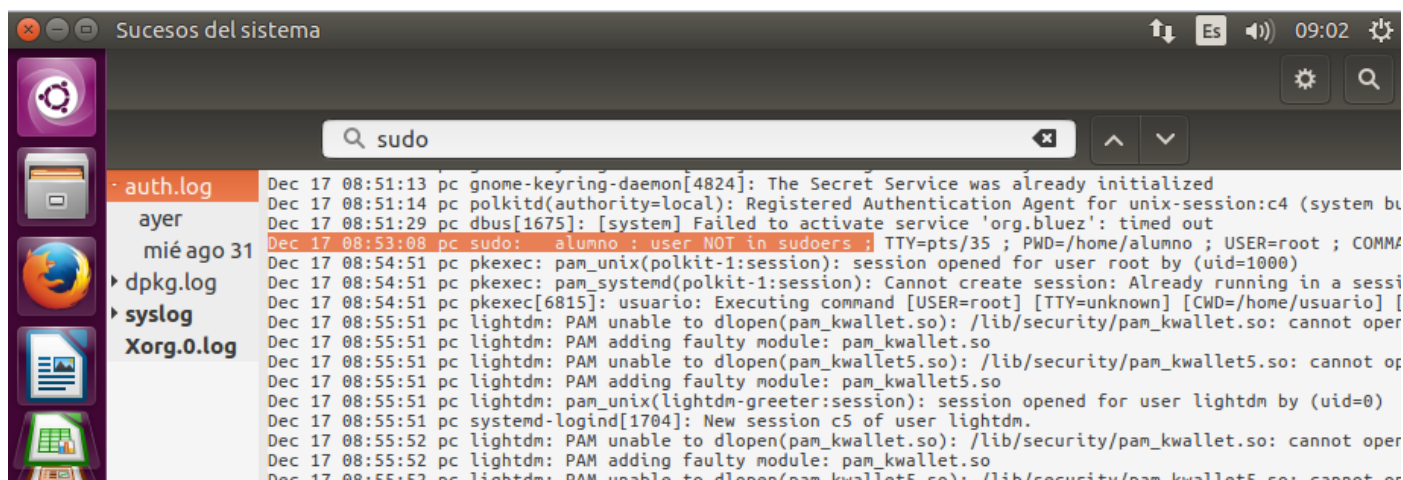
En modo gráfico podemos ver los archivos de sucesos del sistema mediante la utilidad con el mismo nombre "Sucesos del sistema".

El visor de archivos de sucesos nos muestra información sobre los servicios del sistema, la interacción entre los servicios y las aplicaciones y, en general, sobre el rendimiento del equipo.



La utilidad del visor de archivos de sucesos nos muestra información sobre el contenido de los ficheros log, de una forma cómoda y ordenada. Los archivos de sucesos se encuentran en la carpeta /var/log

Por ejemplo: Si abrimos una terminal y al intentar autenticarnos como root escribimos mal la contraseña, se creará un registro en el fichero de sucesos auth.log





## 6. GESTIÓN DISPOSITIVOS HARDWARE

### 6.1. VER DISPOSITIVO HARDWARE

En Ubuntu podemos ver los dispositivos que tiene el equipo de dos formas:

- En modo comando:

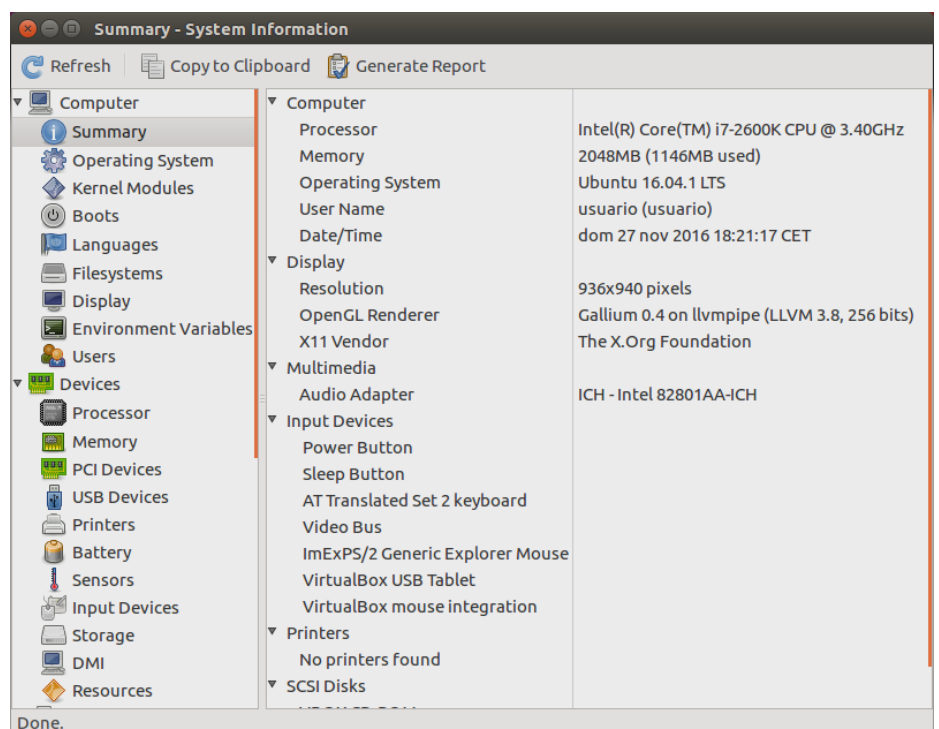
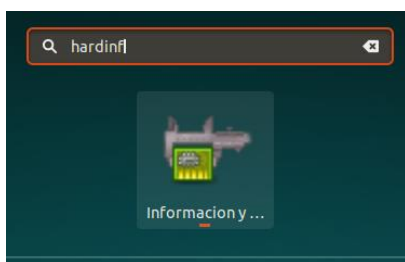
```
sudo lshw
```

```
administrador@ubuntu-server: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
administrador@ubuntu-server:~$ sudo lshw  
ubuntu-server  
descripción: Project-Id-Version: lshwReport-Msgid-Bugs-To: FULL NAME <EMAIL@ADDRESS>PO-Revision-Date: 2012-03-14 06:38+0000Last-Translator: Paco Molinero <paco@byas1.com>Language-Team: Spanish <es@l1.org>HIME-Version: 1.0Content-Type: text/plain; charset=UTF-8Content-Transfer-Encoding: 8bitX-Launchpad-Export-Date: 2018-07-12 13:19+0000X-Generator: Launchpad (build 18719)  
producto: VirtualBox  
fabricante: innotek GmbH  
versión: 1.2  
serie: 0  
anchura: 64 bits  
capacidades: smbios-2.5 dmi-2.5 vsyscall32  
configuración: family=Virtual Machine uuid=2070CCCE-D0E4-454A-88B4-E1A39BA46617  
*-core  
descripción: Placa base  
producto: VirtualBox  
fabricante: Oracle Corporation  
id físico: 0  
versión: 1.2  
serie: 0  
*-firmware  
descripción: BIOS  
fabricante: innotek GmbH  
id físico: 0  
versión: VirtualBox  
date: 12/01/2006  
tamaño: 128KiB  
capacidades: isa pci cdboot bootselect int9keyboard int10video acpi  
*-memory  
descripción: Memoria de sistema  
id físico: 1  
tamaño: 1990MiB  
*-cpu  
producto: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz  
fabricante: Intel Corp.  
id físico: 2  
información del bus: cpu@0
```

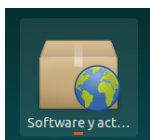
- En modo gráfica podemos instalar la aplicación hardinfo. Para lo cual, abrimos un terminal y escribimos:

```
sudo apt-get install hardinfo
```

Una vez instalada, podemos ejecutarla:



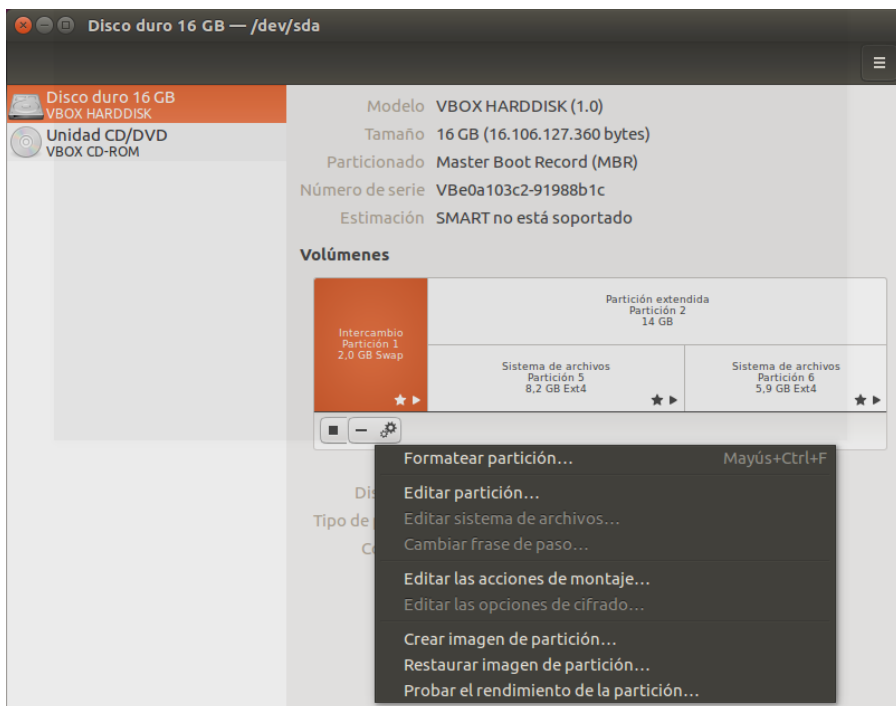
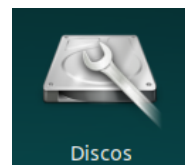
### 6.2. CONTROLADORES PRIVATIVOS



Si es lo que queremos es añadir controladores de dispositivos HW ya instalados o actualizar los controladores existentes, tendremos que instalar en nuestro equipo el que en Ubuntu se llaman **controladores privativos**. Estos controladores HW son suministrados por los fabricantes de HW para Ubuntu. El resto de controladores son genéricos suministrados por el propio SONIDO. Solo podremos instalar nuevos controladores HW si el fabricante los suministra.

### 6.3. GESTIÓN DE DISCOS

Para gestionar los discos en modo gráfica en Ubuntu podemos utilizar la aplicación “discos”:



Dado que las posibilidades de la herramienta “discos” son muy limitadas, podemos hacer uso de una aplicación mucho más completa: “**Gparted**”, la cual nos permitirá realizar todo tipos de operaciones con las particiones.

Para instalarla, podemos ir al “software de Ubuntu” o ejecutar el comando `sudo apt-get install gparted`:



Una vez instalada, lo abrimos y podremos **crear, eliminar o cambiar el tamaño de las particiones de los discos**:



## 7. PROGRAMACIÓN DE TAREAS

---

Algunas veces surge la necesidad de ejecutar un programa de manera regular a intervalos periódicos, o que puedan ejecutarse en determinados momentos sin tener que estar ante el ordenador (por ejemplo, para realizar copias de seguridad a altas horas de la noche, cuando nadie esté usándolo). Para llevar a cabo estos dos tipos de tareas, GNU/Linux dispone de los comandos `at` y `cron`, asociados a dos demonios `atd` y `crond` respectivamente, que tienen que estar en funcionamiento porque esto sea posible.

### 7.1. COMANDO AT

En algunas ocasiones se necesita **ejecutar una tarea en un momento particular y no con una frecuencia**. Para este caso usamos el comando **at**. Por lo tanto, a diferencia de `cron`, las tareas encomendadas mediante `at` solo se ejecutarán una vez.

En Ubuntu por defecto no está instalado `at`. Podemos instalarlo con el comando `sudo apt-get install at`

#### Uso del comando y parámetros:

- Apagar el sistema el día de hoy a las 4 pm  
`at -f /sbin/shutdown 11:55 pm today`
- Ejecutar el script `hola.sh` la semana que viene a las 2.00  
`at -f hola.sh 2.00 next week`
- Listar las tareas: se usa `atq` o el parámetro `-l`  
`at -l`  
  

```
3      Thu Jan 8 02.00:00 2015 a usuario
1      Thu Jan 1 23.55:00 2015 a usuario
```
- Borrar una tarea: se usa `atrm` o el parámetro `-d` y lo vayáis de la tarea. Si queremos borrar la tarea con vayáis  
`3`  
`atrm 1`
- Ver los detalles de la tarea programada: se usa `at -c` seguido de lo vayáis del job. Esto nos mostrará en las últimas líneas los comandos que se ejecutarán  
`at -c 3`

#### Ficheros que permiten regular la utilización del comando at:

- `/etc/at.allow` De existir este fichero, solo los usuarios contenidos en él podrán ejecutar `at`.
- `/etc/at.deny` De existir este fichero, los usuarios listados en él no podrán ejecutar `at`, `atrm`, y `atq`.

Otra forma de uso de `at` es: `at [hora] [fecha]` El comando `at` entonces se queda esperando que el usuario introduzca la serie de comandos que se quiere ejecutar ese día y hora. Para acabar de introducir comandos, hay que pulsar la combinación de teclas `<CTRL+D>`.

## 7.2. COMANDO CRON

Se utiliza para automatizar tareas con una periodicidad concreta, por ejemplo, revisar el espacio ocupado de los discos duros, borrar ficheros temporales, apagar el sistema de manera automática... El demonio `crond` se despierta cada minuto y comprueba los `crontabs` para determinar el que hay que hacer. Los usuarios (con suficientes privilegios) gestionan `crontabs` utilizando el comando `crontab`. El demonio `crond` se inicia normalmente por el proceso `init` en el arranque del sistema.

Generalmente en Ubuntu siempre está instalado en los sistemas (instalado y arrancado). De todos modos, si no se tiene instalado en el sistema y se desea instalar ejecutaremos el comando: `sudo apt-get install cron`

### Arranque y parada del demonio:

- Arranque del demonio cron: `/etc/init.d/cron start`
- Parada del demonio cron: `/etc/init.d/cron stop`
- Saber si está ejecutándose cron: `service cron status`

### Crear tareas con cron intermediando el fichero de configuración `/etc/crontab`:

Para crear una tarea usando cron es necesario **editar el archivo `/etc/crontab`**. Este fichero está dividido en líneas y cada línea representa una tarea programada. Estas líneas tienen el siguiente formato:

**minuto hora día mas día\_semana usuario orden\_a\_ejecutar**

Estos **campos** pueden tener los siguientes valores:

1. Minutos (0-59)
2. Horas (0-23)
3. Día del mes (1-31)
4. Mes del año (1-12)
5. Día de la semana (0-7) (domingo=0=7, lunes=1, sábado=6)
6. Comando, programa o script a ejecutar

```
----- minutos (0 - 59)
| ----- horas (0 - 23)
| | ----- día del mes (1 - 31)
| | | ----- mes (1 - 12)
| | | | ----- día de la semana (0 - 7) (domingo=0=7, lunes=1, sábado=6)
| | | | |
comando a ejecutar
```

### Símbolos especiales para los cinco primeros campos:

- \* : indica cualquier valor
- , : actúa como separador de una lista de valores
- # : indica que el que acompaña es un comentario (no se ejecutará)
- : sirve para indicar un rango de valores
- / : sirve para indicar un paso de valor (por ejemplo, en el campo hora si se indica `/4` se está detallando que la tarea se realizará cada cuatro horas).

### Cadenas comodín:

En vez de la configuración anterior se pueden utilizar las siguientes cadenas comodín:

- |  |   |
|--|---|
| @reboot: Se ejecuta al iniciarse la máquina. | @weekly: Se ejecuta una vez por semana. |
| @yearly: Se ejecuta una vez en el año.       | @daily: Se ejecuta una vez en el día.   |
| @monthly: Se ejecuta una vez en el mes.      | @hourly: Se ejecuta una vez por hora.   |

## Ejemplos:

- Ejecutar el script `/home/usuario/hola.sh` todos los días a 12.01 y a las 23.01  
`1 12,23 * * * /home/usuario/hola.sh`
- Ejecutar el script `/home/usuario/hola.sh` a las 9 y las 18 horas todos los días laborables:  
`0 9,18 * * 1-5 /home/usuario/hola.sh`
- Ejecutar el script `/home/usuario/hola.sh` el día 18 del mes y todos los martes (el martes 18 se ejecutará dos veces):  
`* * 18 * 2 /home/usuario/hola.sh`
- Ejecutar `/home/usuario/arranque.sh` a las 4:01am cada día de cada mes  
`01 04 * * * /home/usuario/arranque.sh`
- Ejecutar `/home/usuario/arranque.sh` cuando arrancada el sistema  
`@reboot /home/usuario/arranque.sh`
- Ejecutar `/home/usuario/arranque.sh` cada 10 minutos  
`*/10 * * * * /home/usuario/arranque.sh`  
`0,10,20,30,40,50 * * * * /home/usuario/arranque.sh`
- Ejecutar `/home/usuario/arranque.sh` a las 4 o 5 y un minuto o 31 minutos desde el 1 hasta el 15 de Enero hasta Junio  
`01,31 04,05 1-15 1-6 * /home/usuario/al_arrancar.sh`
- Ejecutar el comando `dos.sh` a las 4 y 45 de la mañana sola si `uno.sh` se ejecuta con éxito  
`45 04 * * * /home/usuario/uno.sh && /home/usuario/dos.sh`
- Borrar el `/tmp` todos los días laborables a las 4.30 am  
`30 4 * * 1-5 rm -rf /tmp/*`

## Orden crontab:

- Cron también se puede configurar mediante la orden **crontab**. Crontab es lo que hace es gestionar los ficheros crontabs asignados a cada usuario (en `/var/spool/cron/crontabs/`).
- crontab permite a cada usuario poder gestionar sus propias planificaciones de tareas.
- **Uso de la orden y parámetros:**  
`crontab [-l e r o] fichero`  
Significado de los **parámetros**:
  - l: muestra el fichero de configuración del usuario
  - e: edita el fichero de configuración del usuario
  - r: borra el fichero de configuración del usuario
  - o usuario: especifica el usuario propietario de la tarea (normalmente, esta opción lo usa el usuario root para cambiar propietarios de tareas).

- Si el usuario `morenoperezjc` ejecuta un fichero tipo `cron` se guardará un archivo `morenoperezjc` en el directorio `/var/spool/cron/crontabs/`. Para ejecutar un `crontab` puede ser administrador o pertenecer al grupo de usuarios `crontab`.
- **Ejemplo:** Borrar todos los archivos descargados en la máquina cada vez que se reinicie la misma.
  - 1.- Generar un fichero de nombre `ejemplo.cron` de una línea (recordar que se ejecutan las tareas por líneas mediante un shell) con el contenido:
 

```
@reboot rm /home/morenoperezjc/Descargas/*
```
  - 2.- Ejecutar la carga del fichero de planificación:
 

```
crontab ejemplo.cron
```

esto creará un fichero `morenoperezjc` en el directorio `/var/spool/cron/crontabs/`
  - 3.- Para estar seguro que se ha añadido a la lista de tareas, mirar la lista de éstas:
 

```
crontab -l
```

Si queremos desprogramar la tarea, es decir, eliminar las tareas programadas con `crontab` habrá que ejecutar el siguiente comando:

```
crontab -r
```

#### **Ficheros para poder permitir o denegar el servicio a determinados usuarios del sistema:**

- `/etc/cron.allow`: si existe este fichero, solo los usuarios contenidos en él tendrán permiso para ejecutar tareas programadas. Cada usuario tiene que estar en una línea diferente.
- `/etc/cron.deny`: en el supuesto de que queramos denegar el acceso a las tareas programadas, se tendrá que crear este archivo y registrar los usuarios que no podrán programar tareas (uno por línea).

En Ubuntu por defecto no existen estos ficheros y el comportamiento por defecto es permitir en todos los usuarios que ejecutan trabajos vía `crontab`. Sin embargo, si creas un fichero `cron.allow` o `cron.deny` en blanco el comportamiento del demonio cambia. En ese momento solo podrán utilizar el `cron` los usuarios `root` y aquellos que estén inscritos en el fichero `cron.allow`.

#### **Ejecución de cron:**

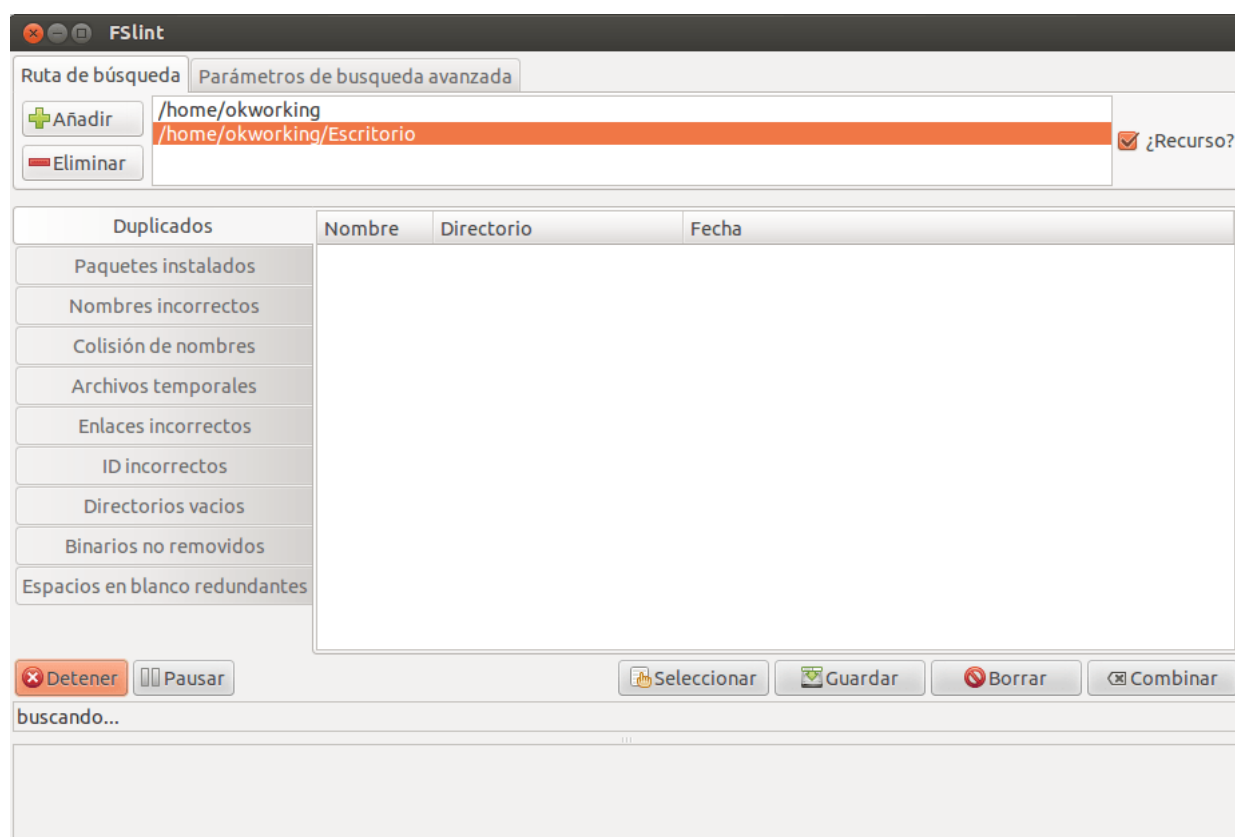
- El demonio `crond` se despierta cada minuto y comprueba los `crontabs` para determinar el que hay que hacer. Por lo tanto, busca ficheros en `/var/spool/cron` para ejecutarlos en la hora indicada. Además también ejecuta las acciones indicadas en los ficheros `/etc/crontab` y en el directorio `/etc/cron.d/`. Estos ficheros suelen ser de mantenimiento del sistema.
- Por otro lado, el administrador también puede crear scripts que se ejecutan con periodicidad horaria, diaria, semanal y mensual colocándolos en los directorios `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` o `/etc/cron.monthly`. La fecha y hora de ejecución de estos scripts se controla en el fichero `/etc/crontab`
- Problema: `cron` está pensado para sistemas funcionando 24/7 como es el caso de servidores. Sin embargo, para el caso de equipos domésticos, si el sistema está apagado en la hora de una acción `cron`, esa tarea no se realiza. Solución: `Anacron`. `Anacron` ejecuta asincrónicamente tareas periódicas programadas. Al iniciarse el sistema comprueba si hay tareas periódicas pendientes (que no se realizaron para estar el sistema apagado). `Anacron` se configura mediante el fichero `/etc/anacrontab`

## 8. APLICACIONES DE MANTENIMIENTO Y OPTIMIZACIÓN

En la actualidad existen varias aplicaciones dedicadas en el mantenimiento y optimización de sistemas GNU/Linux. Es el caso de las aplicaciones Fslint, Ubuntu Tweak o Bleachbit.

### 8.1. FSLINT

Fslint se utiliza para eliminar archivos duplicados en Ubuntu. Es un programa que se encuentra en los repositorios oficiales y podemos instalarlo desde el centro de software. La Interfaz es muy sencilla. Por defecto, la busca de duplicados se restringe a nuestro directorio de usuario, pero pulsando en el botón Añadir podemos agregar nuevos directorios. Pero todavía podemos ir más lejos, si en lugar de buscar archivos duplicados quieres buscar, por ejemplo, entre los Paquetes instalados, tendrías que hacer clic sobre esa entrada en la lista de la izquierda. Por ultimo basta con hacer clic en el botón Buscar para conocer la lista de duplicados en una lista.



### 8.2. UBUNTU TWEAK

Ubuntu-Tweak: es una aplicación que nos permite gestionar nuestro Ubuntu de manera gráfica, de una manera fácil e intuitiva. De este modo permite, entre otros, ver información del hardware, cambiar la apariencia, configurar, instalar aplicaciones, añadir repositorios, actualizar, modificar el arranque, configurar los efectos 3D del escritorio y gestionar el sistema.





### 8.3. BLEACHBIT

Esta aplicación permite:

- Liberarse de archivos innecesarios en nuestro disco duro y de este modo liberar espacio e incrementar en la medida de lo posible la eficiencia y rendimiento de nuestro sistema.
- Preservar nuestra privacidad puesto que con bleachbit podemos triturar archivos y el espacio teóricamente libre de nuestro disco. De este modo un tercero nunca podrá recuperar los datos que nosotros hemos borrado de nuestro disco duro.

## 9. COPIAS DE SEGURIDAD

Las copias de seguridad se tienen que realizar periódicamente y con planificación para evitar cualquier pérdida de información del sistema.

Además de decidir donde las almacenaremos y qué tipo de copias mujeres de seguridad realizaremos (total o integral, incremental o diferencial) es muy importante elegir qué información elegiremos para realizar una copia de ella.

Entre las carpetas que son convenientes salvar en una copia de seguridad están:

- /home: contiene las carpetas personales de los usuarios
- /root: contiene la carpeta personal del usuario root
- /etc: contiene los archivos de configuración
- /var/log: contiene los ficheros de incidencia del sistema para descubrir qué es el que ha provocado un fallo

### 9.1. MODO COMANDO

Una opción es realizar la copia de seguridad en formato comprimido porque ocupo menos espacio. Esto lo puedes hacer con los comandos bzip2, 7zip, gzip... además del comando para empaquetar tar.

Ejemplo: creación de copia comprimida en formato bzip2 de la carpeta /home/usuario:

```
tar -cvjf carpeta_usuario.tar.bz2 /home/usuario
```

(Nota: gráficamente, también se pueden comprimir los archivos mediante el “gestor de archivadores”).

## 9.2. MANERA GRÁFICA



También podemos realizar las copias de seguridad utilizando la herramienta “copias de seguridad”.



## 10. GESTOR DE ARRANQUE: GRUB

### 10.1. CONFIGURAR EL GESTOR DE ARRANQUE

GNU GRUB es el software que la mayoría de distribuciones GNU/Linux utilizan como gestor de arranque. Como gestor de arranque, es el primer programa que se carga del disco duro en el proceso de arranque, por eso se suele instalar en el sector de arranque del disco duro. En el directorio `/boot` se encuentran los archivos que el gestor de arranque necesita para arrancar el sistema, incluyendo lo kernel de Linux.

Este gestor de arranque muestra al usuario un menú con todos los SONIDO Linux y Windows que detecto en el equipo (cosa que no ocurre con el gestor de arranque de Windows que solo detecta a los SONIDO de Microsoft). Por lo tanto, GRUB nos permite tener instalados varios SONIDO y varias versiones de ellos y al arrancar el ordenador nos permite elegir como queremos arrancar. También nos permite decidir qué queremos tener como predeterminado.

La primera versión de GRUB usaba el fichero `/boot/grub/menu.lst` para configurar las opciones de arranque del sistema. En cambio, GRUB 2 utiliza el fichero `/boot/grub/grub.cfg`, el cual se genera a partir de:

- El fichero `/etc/default/grub` que modifica el menú que GRUB2 presenta por pantalla. Siempre que modificamos el fichero `/etc/default/grub` es necesario actualizar el fichero `grub.cfg` con el comando: `sudo update-grub`
- Los archivos del directorio `/etc/grub.d/` que determinan la orden de aparición de las entradas en el menú

```
usuario@linuxserver:~$ sudo ls /etc/grub.d/
[sudo] password for usuario:
00_header          20_linux_xen      30_uefi-firmware  README
05_debian_theme    20_memtest86+    40_custom
10_linux           30_os-prober     41_custom
usuario@linuxserver:~$
```

Aquellos con número menor se ejecutan antes.

- Los archivos `grub-install`, `grub-setup`, `grub-mkconfig...` del directorio `/usr/sbin/`

A continuación, se explica cómo cambiar las opciones del gestor de arranque de Ubuntu GRUB 2:

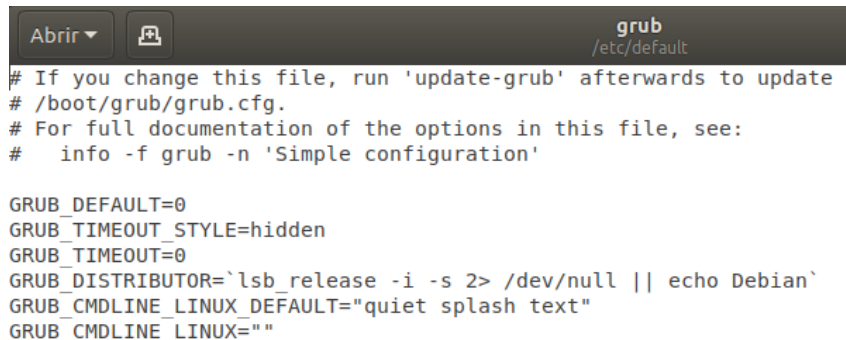
Abrimos un terminal y escribimos:

```
sudo gedit /etc/default/grub
```

Hay que buscar la línea **GRUB\_DEFAULT=0** y modificar su valor por la línea del menú que nos interese que sea la que arranque por defecto. Para lo cual hay que contar la posición que ocupa en el menú el SONIDO con el cual queremos arrancar por defecto, teniendo en cuenta que hay que empezar a contar de arriba abajo y desde **0**. De este modo, la primera opción del Grub será la **0**, la segunda la **1** y así sucesivamente.

Guardamos el fichero, cerramos el editor y escribimos en un terminal:

```
sudo update-grub (o sudo update-grub2)
```



```
grub
/etc/default

# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

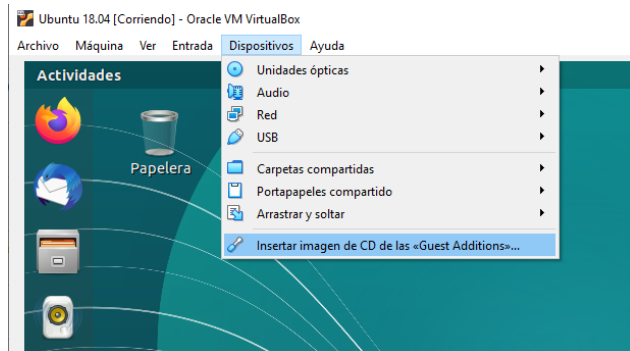
GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash text"
GRUB_CMDLINE_LINUX=""
```

*Nota: no tenemos que modificarlo directamente del fichero `/boot/grub/grub.cfg`, puesto que si el SONIDO se actualiza (se elimina o instala una nueva versión de kernel o se ejecuta el comando `update-grub`), se pierden los cambios hechos. Por eso, siempre tiene que modificarse lo GRUB a través del fichero `/etc/default/grub`*

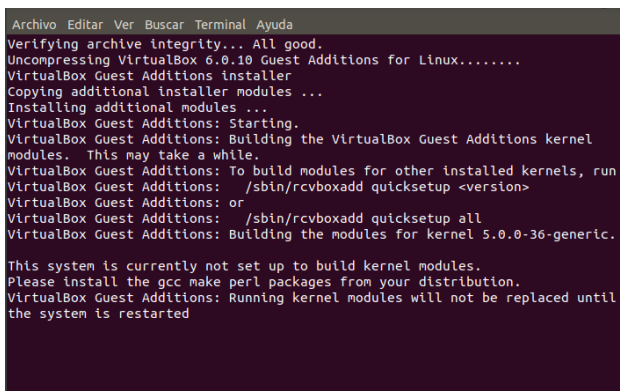
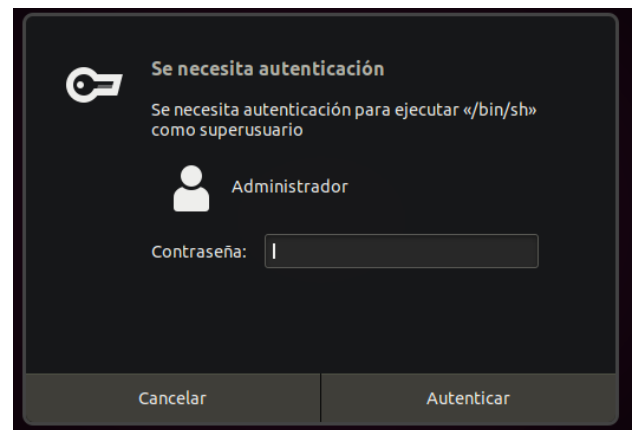
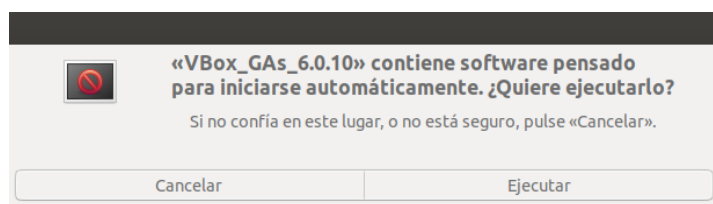
## 11. ANEXOS

### 11.1. INSTALAR GUEST ADDITIONS EN UBUNTU

Ya sabréis que las Guest Additions de VirtualBox es un complemento indispensable que es necesario instalar para poder usar la máquina virtual en pantalla completa, para trabajar visualmente de una manera más cómoda.



Lo instalaremos mediante la propia opción de VirtualBox. Para lo cual hay que ir a DispositivosInsertar→ imagen de CD de las Guest Additions y seguir los siguientes pasos:



Después de esto, solo tenemos que reiniciar la máquina virtual, y las guest additions quedaron instaladas en nuestro sistema Ubuntu virtualizado.