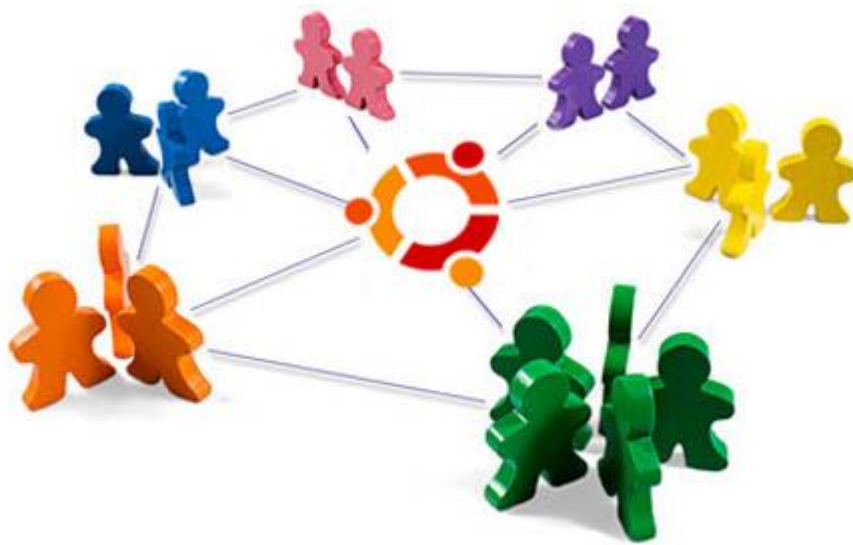


Sistemas Informáticos

UD 8. Administración de GNU/Linux Ubuntu 20.04



ÍNDICE

1.	Administración de usuarios locales.....	3
1.1.	El fichero /etc/passwd.....	3
1.2.	Fichero /etc/shadow	3
1.3.	Directorio /etc/skel	3
1.4.	Creación de usuarios	4
1.5.	Eliminación de usuarios.....	6
1.6.	Cambiar la contraseña de un usuario.....	7
1.7.	Modificar cuentas de usuario	7
1.8.	Bloqueo de cuentas de usuario	8
2.	Administración de grupos locales	9
2.1.	Fichero /etc/group	9
2.2.	Creación, eliminación y modificación de grupos.....	10
2.3.	Añadir un usuario a un grupo	10
3.	Gestión de procesos.....	11
3.1.	Modo comando: ps, top, htop.....	11
3.2.	Modo gráfico: monitor del sistema	12
4.	Servicios	13
4.1.	En modo gráfico	13
4.2.	En modo comando	13
5.	Rendimiento y monitorización de sistema	14
5.1.	En modo Comando.....	14
5.2.	En modo gráfico: Monitor del sistema	14
5.3.	Archivos o registros de sucesos (logs)	15
6.	Gestión dispositivos hardware.....	16
6.1.	Ver dispositivos hardware	16
6.2.	Controladores privativos	16
6.3.	Gestión de discos.....	17
7.	Programación de tareas.....	18
7.1.	Comando At.....	18
7.2.	Comando Cron.....	19
8.	Copias de seguridad	22
8.1.	Modo comando	22
8.2.	Modo gráfico	22
9.	Gestor de arranque: grub	23
9.1.	Configurar el gestor de arranque	23
10.	Runlevels o niveles de arranque	24
10.1.	Runlevels en la mayoría de distribuciones Linux.....	24
10.2.	Runlevels en Ubuntu	24
10.3.	Iniciar el sistema en modo texto o en modo gráfico	24
10.4.	Iniciar la interfaz gráfica desde el modo texto	25

1. ADMINISTRACIÓN DE USUARIOS LOCALES

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

1.1. 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:

<code>mar : x : 1002 : 1002 : Maria M. Soler ,,, : /home/mar : /bin/bash</code>	
	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 <code>/etc/groups</code> .
	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 <code>/etc/shadow</code> . 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`

1.2. FICHERO `/etc/shadow`

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

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

<code>mar :\$1\$NLJJ6\$ow5g1l1NgYITqqQQy5D21:14234:0:99999:7:::</code>	
	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$THf0up8wwwprZ8ik6Ddrvmt3WQXiDbQpmve5hgmCmSw9V7tt47RKgagI0xo$
```

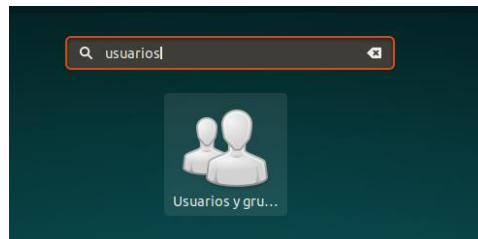
1.3. DIRECTORIO `/etc/skel`

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.

1.4. CREACIÓN DE USUARIOS

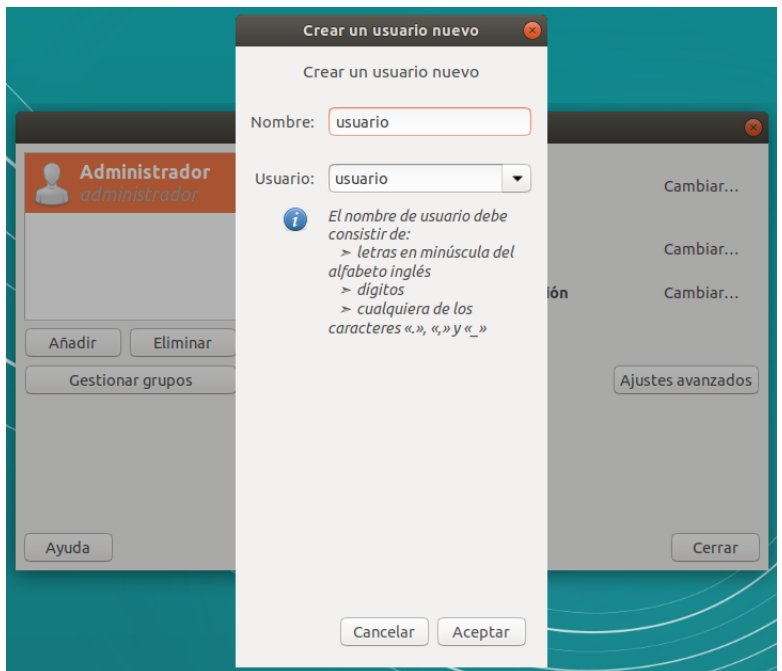
1.4.1. En modo gráfico

Buscamos la aplicación “Cuentas de usuario”:

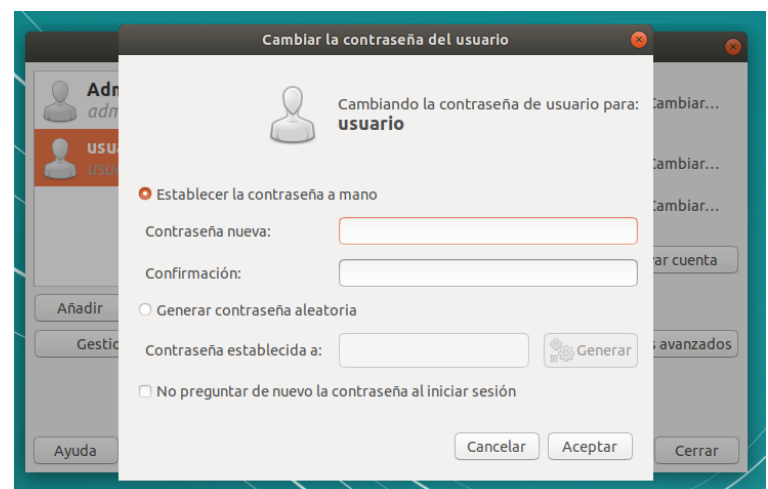


Hacemos clic en “Desbloquear” e introducimos la contraseña.

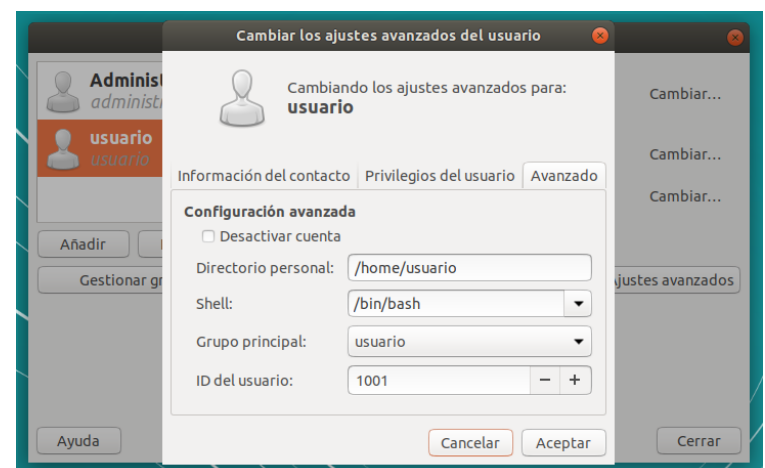
A continuación, hacemos clic en “añadir” y luego escribimos el nombre completo de la persona y el nombre de usuario (login).



Finalmente, nos pedirá la contraseña y que indiquemos si debe 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.



1.4.2. 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 "Mar Soler" -d /home/mar -m -s /bin/bash mar`

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 mar`

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 debería 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:::
```

1.5. ELIMINACIÓN DE USUARIOS

1.5.1. En modo gráfico

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

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



1.5.2. En modo comando

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

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

Opciones:

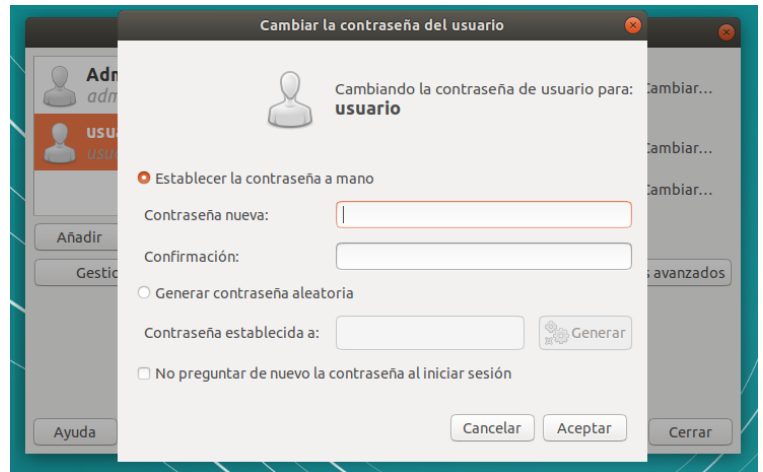
- r: elimina el directorio home del usuario
- f: elimina la cuenta del usuario aunque esté conectado en el sistema

- **sudo deluser login**

1.6. CAMBIAR LA CONTRASEÑA DE UN USUARIO

1.6.1. En modo gráfico

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



1.6.2. En modo comando

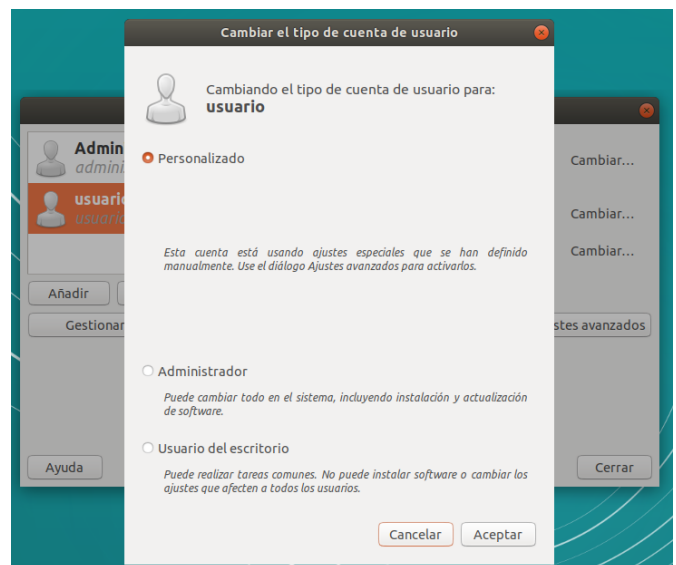
Mediante el comando **sudo passwd login** modificamos la contraseña de un usuario.

Ejemplo: `sudo passwd mar`

1.7. MODIFICAR CUENTAS DE USUARIO

1.7.1. En modo gráfico

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



1.7.2. En modo comando

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

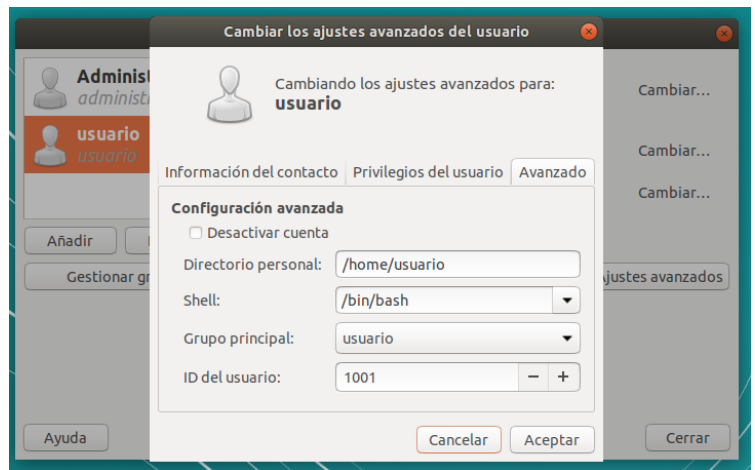
Ejemplo:

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

1.8. BLOQUEO DE CUENTAS DE USUARIO

1.8.1. En modo gráfico

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



1.8.2. 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 debería 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$dANlIx189z.P02F2teFDN0ItvXWbEKp.wpHusDGrlo.gEQLKNq
hqpviQkiETyCrMZHbW43cjeejLSCUm.CA5a..:17132:0:99999:7:::
usuario@linuxserver:~$
```


2. ADMINISTRACIÓN DE GRUPOS LOCALES

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

En este apartado veremos cómo 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.

2.1. 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

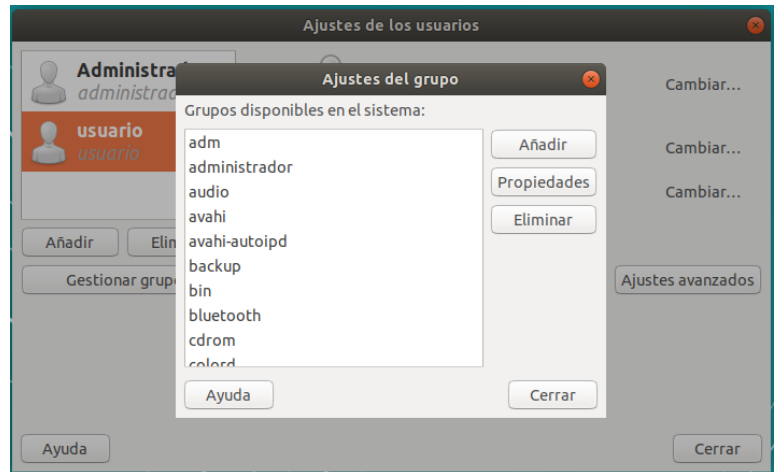
Nombre del grupo

```
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
```

2.2. CREACIÓN, ELIMINACIÓN Y MODIFICACIÓN DE GRUPOS

2.2.1. En modo gráfico

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



2.2.2. 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 para que sea profesores: `sudo usermod -g profesores juan`
- Para cambiar el nombre del grupo profesores a alumnos: `sudo groupmod -n alumnos profesores`

2.3. 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 que 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 que 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 sigue conservando los grupos en los que ya estaba antes de ejecutar el comando usermod.

Nota: Alternativamente, para sistemas pequeños suele ser mejor "desproteger" los dispositivos adecuados para que 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*
```

3. GESTIÓN DE PROCESOS

Un **proceso** es un programa en ejecución. Por un lado, los **usuarios** ponen en ejecución programas como las aplicaciones, es decir, se ponen en marcha procesos. Por otro lado, el propio **SO** ejecuta una gran cantidad de “servicios” llamados “demonios” (daemons) en GNU/Linux. Estos servicios se ejecutan en segundo plano o background sin necesidad de interactuar con los usuarios del sistema. Se identifican porque suelen tener una **d** al final de su nombre, por ejemplo, el servidor web httpd. Estos servicios o demonios se configuran para ser iniciados manualmente o bien cuando se arranca el sistema.

3.1. MODO COMANDO: PS, TOP, HTOP...

Ver procesos en ejecución:

El comando **ps** informa del estado de los procesos.

Para obtener información detallada sobre los procesos en ejecución usamos **ps aux**

```
usuario@linuxserver: ~  
usuario@linuxserver:~$ ps aux  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root         1  0.0  0.3 119932 6208 ?        Ss   17:03   0:01 /sbin/init splash  
root         2  0.0  0.0      0     0 ?        S    17:03   0:00 [kthreadd]  
root         3  0.0  0.0      0     0 ?        S    17:03   0:00 [ksoftirqd/0]
```

El campo **STAT** contiene información sobre el estado del proceso. Los posibles estados son:

- R: preparado para ejecución
- S: dormido
- D: letargo no interrumpible
- T: parado
- Z: zombi

Monitorizar procesos:

El comando **ps** muestra una instantánea de los procesos actuales, pero si deseamos monitorizar el sistema, es decir, observar la actividad de la CPU en tiempo real, empleamos el comando **top**. **top** nos ofrece una interfaz interactiva para manipular los procesos y clasificar las tareas por uso de CPU memoria y tiempo de ejecución.

Podremos finalizar la ejecución de cualquier proceso pulsando la tecla **k** y escribiendo el PID del proceso.

Pulsando la tecla **u** y escribiendo un nombre de usuario, se mostrarán únicamente los procesos iniciados por dicho usuario. Para volver a verlos todos, sólo tenemos que volver a pulsar la tecla **u** y pulsar Intro.

Para salir del comando **top**, sólo hay que pulsar la tecla **q**.

```
top - 18:56:36 up 3 days, 6:46, 4 users, load average: 0.06, 0.03, 0.05  
Tasks: 232 total, 1 running, 205 sleeping, 2 stopped, 24 zombie  
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st  
Mem: 507552k total, 486616k used, 20936k free, 18804k buffers  
Swap: 522236k total, 260056k used, 262180k free, 152472k cached  


| PID   | USER | PR | NI  | VIRT  | RES  | SHR  | S | %CPU | %MEM | TIME+   | COMMAND        |
|-------|------|----|-----|-------|------|------|---|------|------|---------|----------------|
| 2491  | mar  | 20 | 0   | 92612 | 6060 | 3020 | S | 0.3  | 1.2  | 0:03.90 | gnome-terminal |
| 11116 | root | 20 | 0   | 0     | 0    | 0    | S | 0.3  | 0.0  | 0:00.50 | kworker/0:0    |
| 1     | root | 20 | 0   | 3516  | 1276 | 740  | S | 0.0  | 0.3  | 0:00.82 | init           |
| 2     | root | 20 | 0   | 0     | 0    | 0    | S | 0.0  | 0.0  | 0:00.01 | kthreadd       |
| 3     | root | 20 | 0   | 0     | 0    | 0    | S | 0.0  | 0.0  | 0:04.20 | ksoftirqd/0    |
| 6     | root | RT | 0   | 0     | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | migration/0    |
| 7     | root | RT | 0   | 0     | 0    | 0    | S | 0.0  | 0.0  | 0:02.28 | watchdog/0     |
| 8     | root | 0  | -20 | 0     | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | cpuset         |
| 9     | root | 0  | -20 | 0     | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | khelper        |


```

htop es muy parecido a **top**, pero más flexible y fácil de usar ya que podemos interactuar con él por medio del ratón Para instalar **htop** ejecutamos **sudo apt-get install htop**.

Otras herramientas de monitorización que funcionan en modo texto son: **sar** (monitorización en tiempo real del a CPU, RAM, la E/S...), **iostat** (estadísticas de CPU, E/S disco y uso NFS), **mpstat** (estadísticas de la CPU), **vmstat** (info sobre uso de la memoria virtual), **ps tree** (procesos ejecutados en forma de árbol), **uptime** (tiempo que lleva funcionando el sistema), **free** (info de la memoria RAM y la virtual), **mpmap** (mapa de memoria de un proceso), **w** o **who** (usuarios autenticados en el sistema)...

Matar un proceso:

Matar un proceso significa interrumpirlo de forma que no sea posible volver a ponerlo en ejecución. Formas de hacerlo:

- Con la combinación de teclas Ctrl+C mientras el proceso está en primer plano
- Con el comando kill cuando el proceso se encuentra en segundo plano: kill [-señal] PID

Suspender un proceso

Suspender un proceso significa enviar una señal para que suspenda su ejecución por el momento. El proceso permanecerá a la espera para continuar ejecutándose más tarde, cuando reciba la señal adecuada. El sistema pasa a segundo plano los procesos suspendidos. Un proceso, una vez está en ejecución, se suspende con la combinación de teclas Ctrl+Z

Reanudar un proceso:

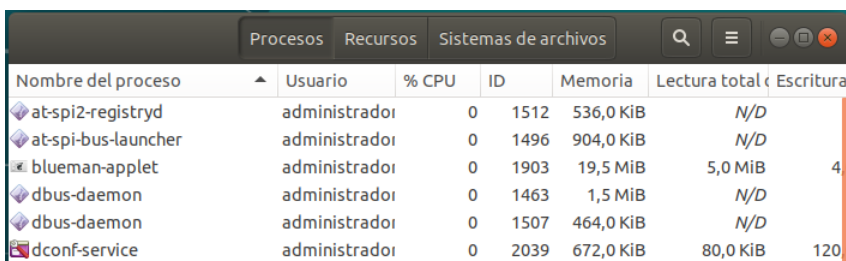
Un proceso que se encuentra suspendido en segundo plano, puede ser devuelto al primer plano con el comando fg. Esto supone volver a ponerlo en ejecución.

fg %numero_de_tarea (puedes ver el número de tarea mediante el comando Jobs)

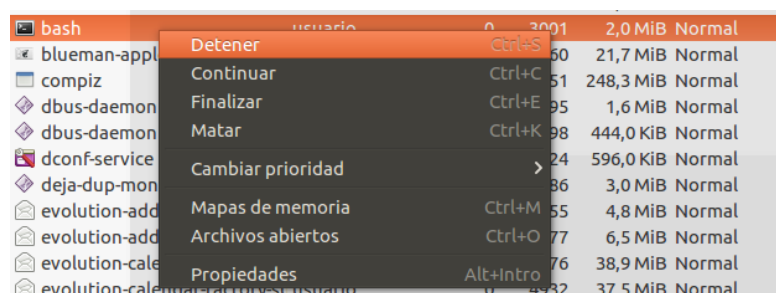
3.2. MODO GRÁFICO: MONITOR DEL SISTEMA

En modo gráfico otra manera de monitorizar los procesos del sistema es mediante el “Monitor del sistema”.

En la pestaña “procesos” se ofrece la misma información que con top.



Nombre del proceso	Usuario	% CPU	ID	Memoria	Lectura total	Escritura
at-spi2-registr...	administrador	0	1512	536,0 KiB	N/D	
at-spi-bus-launcher	administrador	0	1496	904,0 KiB	N/D	
blueman-applet	administrador	0	1903	19,5 MiB	5,0 MiB	4...
dbus-daemon	administrador	0	1463	1,5 MiB	N/D	
dbus-daemon	administrador	0	1507	464,0 KiB	N/D	
dconf-service	administrador	0	2039	672,0 KiB	80,0 KiB	120...

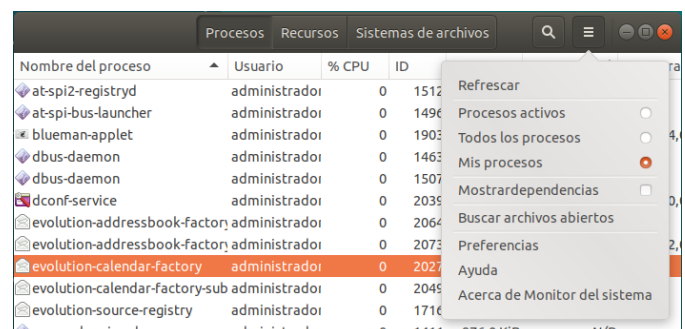


Nombre del proceso	Usuario	% CPU	ID	Memoria	Normal
bash	usuario	0	3001	2,0 MiB	Normal
blueman-applet	usuario	0	150	21,7 MiB	Normal
compiz	usuario	0	151	248,3 MiB	Normal
dbus-daemon	usuario	0	155	1,6 MiB	Normal
dbus-daemon	usuario	0	158	444,0 KiB	Normal
dconf-service	usuario	0	124	596,0 KiB	Normal
deja-dup-mon	usuario	0	136	3,0 MiB	Normal
evolution-add	usuario	0	155	4,8 MiB	Normal
evolution-add	usuario	0	177	6,5 MiB	Normal
evolution-cale	usuario	0	176	38,9 MiB	Normal
evolution-cale	usuario	0	1532	37,5 MiB	Normal

Con el botón derecho encima de un proceso podemos: detenerlo, continuarlo, finalizarlo, matarlo, cambiar su prioridad...

También podemos seleccionar el tipo de procesos que queremos ver.

Por otro lado, en la pestaña “recursos” se ve el histórico de consumo de procesador, memoria y red.



Nombre del proceso	Usuario	% CPU	ID	Memoria	Normal
at-spi2-registr...	administrador	0	1512		
at-spi-bus-launcher	administrador	0	1496		
blueman-applet	administrador	0	1903		
dbus-daemon	administrador	0	1463		
dbus-daemon	administrador	0	1507		
dconf-service	administrador	0	2039		
evolution-addressbook-factory	administrador	0	2064		
evolution-addressbook-factory	administrador	0	2073		
evolution-calendar-factory	administrador	0	2027		
evolution-calendar-factory-sub	administrador	0	2045		
evolution-source-registr...	administrador	0	1716		
gnome-keyring-daemon	administrador	0	1411	976,0 KiB	N/D

4. SERVICIOS

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

Ejemplos de servicios que frecuentemente se cargan en el sistema son, entre otros:

- **acpid**: servicio para el control de ahorro de energía. Se usa para que apague 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 forma 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 ello 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 nombre_servicio status/start/stop/restart`
- Usando el **script init**: `/etc/init.d/nombre_servicio status/start/stop/restart`

*Nota: Desde la versión 9.10 de Ubuntu, comenzó 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 ello, es posible que no funcione para algún servicio usar el script **ini**. En estos casos, o bien usamos **service**, o **sudo start/stop/restart nombre_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`

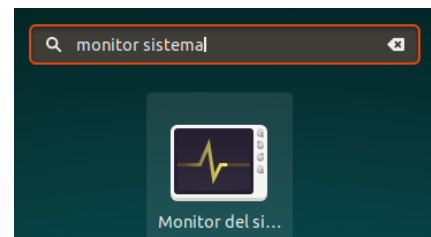
5. RENDIMIENTO Y MONITORIZACIÓN DE SISTEMA

5.1. EN MODO COMANDO

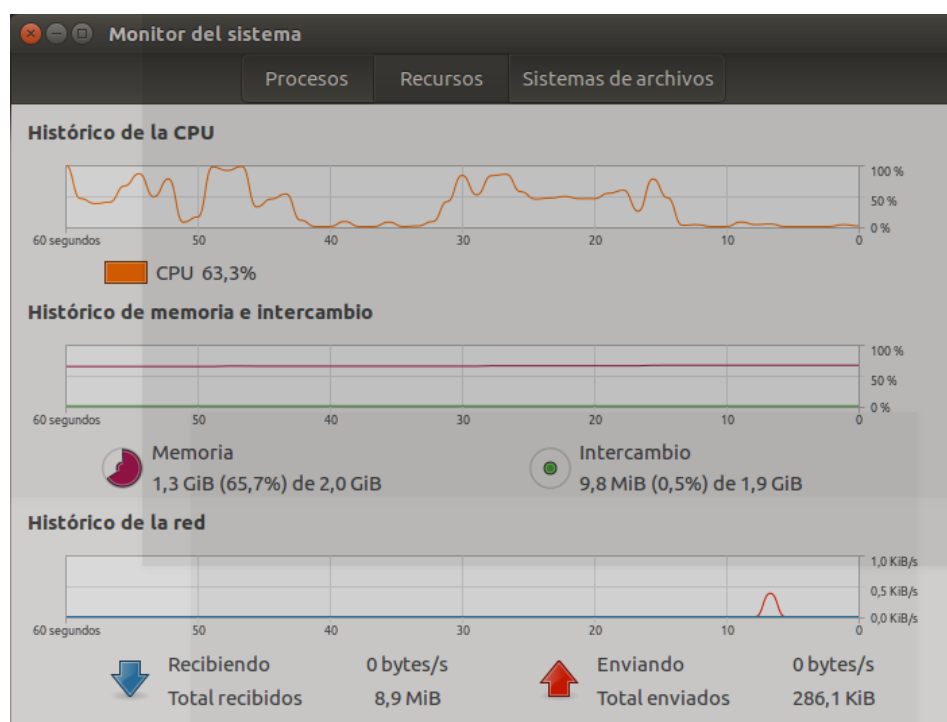
A la hora de monitorizar el sistema, podemos hacer uso de los comandos vistos en los apartados de procesos y servicios: ps, top, htop...

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	
bamfd daemon	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 SUCECOS (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:

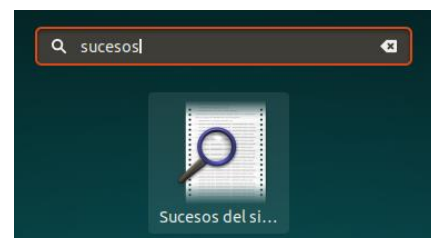
- auth: registro con mensajes relativos a la seguridad y a las autorizaciones
- cron: mensajes sobre demonios periódicos como cron, anacron, at...
- daemon: mensajes sobre otros demonios del sistema
- kern: mensajes relacionados con el núcleo
- lpr: mensajes relativos al subsistema de impresión
- syslog: 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 sólo las 10 últimas líneas del archivo auth.log
- `cat /var/log/auth.log | grep "lightdm"` nos permite ver sólo las líneas que contienen el texto lightdm.

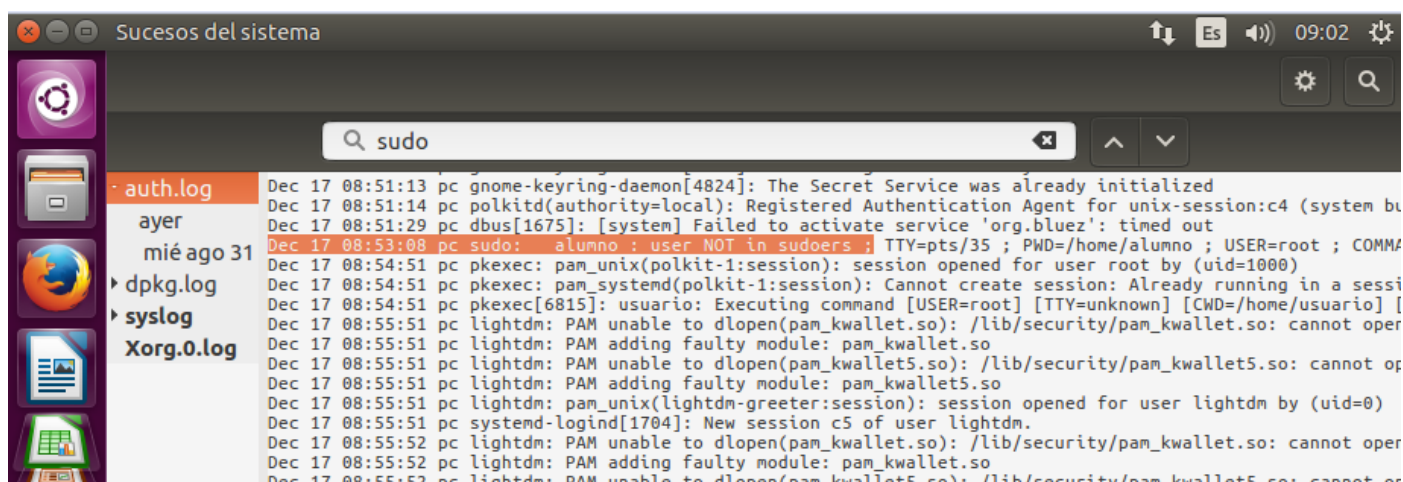
En modo gráfico podemos ver los archivos de sucesos del sistema mediante la utilidad con el mismo nombre "Sucesos del sistema".

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 DISPOSITIVOS HARDWARE

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

- En modo comando:

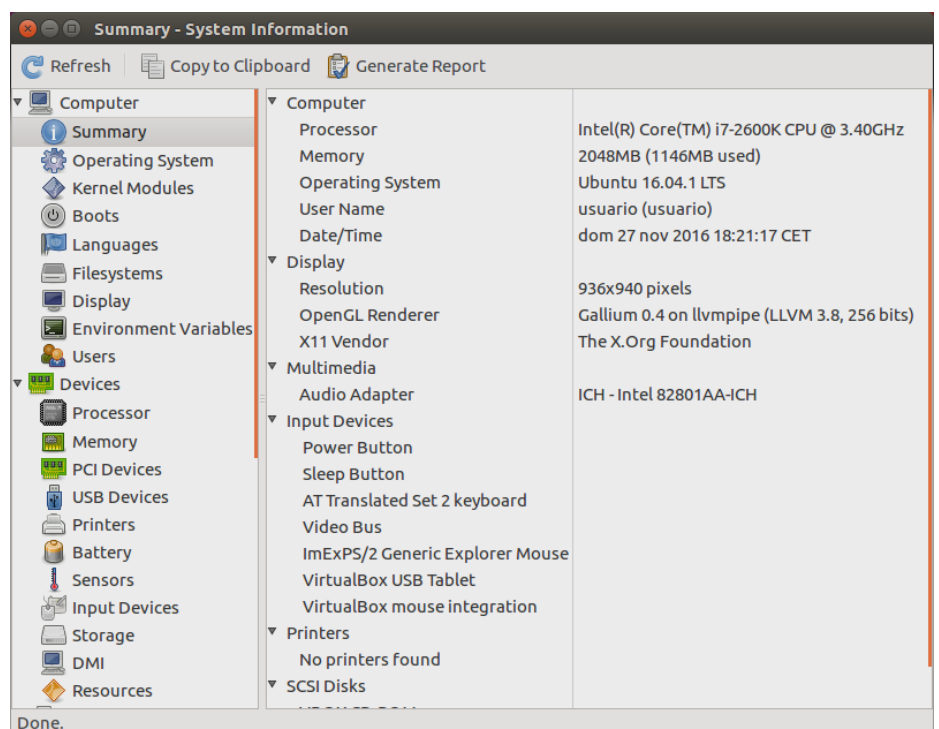
```
sudo lshw
```

```
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-8  
Content-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  
  fabricantes: 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@00
```

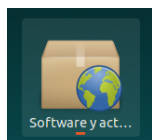
- En modo gráfico podemos instalar la aplicación hardinfo. Para ello, abrimos un terminal y escribimos:

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

Una vez instalada, podemos ejecutarla:



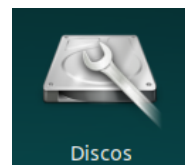
6.2. CONTROLADORES PRIVATIVOS



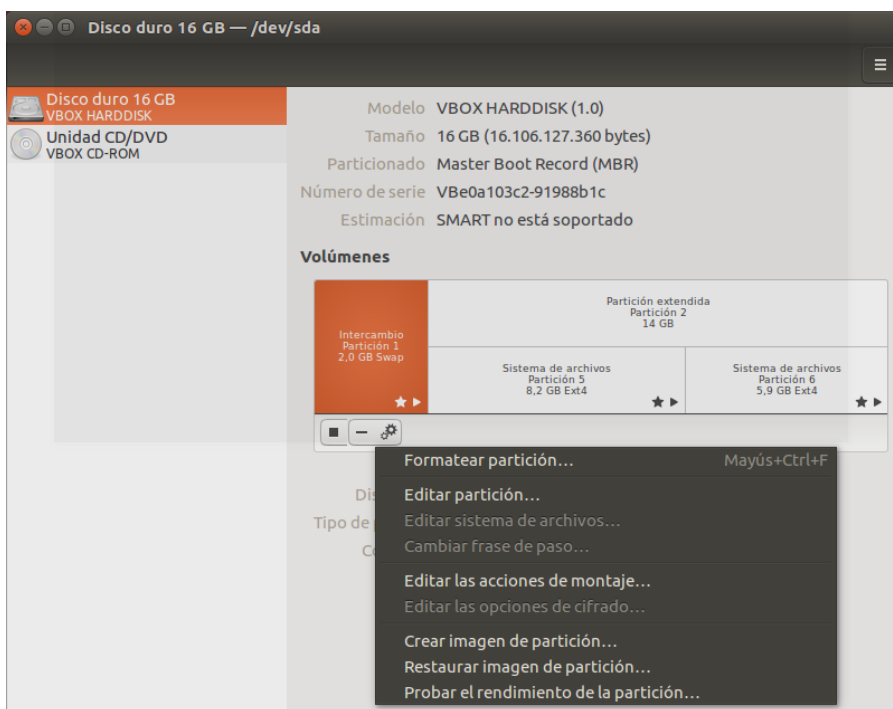
Si lo que queremos es añadir controladores de dispositivos HW ya instalados o actualizar los controladores existentes, tendremos que instalar en nuestro equipo lo que en Ubuntu se llaman **controladores privativos**. Estos controladores HW son suministrados por los fabricantes de HW para Ubuntu. El resto de controladores son genéricos suministrados por el propio SO. Solamente podremos instalar nuevos controladores HW si el fabricante los suministra. Para ello, iremos a “**software y actualizaciones**” -> pestaña “controladores adicionales”.

6.3. GESTIÓN DE DISCOS

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



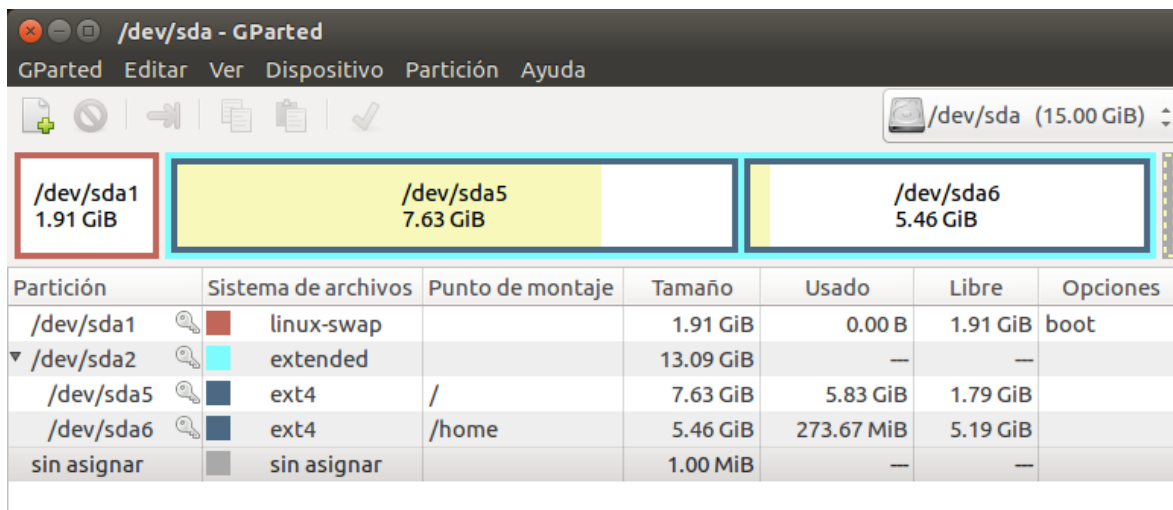
Dado que las posibilidades de la herramienta “discos” son muy limitadas, podemos hacer uso de una aplicación mucho más completa: “Gparted”, la cual nos permitirá realizar todo tipo de operaciones con las particiones.



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



Una vez instalada, la 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 delante del 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 deben estar en funcionamiento para que 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 tanto, a diferencia de `cron`, las tareas encomendadas a `at` solamente 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 11:55 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 el id de la tarea. Si queremos borrar la tarea con id 3
`atrm 1`
- Ver los detalles de la tarea programada: se usa `at -c` seguido del id 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, solamente 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 a que el usuario introduzca la serie de comandos que se quiere ejecutar ese día y hora. Para terminar 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 forma automática... El demonio crond se despierta cada minuto y comprueba los crontabs para determinar lo 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:

Para crear una tarea usando cron se realizaría del siguiente modo:

- En el caso de Ubuntu 20 se realizaría mediante el comando:
 - `sudo crontab -e` (y elegimos, por ejemplo, el editor `/bin/nano`)
- En Ubuntu 18 es necesario editar el archivo `/etc/crontab`
 - `sudo nano /etc/crontab`

Una vez realizado esto aparece el contenido del archivo a editar. Este fichero está dividido en líneas y cada línea representa una tarea programada. Estas líneas tienen el siguiente formato:

minuto hora día mes día_semana 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 lo 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 al año.

@daily: Se ejecuta una vez al día.

@monthly: Se ejecuta una vez al 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 arranque 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 cada enero y 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 solo 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 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 u] 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
 - u usuario: especifica el usuario propietario de la tarea (normalmente, esta opción la 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 las mismas:


```
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, sólo 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 caso que queramos denegar el acceso a las tareas programadas, se deberá 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 a todos los usuarios que ejecuten trabajos vía `crontab`. Sin embargo, si creas un fichero `cron.allow` o `cron.deny` en blanco el comportamiento del demonio cambia. En ese momento solamente 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 lo que hay que hacer. Por tanto, busca ficheros en `/var/spool/cron` para ejecutarlos a 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 ejecuten 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 a 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 por estar el sistema apagado). `Anacron` se configura mediante el fichero `/etc/anacrontab`

8. COPIAS DE SEGURIDAD

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

Además de decidir dónde las vamos a almacenar y qué tipo de copias de seguridad vamos a realizar (total o integral, incremental o diferencial) es muy importante elegir qué información vamos a escoger 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 lo que ha provocado un fallo

8.1. MODO COMANDO

Una opción es realizar la copia de seguridad en formato comprimido para que ocupe 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”).

8.2. MODO GRÁFICO



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



9. GESTOR DE ARRANQUE: GRUB

9.1. 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 el kernel de Linux.

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

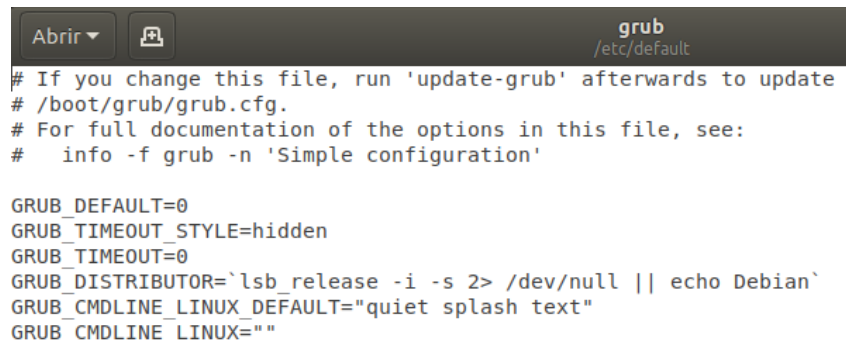
- Los archivos `grub-install`, `grub-setup`, `grub-mkconfig...` del directorio `/usr/sbin/`

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 ello hay que contar la posición que ocupa en el menú el SO con el que 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.



```
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=""
```

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

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

Nota: no debemos modificarlo directamente del fichero `/boot/grub/grub.cfg`, ya que si el SO 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 ha de modificarse el GRUB a través del fichero `/etc/default/grub`

10. RUNLEVELS O NIVELES DE ARRANQUE

10.1. RUNLEVELS EN LA MAYORÍA DE DISTRIBUCIONES LINUX

La mayoría de las distribuciones Linux usan el concepto de **nivel de arranque o ejecución (runlevel)** para indicar el **modo o configuración de arranque del sistema operativo**. En la práctica se emplean los siguientes runlevels o niveles de arranque o ejecución:

runlevel	Significado
0	apagar el equipo
1	arrancar en modo monousuario como root (se suele usar para analizar y reparar problemas)
2	arrancar en modo multiusuario sin soporte de red
3	arrancar en modo multiusuario con soporte de red
4	arrancar en modo multiusuario con soporte de red (con el 3, pero no se suele usar)
5	arrancar en modo multiusuario con soporte de red y entorno gráfico
6	reiniciar el equipo

De este modo, cuando realizamos un inicio normal, con interface gráfica, estaremos usando el nivel de ejecución 5. Si necesitamos un inicio normal, pero sin interface gráfica, usaremos el nivel de ejecución 3.

Para cambiar de nivel de ejecución sólo hay que ejecutar el comando `init` seguido del número del runlevel. Ejemplos:

- `init 0`: Cambia al runlevel 0 (se apaga el sistema, equivalente al comando `halt`).
- `init 2`: Cambia al runlevel 2.
- `init 6`: Cambia al runlevel 6 (reinicia el sistema, equivalente al comando `reboot`)

Cuando usamos este tipo de distribuciones, si queremos cambiar el runlevel por defecto, es decir, el modo en que arranca el sistema por defecto, sólo hay que cambiar una línea en el archivo `/etc/inittab`, concretamente:

```
id:N:initdefault:
```

donde N es el runlevel por defecto.

10.2. RUNLEVELS EN UBUNTU

Sin embargo, Ubuntu utiliza un mecanismo de inicio diferente, llamado Upstart, y no existe el archivo `/etc/inittab`. Además, también se ha simplificado el uso de los niveles de ejecución, habiendo quedado de esta forma:

runlevel	Significado
0	Apagado (<code>poweroff.target</code>)
1	Monousuario (<code>rescue.target</code>) para rescate
2	Modo texto (<code>multi-user.target</code>)
3	No usado (<code>multi-user.target</code>) como el 2
4	No usado (<code>multi-user.target</code>) como el 2
5	Modo gráfico (<code>graphical.target</code>)
6	Reinicio (<code>poweroff.target</code>)

Así pues, en Ubuntu el nivel de ejecución se guarda en la variable `DEFAULT_RUNLEVEL`, dentro del archivo `/etc/init/rc-sysinit.conf`. Una de las formas más sencillas que tenemos para cambiarlo es a través de comandos del sistema, como podemos ver en el siguiente apartado.

10.3. INICIAR EL SISTEMA EN MODO TEXTO O EN MODO GRÁFICO


El comando `sudo systemctl set-default multi-user.target` forzaría a que el sistema arrancase en modo texto por defecto. En cambio `systemctl set-default graphical.target` forzaría el modo gráfico. Después de esto, sólo nos faltará reiniciar el sistema para comprobar que todo ha sido correcto. Cuando termine el reinicio, podremos identificarnos con nuestro usuario y contraseña habituales.

10.4. INICIAR LA INTERFAZ GRÁFICA DESDE EL MODO TEXTO

Aunque hayamos configurado el sistema para que arranque en modo texto, si queremos iniciar temporalmente la interfaz gráfica, sólo tenemos que ejecutar cualquiera de los siguientes comandos:

```
systemctl start graphical.target  
systemctl start gdm3.service
```

Y tras unos instantes dispondremos de la interfaz gráfica.

Cuando hayamos terminado el trabajo con la interfaz gráfica, sólo tendremos que hacer clic sobre el icono  y en el menú de contexto que aparece elegir Cerrar sesión. Tras la solicitud de confirmación, la sesión gráfica se cierra y estaremos de vuelta en el modo texto.