

Administración de Ubuntu 22.04 LTS

En este capítulo nos introduciremos en la Administración de Ubuntu. Veremos algunas herramientas gráficas que incorpora Ubuntu para la administración del sistema y los comandos habituales en CLI. Comenzaremos con la gestión de usuarios y grupos. Después, seguiremos con la instalación de aplicaciones, la administración de procesos, la configuración de la red y terminaremos con la gestión de la impresión.



Administración en Ubuntu 20.04 LTS by Rafael Lozano is licensed under a [Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 España License](https://creativecommons.org/licenses/by-nc-sa/3.0/es/).

Tabla de contenido

1	Introducción	1
1.1	Usuarios administradores	1
1.2	Configuración	2
2	Usuarios y grupos	3
2.1	Bases de datos de usuarios y grupos.....	4
2.1.1	Identificadores de usuario y grupo	4
2.1.2	Archivo /etc/passwd	4
2.1.3	Archivo /etc/group	6
2.2	Gestión de usuarios	6
2.2.1	Gestión de usuarios en formato gráfico	6
2.2.2	Comando useradd	8
2.2.3	Comando usermod	9
2.2.4	Comando userdel	11
2.2.5	Comandos adduser y addgroup.....	11
2.3	Gestión de grupos	13
2.3.1	Comando groupadd	13
2.3.2	Comando groupmod	13
2.3.3	Comando groupdel.....	14
2.3.4	Comando gpasswd	14
2.3.5	Comando newgrp	14
2.4	El archivo /etc/sudoers	15
3	Administración de contraseñas	17
3.1	El archivo /etc/shadow	17
3.2	El archivo /etc/gshadow	19
3.3	Establecer la contraseña de usuario. Comando passwd.....	20
3.4	Políticas de contraseña	21
3.4.1	Complejidad de la contraseña. El archivo /etc/security/pwquality.conf.....	21
3.4.2	Complejidad de la contraseña. Librería pam_pwquality	23
3.4.3	Bloqueo de cuenta	27
3.4.4	Vigencia de la contraseña.....	28
4	Dispositivos hardware y controladores	30
4.1	Instalación de un controlador propietario.....	31
4.2	Gestión de controladores adicionales	32
5	Instalación de aplicaciones	33
5.1	Introducción a los paquetes binarios	34
5.1.1	Formato de un paquete en Debian.....	34
5.1.2	Dependencias de paquetes.....	35
5.1.3	Lista de repositorios	36
5.1.4	Software y actualizaciones	39
5.2	Instalación de aplicaciones desde Ubuntu Software.....	40

5.2.1	Instalar una aplicación.....	41
5.2.2	Desinstalar una aplicación.....	42
5.2.3	Añadir una reseña.....	42
5.3	Instalación de paquetes binarios. Comandos APT.....	43
5.3.1	Comando apt.....	43
5.3.2	Comando apt-get.....	45
5.3.3	Comando apt-cdrom.....	45
5.4	Snap.....	46
5.5	Instalación de paquetes binarios con dpkg.....	47
5.6	Añadir repositorios de terceros.....	47
5.6.1	Añadir un PPA a la lista de repositorios e instalar el software.....	48
5.6.2	Eliminar un PPA.....	48
5.7	Instalación de aplicaciones desde código fuente.....	49
5.7.1	Preparar el sistema para la construcción de paquetes.....	50
5.7.2	Descargar el código fuente de la aplicación.....	50
5.7.3	Resolviendo las dependencias.....	51
5.7.4	Construir e instalar la aplicación.....	52
5.7.5	Ejemplo de instalación.....	52
5.8	Actualizaciones automáticas.....	56
6	Procesos.....	59
6.1	Listado de procesos.....	59
6.1.1	Comando ps.....	60
6.1.2	Comando top.....	61
6.2	Enviar señales a procesos.....	63
6.3	Ejecutar procesos en 2º plano.....	64
6.4	Detención e inicio del sistema.....	65
6.4.1	Comando systemctl.....	65
6.4.2	Comando shutdown.....	66
6.5	Tareas programadas.....	66
6.5.1	Comando at.....	66
6.5.2	Cron y crontab.....	67
6.6	Monitor del sistema.....	69
7	Servicios.....	71
7.1	Arranque del sistema.....	72
7.2	Systemd.....	73
7.3	Arranque, parada y reinicio de servicios.....	74
7.4	Estado de un servicio.....	74
7.5	Habilitar o deshabilitar un servicio.....	75
7.6	Información de un servicio.....	75
7.7	Objetivos (niveles de ejecución).....	76
8	Configuración de la red.....	77
8.1	Interfaces de red.....	77
8.2	NetPlan.....	78
8.2.1	Configuración con Network Manager.....	79
8.2.2	Configuración con systemd-networkd.....	82
8.2.3	Configuración con Netplan.....	84

8.2.4 Precedencia en la configuración.....	85
8.3 Servicio networking.....	85
8.3.1 El archivo /etc/network/interfaces.....	85
8.3.2 Activar y desactivar las interfaces de red.....	87
8.3.3 El comando ifconfig.....	88
8.4 Interfaz inalámbrica	89
8.4.1 Netplan y servicio systemd-networkd	90
8.4.2 Servicio networking.....	92
8.5 Comando ip.....	92
8.5.1 Comprobar información de las interfaces de red.....	94
8.5.2 Habilitar o deshabilitar las interfaces de red	95
8.5.3 Configurar las interfaces de red.....	95
8.6 Resolución de nombres.....	95
8.6.1 Archivo /etc/hosts	96
8.6.2 Servicio systemd-resolved.....	97
8.6.3 Archivo /etc/nsswitch.conf.....	99
8.6.4 Resolución de nombres en /etc/network/interfaces	99
9 Bibliografía	101

Administración de Ubuntu

1 Introducción

La administración de un sistema Linux es una tarea muy amplia para centralizarla en una sola herramienta o utilidad. Lo habitual es combinar comandos del sistema operativo con herramientas gráficas. Sin embargo, mientras que las herramientas gráficas son específicas de cada distribución, los comandos del sistema operativo son comunes en la mayoría de ellas. Además, la mayoría de las herramientas gráficas son *front-ends* de comandos del sistema operativo. En el caso de Ubuntu 20.04 LTS, las herramientas gráficas están en su mayor parte en *Configuración*.

1.1 Usuarios administradores

Las tareas de administración en un sistema Linux solamente pueden ser hechas por usuarios administradores. En sistemas Linux el superusuario es `root` que por defecto está inhabilitado, sin embargo todo usuario que pertenece a los grupos `admin` o `sudo` puede realizar tareas administrativas. Por defecto, el usuario que se creó durante la instalación del sistema pertenece al grupo `sudo`.

Cada vez que un usuario de los grupos `admin` o `sudo` ejecuta una aplicación gráfica que puede cambiar la configuración del sistema le aparecerá un cuadro de autenticación en el que tendrá que introducir su contraseña.

El usuario administrador tendrá que introducir su contraseña correctamente para poder continuar. En el caso necesitar ejecutar un comando como el usuario `root` tendremos que precederlo de `sudo`.

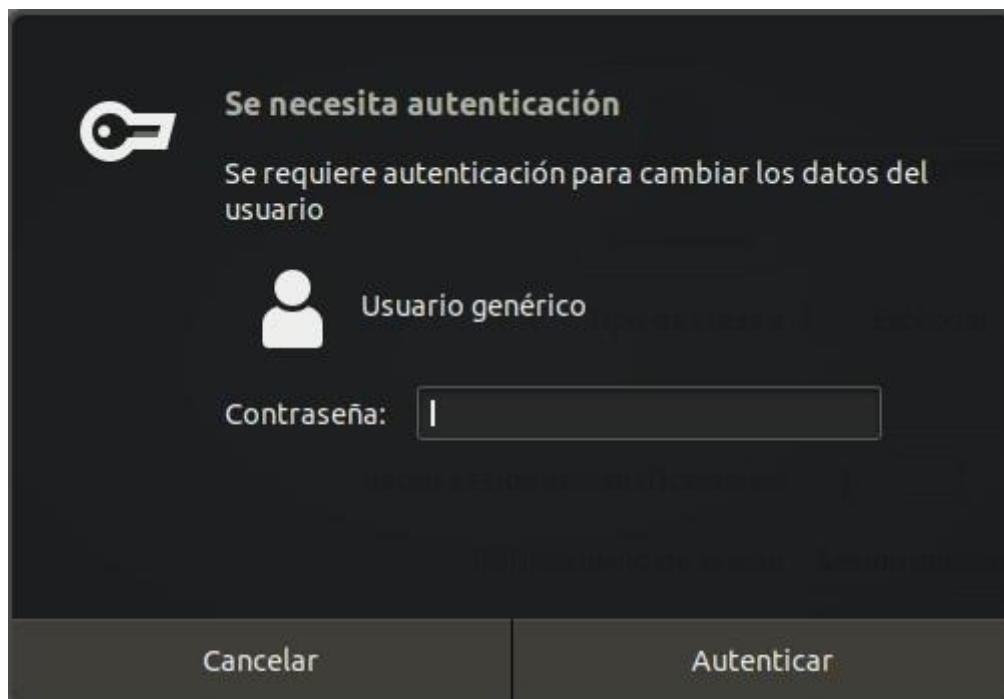


Figura 1.- Autenticación para tareas administrativas

Cuando el usuario precede con `sudo` la ejecución de un comando, también se le solicitará su contraseña antes de proceder a la ejecución del mismo. Después de introducir su contraseña dispone de quince minutos para continuar ejecutando comandos con `sudo` sin que se le vuelva a solicitar la contraseña. Pasado ese tiempo tendrá que volver a introducirla con el siguiente comando que ejecute con `sudo`. En el siguiente ejemplo se ha hecho un listado de las particiones del disco duro.

```
usuario@PC00: ~
usuario@PC00:~$ sudo fdisk -l /dev/sda
[sudo] password for usuario:
Disk /dev/sda: 15 GiB, 16106127360 bytes, 31457280 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: A9B9781A-7E03-49F5-9545-032F67350878

Disposit.    Start      Final  Sectores  Size Tipo
/dev/sda1      2048    1023999    1021952  499M EFI System
/dev/sda2    1024000  17031167  16007168  7,6G Linux filesystem
/dev/sda3    17031168  27457535  10426368   5G Linux filesystem
/dev/sda4    27457536  31455231   3997696  1,9G Linux swap
usuario@PC00:~$
```

Figura 2.- Ejecución de comando como administrador

1.2 Configuración

En Ubuntu 20.04 disponemos de *Configuración*, un lugar donde están centralizadas la mayoría de las aplicaciones de administración.

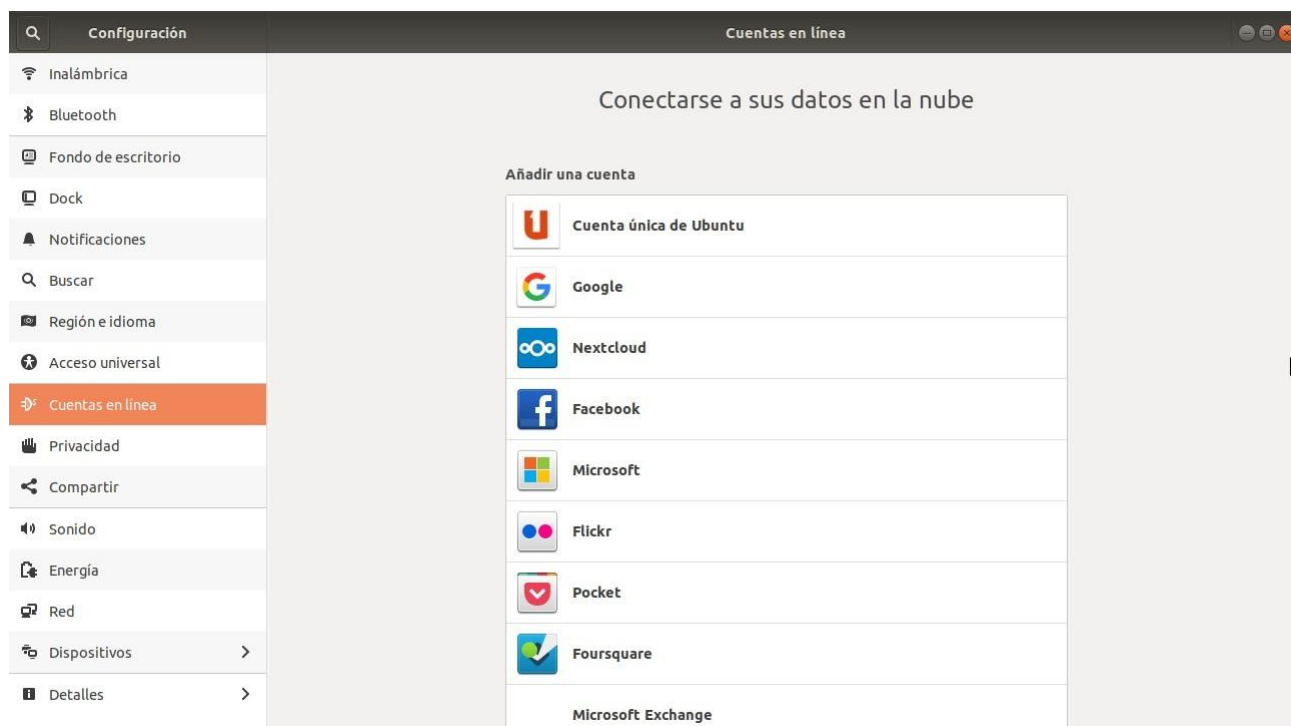


Figura 3.- Configuración del sistema

Para acceder a ella hay que seguir los siguientes pasos:

1. Hacer clic en el icono de flecha abajo del área de notificaciones.
2. En la parte inferior izquierda hacer clic en el botón de *Configuración*.

También podemos hacer una búsqueda en *Aplicaciones con Configuración*. Como vemos en la imagen anterior las aplicaciones y herramientas para realizar tareas administrativas están en el panel izquierdo.

Una pequeña parte de la gestión de la configuración del sistema se hará desde este lugar, si bien va a ser necesario realizar también esta tarea mediante comandos en línea. Para tener una visión global de la configuración del sistema Linux emplearemos los dos métodos (aplicación gráfica y comando en línea) donde sea posible.

2 Usuarios y grupos

Linux es un sistema operativo multiusuario. Ya sea de forma local o por acceso remoto, los usuarios pueden abrir sesión en el sistema de forma simultánea. Por tanto la tarea de añadir y mantener las cuentas de usuario es frecuente en la administración de sistemas Linux.

Además, en un sistema Linux existen cuentas de usuario para personas y para los procesos del sistema, ya que es frecuente que estos necesiten una cuenta de usuario específica para controlar los privilegios y los derechos de acceso.

Antes de proceder a ver como se gestionan los usuarios en Linux hay que ver las

bases de datos de usuarios y grupos en Linux.

2.1 Bases de datos de usuarios y grupos

Hay tres tipos de cuentas usuarios en un sistema Linux: `root`, cuentas del sistema y usuario estándar. El usuario `root` es único por varias razones. La primera y principal es la única cuenta de usuario con privilegios sobre todo el sistema. Se pueden configurar otras cuentas como un clon exacto de la cuenta del usuario `root`, pero esto es muy poco aconsejable. El usuario `root` es con frecuencia llamado el superusuario ya que tiene derechos y acceso para realizar cualquier tarea o visualizar cualquier archivo del sistema. Básicamente, no hay nada que el superusuario no pueda hacer.

En Linux también existen una serie de cuentas configuradas, como `bin`, `daemon`, `adm`, `lp`, `sync`, `mail`, etc. Estas cuentas también son especiales. Se denominan cuentas del sistema y tienen varios objetivos. No gozan de privilegios de que disfruta el usuario `root`, solo están para ejecutar determinadas aplicaciones en lugar de usar el usuario `root` para proteger al sistema de posibles vulnerabilidades de seguridad. Estas cuentas no tienen contraseña, porque no están diseñadas para poder iniciar sesión con ellas.

Por último, la cuenta de usuario estándar es el tipo de cuenta que se configura para que cada usuario individual pueda iniciar sesión en el sistema.

2.1.1 Identificadores de usuario y grupo

Cada usuario tiene un número único que lo identifica en el sistema. Este número se denomina UID (*User IDentifier*) y es un valor entre 0 y 65536. Generalmente a la hora de asignar un UID a un usuario se sigue el siguiente esquema:

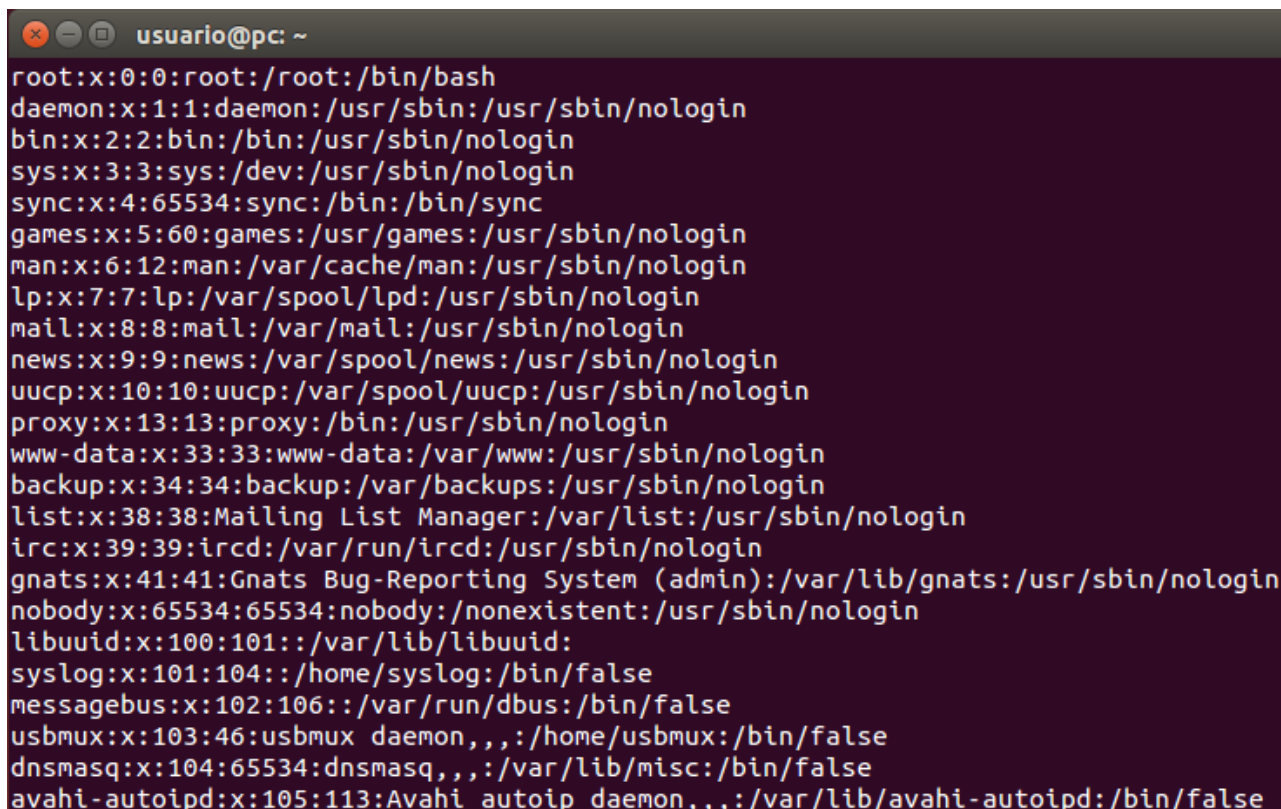
- ✓ El UID 0 es para el usuario `root`.
- ✓ De 1 a 99 son para las cuentas del sistema.
- ✓ De 100 a 999 son para programas instalados por usuarios y demonios.
- ✓ De 1000 a 29999 son para cuentas de usuario estándar.
- ✓ De 30000 a 65533 están reservadas, pero pueden usarse si se quiere.

65534 pertenece al usuario `nobody`, una cuenta sin derechos ni permisos. Similar a la cuenta de Invitado en Windows.

Los grupos también tienen un número que los identifica, el GID (*Groud IDentifier*). Para usuarios estándar suele ser normal crear un grupo con el mismo que el usuario. También el GID de este grupo coincide con el UID del usuario.

2.1.2 Archivo `/etc/passwd`

Este archivo contiene las cuentas de usuario registradas en el sistema. Es un archivo de texto ASCII donde cada línea representa un usuario. La información de cada usuario está dividida en siete campos delimitados por el carácter dos puntos (:). La siguiente imagen muestra el contenido de este archivo.

A terminal window titled 'usuario@pc: ~' displays the contents of the /etc/passwd file. The output shows system users like root, daemon, bin, sys, sync, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, irc, gnats, nobody, libuuid, syslog, messagebus, usbmux, dnsmasq, and avahi-autoipd, each with their respective UID, GID, name, and shell path.

```
usuario@pc: ~  
root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
sys:x:3:3:sys:/dev:/usr/sbin/nologin  
sync:x:4:65534:sync:/bin:/bin/sync  
games:x:5:60:games:/usr/games:/usr/sbin/nologin  
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin  
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin  
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin  
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin  
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin  
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin  
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin  
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin  
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin  
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin  
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
libuuid:x:100:101::/var/lib/libuuid:  
syslog:x:101:104::/home/syslog:/bin/false  
messagebus:x:102:106::/var/run/dbus:/bin/false  
usbmux:x:103:46:usbmux daemon,,,:/home/usbmux:/bin/false  
dnsmasq:x:104:65534:dnsmasq,,,:/var/lib/misc:/bin/false  
avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
```

Figura 4.- Archivo /etc/passwd

La descripción de cada uno de los campos es la siguiente:

- ✓ Login.- Nombre de la cuenta de usuario y que éste utiliza para abrir sesión. Tiene que ser único.
- ✓ Contraseña.- Todos tienen una x minúscula para indicar que se usan contraseñas ocultas. Las contraseñas reales encriptadas se almacenan en otro archivo aparte por motivos de seguridad.
- ✓ UID.- Identificador de usuario. Tiene que ser único.
- ✓ GID.- Identificador de grupo. Aquí solo aparece el grupo principal del usuario, pero un usuario puede pertenecer a más de un grupo.
- ✓ Comentario.- Normalmente contiene el nombre completo del usuario.
- ✓ Carpeta personal del usuario.- Indica el directorio HOME del usuario y donde se colocará cuando abra sesión.
- ✓ Intérprete de comandos.- Indica la Shell del usuario. Habitualmente será bash.

Cuando se gestionan los usuarios, las herramientas empleadas modifican el contenido de este fichero. Aunque puede añadirse nuevas líneas para crear nuevos usuarios, es mejor utilizar las herramientas habilitadas para ello.

2.1.3 Archivo /etc/group

De forma análoga a los usuarios, los grupos se almacenan en un fichero de texto ASCII, el fichero /etc/group. Al igual que con los usuarios, cada línea se refiere a un grupo del sistema y tiene el siguiente formato:

- ✓ Nombre del grupo.- Tiene que ser único y como máximo tiene 8 caracteres.
- ✓ Contraseña del grupo.- Este campo está vacío porque las contraseñas de grupo se almacenan en otro archivo.
- ✓ GID.- Identificador de grupo
- ✓ Miembros.- Lista de usuarios que pertenecen a este grupo.

2.2 Gestión de usuarios

A continuación vamos a ver las operaciones habituales en la gestión de usuarios, comenzando por la herramienta del Centro de control para esta tarea.

2.2.1 Gestión de usuarios en formato gráfico

En la categoría *Sistema* de la *Configuración del Sistema* disponemos de la herramienta *Cuentas de usuario* que nos permite hacer la gestión de los usuarios y grupos del sistema.

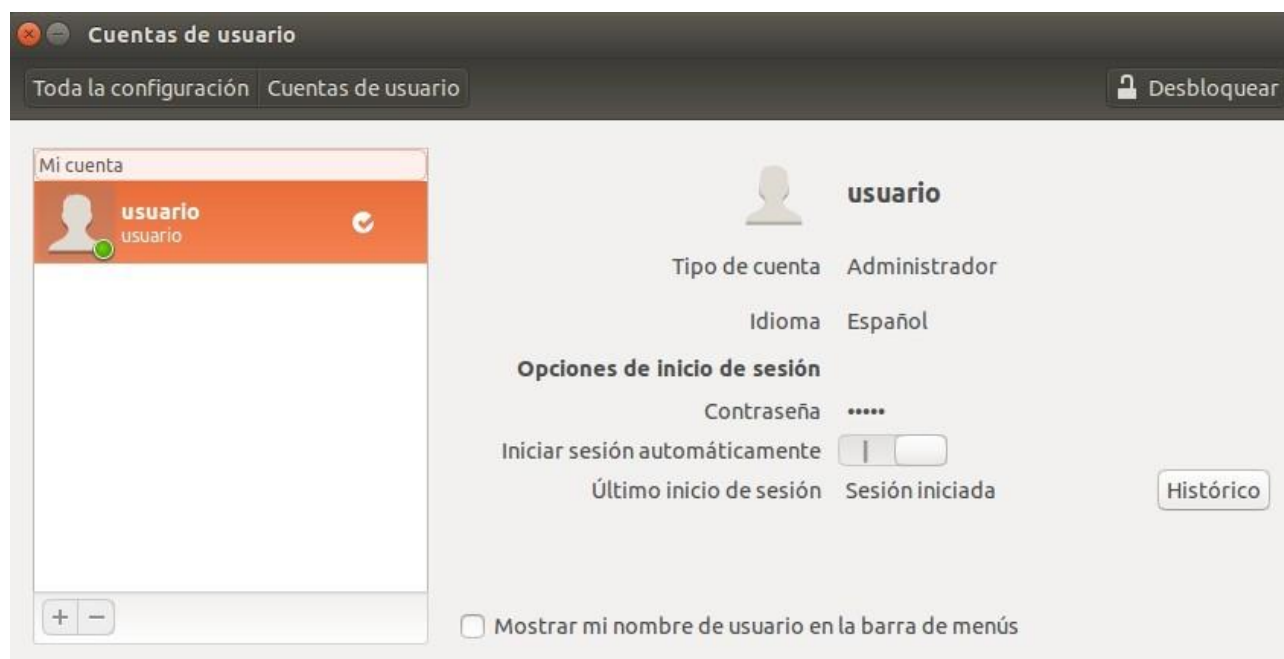


Figura 5.- Cuentas de usuario

En la parte izquierda veremos un panel con la lista de los usuarios que actualmente hay en el sistema. Si queremos añadir un nuevo usuario deberemos seguir los siguientes pasos:

1. Primero haremos clic en el botón *Desbloquear* y nos autenticaremos para realizar tareas administrativas.


2. Posteriormente hacer clic en el botón .
3. En la ventana que aparece introducir el nombre completo y el login. En la lista *Tipo de cuenta* indicaremos si el nuevo usuario es administrador o es una cuenta de usuario estándar sin privilegios. Posteriormente hacer clic en el botón *Aceptar*.



Figura 6.- Crear un usuario

Una vez creada la cuenta aparecerá en la lista de usuarios y sus propiedades en el panel derecho. Este usuario no puede abrir sesión hasta que se desbloquee la cuenta. El desbloqueo de la cuenta permite asignarle una contraseña al usuario o permitirle abrir sesión sin contraseña. Si hacemos clic en el botón *cuenta desactivada* elegiremos la acción de desbloqueo a realizar y si elegimos asignarle contraseña, tendremos que escribir la contraseña dos veces.


The image shows a window titled "Cambiando la contraseña para María López". It features a user icon and the name "María López". Below this, there is a dropdown menu labeled "Acción" with the option "Establecer una contraseña ahora". Underneath, there are two password input fields: "Contraseña nueva" and "Confirmar contraseña". The "Contraseña nueva" field has a strength indicator bar below it and a gear icon for password requirements. A checkbox labeled "Mostrar contraseña" is located below the confirmation field. At the bottom, there is a link "Cómo elegir una contraseña fuerte" and two buttons: "Cancelar" and "Cambiar".

Figura 7.- Desbloqueo de la cuenta de usuario

También podemos modificar sus propiedades:

- ✓ En *Tipo de cuenta* podemos cambiar la cuenta de un tipo a otro.
- ✓ *Iniciar sesión automáticamente* permite al usuario iniciar una sesión sin necesidad de introducir la clave.
- ✓ El botón *Histórico* nos muestra una lista de los inicios de sesión del usuario.

Por defecto al nuevo usuario se le asigna un UID superior en uno al último asignado para usuarios de escritorio. También se crea automáticamente un grupo con el mismo nombre que el login de usuario.

Para borrar un usuario solo tenemos que seleccionarlo de la lista y hacer clic en el botón . Nos preguntará si queremos además eliminar su carpeta personal. De forma automática borrará el grupo que creó con el mismo nombre.

2.2.2 Comando `useradd`

El comando `useradd` añade un nuevo usuario al sistema. Esta cuenta de usuario nuevo estará deshabilitada hasta que se le asigne una contraseña.

Sintaxis

```
useradd [opciones] login
```

Parámetros

`login`

El login de la nueva cuenta de usuario

Opciones

-c comentario

--coment comentario

Nombre completo del usuario. Si tiene espacios en blanco debe ir entrecomillada.

-d directorio

--home directorio

Directorio HOME del usuario. Por defecto es /home/login

--home directorio

-e fecha

--expiredate fecha

Fecha en que la cuenta expira y será deshabilitada. La fecha en formato yyyy-mm-dd.

-f días_inactiva

--inactive días_inactiva

Días después de que expire la contraseña hasta que la cuenta se desactiva.

-g grupo

--gid grupo

Nombre o GID del grupo inicial del usuario. El grupo debe existir. Si no se especifica se creará un grupo con el mismo nombre que el login de usuario y se le asignará como grupo principal

-G grupo,...

--gropus grupo,...

Una lista de grupos adicionales de los que el usuario también es miembro

-m

--create-home

El directorio HOME será creado si no existe. Además de crearlo copia los archivos de configuración del directorio /etc/skel.

-M

No crea el directorio HOME

-s shell

--shell shell

Nombre del shell del usuario

-u uid

--uid uid

Valor del UID. Debe ser único y no negativo.

-r

--system

Crea una cuenta del sistema

2.2.3 Comando usermod

El comando usermod se emplea para modificar alguna o varias propiedades de una

cuenta de usuario.

Sintaxis

```
usermod [opciones] login
```

Parámetros

login

El login de la nueva cuenta de usuario

Opciones

-c comentario

--coment comentario

Nombre completo del usuario. Si tiene espacios en blanco debe ir entrecomillada.

-d directorio

--home directorio

Directorio HOME del usuario. Por defecto es /home/login

-e fecha

--expiredate fecha

Fecha en que la cuenta expira y será deshabilitada. La fecha en formato yyyy-mm-dd.

-f días_inactiva

--inactive días_inactiva

Días después de que expire la contraseña hasta que la cuenta se desactiva.

-g grupo

--gid grupo

Nombre o GID del grupo inicial del usuario. El grupo debe existir. Si no se especifica se creará un grupo con el mismo nombre que el login de usuario y se le asignará como grupo principal

-G grupo,...

--gropus grupo,...

Una lista de grupos adicionales de los que el usuario también es miembro

-m

--create-home

El directorio HOME será creado si no existe. Además de crearlo copia los archivos de configuración del directorio /etc/skel.

-M

No crea el directorio HOME

-s shell

--shell shell

Nombre del shell del usuario

-u uid

--uid uid

Valor del UID. Debe ser único y no negativo.

`-l login`
`--login login`
Asigna un nuevo login

`-a`
`--append`
Añade al usuario al grupo complementario. Usado solo con la opción `-G`

`-L`
`--lock`
Bloquea la contraseña de usuario añadiéndole al principio un símbolo `!`.

`-m`
`--move-home`
Mueve el contenido del directorio HOME actual del usuario a la nueva localización. Solo se emplea con la opción `-d`

`-U`
`--unlock`
Desbloquea la contraseña de usuario.

2.2.4 Comando `userdel`

El comando `userdel` elimina una cuenta de usuario.

Sintaxis

```
userdel [opciones] login
```

Parámetros

`login`
El login de la nueva cuenta de usuario

Opciones

`-f`
`--force`
Fuerza el borrado del usuario aunque tenga sesión abierta.

`-r`
`--remove`
Borra el directorio HOME y el buzón del usuario.

2.2.5 Comandos `adduser` y `addgroup`

Estos comandos permiten agregar usuarios al sistema entre otras cosas, pero con una interfaz más amigable que los anteriores comandos ya que solicitan la información del nuevo usuario por teclado. Puede trabajar en cinco modos:

- ✓ Ejecutar `adduser` con el login del nuevo usuario como argumento y sin opciones para crear un nuevo usuario.

- ✓ Ejecutar `adduser` con el login del nuevo usuario y la opción `--system` para crear un usuario del sistema.
- ✓ Ejecutar `adduser` con la opción `--group` o `addgroup` sin la opción `--system` creará un nuevo grupo.
- ✓ Ejecutar `addgroup` con la opción `--system` creará un nuevo grupo de sistema.
- ✓ Ejecutar `adduser` con el login de un usuario existente y un grupo existente, añade al usuario al grupo.

Sintaxis

```
adduser [opciones] login
addgroup [opciones] grupo
adduser login grupo
```

Parámetros

login

El login de la nueva cuenta de usuario

grupo

El nuevo grupo o el grupo al que se añade el usuario

Opciones

`--gid` GID

Cuando se crea un grupo, éste tendrá el GID especificado. Cuando se crea un usuario le asigna este GID como grupo inicial.

`--group`

Cuando se usa con `--system` se crea un nuevo grupo con el mismo nombre e ID del usuario. Si no se emplea `--system` se crea un nuevo grupo con el nombre dado.

`--home` directorío

Especifica el directorio del usuario. Si no existe se crea y se copia el perfil del usuario desde `/etc/skel`.

`--shell` shell

Indica la shell del usuario.

`--ingroup` GID

Añade al nuevo usuario al grupo existente en lugar de crear uno por defecto con el mismo nombre e ID del nuevo usuario.

`--no-create-home`

No crea el directorio HOME del usuario.

`--system`

Crear un nuevo usuario o grupo del sistema

`--uid` UID

Especifica el nuevo UID del usuario.

Se recomienda consultar el manual del comando para conocer más detalles del mismo.

2.3 Gestión de grupos

Al igual que los usuarios, disponemos de un conjunto de comandos para gestionar los grupos.

2.3.1 Comando groupadd

El comando groupadd se emplea para añadir un nuevo grupo.

Sintaxis

```
groupadd [opciones] grupo
```

Parámetros

grupo

El nombre del grupo que se crea.

Opciones

-g GID

--gid GID

Establece el GID del nuevo grupo.

-r

--system

Crea un grupo del sistema.

2.3.2 Comando groupmod

Modifica la información de un grupo.

Sintaxis

```
groupmod [opciones] grupo
```

Parámetros

grupo

El nombre del grupo que se modifica.

Opciones

-g GID

--gid GID

Establece el GID del nuevo grupo.

-n nuevo_nombre

--new-name nuevo_nombre

Establece el nuevo nombre del grupo

2.3.3 Comando groupdel

Elimina un grupo del sistema.

Sintaxis

```
groupdel grupo
```

Parámetros

grupo

El nombre del grupo que se borra.

2.3.4 Comando gpasswd

El comando gpasswd gestionar los grupos y las contraseñas de los grupos. Si se invoca sin opciones se establece la contraseña del grupo.

Sintaxis

```
gpasswd [opciones] grupo
```

Parámetros

grupo

El nombre del grupo que se gestiona.

Opciones

-a login

--add login

Añade el usuario al grupo.

-d login

--delete login

Borra al usuario del grupo.

-r

--remove-password

Elimina la contraseña del grupo

2.3.5 Comando newgrp

Este comando cambia el grupo por defecto del usuario.

Sintaxis

```
newgrp [-] grupo
```

Parámetros

grupo

El nombre del grupo al que se une el usuario.

Opciones

–

Reinicializa el entorno de usuario como si hubiera abierto sesión con el nuevo grupo asignado.

Si lo invoca un usuario estándar tendrá que introducir la contraseña del grupo para poder unirse al mismo. Si el usuario figura como posible miembro del grupo en el fichero `/etc/gshadow` no se le pedirá contraseña.

2.4 El archivo `/etc/sudoers`

La separación de privilegios es uno de los paradigmas de seguridad fundamentales en un sistema operativo. Los usuarios regulares operan con privilegios limitados para reducir el ámbito de su influencia a su propio entorno y no la totalidad del sistema operativo.

Por defecto, el superusuario (`root`) está deshabilitado, y se recomienda que siga así por motivos de seguridad. Para realizar tareas administrativas tenemos a los miembros del grupo `sudo`. El sistema está configurado para que los miembros de este grupo puedan ejecutar cualquier comando con `sudo` como si fueran el usuario `root`. Por tanto, simplemente añadiendo a los usuarios que se necesiten como miembros del grupo `sudo` podemos establecer qué usuarios puedan realizar tareas administrativas.

Sin embargo, hay situaciones y escenarios en los que necesitamos que usuarios no administradores realicen algún tipo de tarea administrativa, pero no que puedan realizar cualquier tarea administrativa. Para controlar los privilegios que podemos asignar a los usuarios tenemos el archivo `/etc/sudoers`.

El archivo `/etc/sudoers` contiene reglas que los usuarios están obligados a seguir cuando ejecutan comandos con `sudo`. Este archivo no se edita directamente, sino con el comando `sudo visudo`. Al ejecutarlo veremos el contenido del archivo `/etc/sudoers` y podremos editarlo.

```
Defaults env_reset
Defaults mail_badpass
Defaults secure_path =
"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
...
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
...
```

Vamos a ver la configuración establecida en el fichero `sudoers` por defecto.

La primera línea, `Defaults env_reset` elimina cualquier variable de usuario creado con `sudo`. Esto es una medida de seguridad para limpiar cualquier variable de entorno potencialmente dañina desde una sesión con `sudo`.

La segunda línea, `Defaults mail_badpass`, le dice al sistema que realice cualquier notificación de intentos fallidos de `sudo` por clave errónea mediante el envío de mensajes al usuario configurado como destino de mensajes de aviso, `root` por defecto.

La tercera línea, `Defaults secure_path="..."`, especifica el path (directorios en los que el sistema operativo busca aplicaciones) que usará para operaciones `sudo`. Esto evita que el usuario emplea otras localizaciones con aplicaciones dañinas.

En el fragmento del archivo anterior vemos las reglas establecidas para el usuario `root` y los grupos `admin` y `sudo`. La línea del usuario `root` establece que puede ejecutar desde todos (ALL) los terminales, actuando como cualquier (ALL) usuario y ejecutando cualquier (ALL) comando. El formato de línea es la siguiente:

```
{usuario | %grupo} hosts=(usuario:grupo) comando
```

- ✓ Nombre de usuario o grupo. Si se especifica un grupo, el nombre del grupo se precede con %.
- ✓ Hosts → Indica desde donde se pueden ejecutar comandos, es decir, a qué hosts se aplica. Con ALL se indica desde todos los hosts.
- ✓ Usuario → Indica cómo quién se puede actuar. Con ALL se actúa como cualquier usuario. Si se omite por defecto es `root`.
- ✓ Grupo → Indica como qué grupo se puede actuar. Con ALL se actúa como cualquier grupo. Si se omite por defecto es `root`.
- ✓ Comando → Indica que comandos se pueden ejecutar.

Por ejemplo, con la siguiente línea establecemos que el `usuario2` puede ejecutar comandos para añadir nuevos usuarios actuando como `root` y grupo `root`.

```
usuario2 ALL = /usr/sbin/useradd *
```

El comando se debe especificar con ruta completa.

Es posible que queramos indicar la misma regla anterior para un conjunto de usuarios. Una forma de hacerlo es mediante un alias. Podemos definir un alias con una lista de usuarios separados por coma y posteriormente usar el alias para establecer la regla. Por ejemplo

```
User_Alias INSTALADORES = usuario2, usuario3, usuario4  
INSTALADORES ALL=(ALL:ALL) /usr/sbin/apt-get install *
```

Primero hemos definido un alias llamado `INSTALADORES` que lo forman los usuarios `usuario2`, `usuario3` y `usuario4`. Posteriormente empleamos el alias definido para indicar que pueden ejecutar el comando `apt-get install` desde cualquier host y

actuando como cualquier usuario o grupo. El * final en el comando es para poder ejecutarlo con argumentos.

3 Administración de contraseñas

En este epígrafe vamos a ver como realizar una adecuada gestión de las contraseñas. Primero veremos como los usuarios pueden cambiar su contraseña y como el administrador puede asignar contraseñas a los usuarios. Además, gestionaremos la política de contraseñas para que los usuarios estén obligados a emplear contraseñas seguras bajo unos criterios de complejidad.

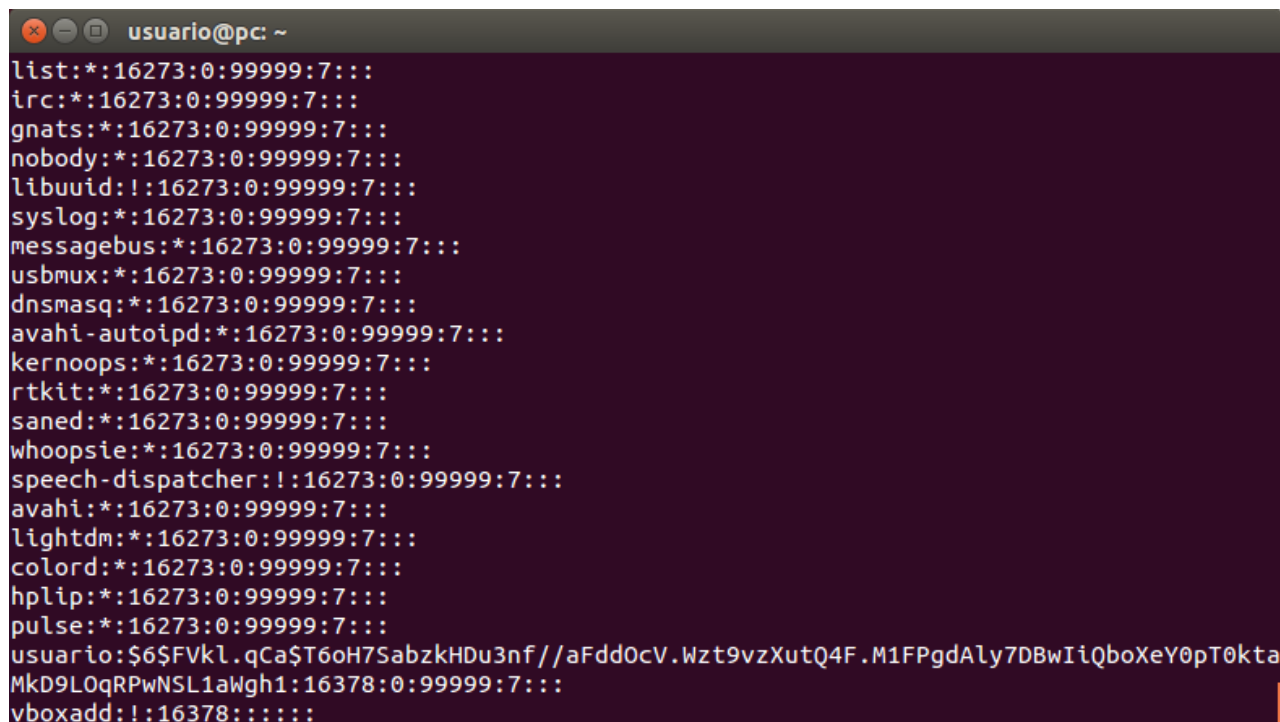
3.1 El archivo `/etc/shadow`

Hace mucho tiempo las contraseñas se guardaban cifradas en el archivo `/etc/passwd`. Este archivo puede ser leído por cualquier usuario del sistema, pero sólo puede ser modificado por el usuario `root`. Sin embargo, esto representa un problema porque significa que cualquiera puede ver una contraseña cifrada. Con las potentes máquinas actuales es más fácil descifrar una contraseña, incluso de las más complejas.

El lector podría pensar que estableciendo los permisos de este archivo para que solamente el usuario `root` pueda leerlo se solucionaría el problema. Sin embargo no es así, ya que es habitual que las aplicaciones tengan que leer la información de este archivo, por tanto tiene que tener permiso de lectura para todos los usuarios.

Para combatir esta amenaza potencial, se definió el concepto de contraseñas ocultas. Consiste en almacenar las contraseñas en otro archivo, el archivo `/etc/shadow`, mientras que el archivo `/etc/passwd` almacena la información de usuarios y puede ser leído por cualquiera. El archivo de contraseñas ocultas `/etc/shadow` sólo puede ser leído y actualizado por el usuario `root`.

El archivo `/etc/shadow` también es de texto ASCII y tiene el siguiente aspecto



```

usuario@pc: ~
list:*:16273:0:99999:7:::
irc:*:16273:0:99999:7:::
gnats:*:16273:0:99999:7:::
nobody:*:16273:0:99999:7:::
libuuid!:16273:0:99999:7:::
syslog:*:16273:0:99999:7:::
messagebus:*:16273:0:99999:7:::
usbmux:*:16273:0:99999:7:::
dnsmasq:*:16273:0:99999:7:::
avahi-autoipd:*:16273:0:99999:7:::
kernoops:*:16273:0:99999:7:::
rtkit:*:16273:0:99999:7:::
saned:*:16273:0:99999:7:::
whoopsie:*:16273:0:99999:7:::
speech-dispatcher!:16273:0:99999:7:::
avahi:*:16273:0:99999:7:::
lightdm:*:16273:0:99999:7:::
colord:*:16273:0:99999:7:::
hplip:*:16273:0:99999:7:::
pulse:*:16273:0:99999:7:::
usuario:$6$FVkl.qCa$T6oH7SabzkHDu3nf//aFdd0cV.Wzt9vzXutQ4F.M1FPgdAly7DBwIiQboXeY0pT0kta
MkD9LOqRPwNSL1aWgh1:16378:0:99999:7:::
vboxadd!:16378:0:99999:7:::

```

Figura 8.- Archivo /etc/shadow

De forma semejante al archivo `/etc/passwd`, el archivo `/etc/shadow` tiene una línea por cada contraseña de usuario. Cada línea tiene una serie de campos separados por puntos (:). Estos campos son:

- ✓ Login.- Nombre de la cuenta de usuario.
- ✓ Contraseña.- Cifrada. Si aparece un ! significa que la cuenta está deshabilitada. Si está vacío significa que no tiene contraseña y una contraseña * hace referencia a una cuenta que no puede abrir sesión.
- ✓ Fecha del último cambio de clave.- Este dato está expresado en número de días desde el 1 de Enero de 1970.
- ✓ Vigencia mínima de la clave.- Número de días que el usuario tiene que esperar antes de poder cambiar su clave. Si está vacío o es 0 significa que no hay vigencia mínima.
- ✓ Vigencia máxima de la clave.- Número de días después de los cuales el usuario está obligado a cambiar la clave.
- ✓ Periodo de aviso.- Número de días que el sistema avisa al usuario que tiene que cambiar su contraseña antes de que expire.
- ✓ Periodo de inactividad.- Número de días, después de expirar la contraseña, que el usuario puede abrir sesión solo para cambiarla. Si está vacío significa que no hay periodo de inactividad.
- ✓ Fecha de expiración de la cuenta.- Fecha en que la cuenta expira expresado en número de días desde el 1 de Enero de 1970.

- ✓ Reservado.- Campo sin uso para futuras aplicaciones.

Centrémonos ahora en el campo de la contraseña. Un ejemplo podría ser el siguiente.

```
$6$oH4ZzMfP$L.yYo2PwUjqTo7.KbvmnEkoRBg02yR8QlIGsHeASFCRdK9PI
YvePBKbByvSxd6cLYWDSCNLEcxrSgJiIGrPD8.
```

El formato del campo contraseña es `idsa1t$función-hash`. El `id` es la función hash que se emplea para cifrar la contraseña. Puede tener uno de los siguientes valores:

- ✓ `1` es MD5
- ✓ `$2a$` es Blowfish
- ✓ `$2y$` es Blowfish
- ✓ `5` es SHA-256
- ✓ `6` es SHA-512

En el ejemplo aparece 6 indicando que se emplea la función SHA512, la cual produce una clave cifrada de 512 bits.

El `sa1t` es un valor generado aleatoriamente y utilizado para unirlo a la contraseña de usuario. Posteriormente, el conjunto se cifra con la función hash. De esta forma, el resultado del cifrado no es solo la contraseña del usuario, sino la misma unidad al `sa1t` generado. Así, se complica el descifrado de la clave.

Finalmente, el último valor es el resultado de cifrar el `sa1t` y la contraseña del usuario.

Con el comando `mkpasswd` podemos obtener una contraseña cifrada. Este comando pueden emplearlo las aplicaciones para verificar el hash de una contraseña.

Si tenemos un usuario con la siguiente información de contraseña.

```
usuario@PC00:~$ sudo cat /etc/shadow | grep usuario:
usuario:
$6$oH4ZzMfP$L.yYo2PwUjqTo7.KbvmnEkoRBg02yR8QlIGsHeASFCRdK9PI
YvePBKbByvSxd6cLYWDSCNLEcxrSgJiIGrPD8.:18228:0:99999:7:::
```

Podemos ejecutar `mkpasswd` y emplear el mismo salto para verificar la contraseña

```
usuario@PC00:~$ mkpasswd --method=sha-512 --sa1t=oH4ZzMfP
usuario
$6$oH4ZzMfP$L.yYo2PwUjqTo7.KbvmnEkoRBg02yR8QlIGsHeASFCRdK9PI
YvePBKbByvSxd6cLYWDSCNLEcxrSgJiIGrPD8.
```

3.2 El archivo `/etc/gshadow`

Las contraseñas de grupo también se almacenan en un archivo aparte que solamente puede ser leído por el usuario `root`. ¿Para qué sirve una contraseña de grupo? Cuando un usuario abre sesión tiene que autenticarse con su contraseña, pero aparentemente las

contraseñas de grupo no sirven para autenticación. Esta contraseña es la que un usuario tiene que introducir si quiere unirse al grupo.

El archivo también es de texto ASCII y el formato es similar al del archivo `/etc/shadow` y cada línea representa la contraseña de un grupo con los siguientes campos:

- ✓ Nombre del grupo.
- ✓ Contraseña cifrada.
- ✓ Administradores.- Lista de usuarios separados por comas que pueden cambiar la contraseña de grupo.
- ✓ Miembros.- Lista de usuarios separados por comas que pueden unirse al grupo sin necesidad de contraseña.

3.3 Establecer la contraseña de usuario. Comando `passwd`

El comando `passwd` permite cambiar la contraseña de un usuario. Si es invocado por un usuario estándar solamente podrá cambiar su propia contraseña. El usuario `root` puede cambiar cualquier contraseña. Este comando también se emplea para cambiar información sobre la contraseña de usuario.

Cuando se invoca solicita primero por teclado la contraseña actual para verificar que el usuario está cambiando su propia contraseña. Si lo ejecuta el usuario `root` para cambiar la contraseña de otro usuario no pedirá la actual. Posteriormente solicitará la nueva contraseña dos veces para verificar que se ha escrito correctamente. Si ambas coinciden la nueva contraseña se almacenará en el fichero de contraseñas y estará en vigor a partir del siguiente inicio de sesión del usuario.

Sintaxis

```
passwd [opciones] [login]
```

Parámetros

`login`

El login del usuario al que se cambia la contraseña. Si se omite se cambia la contraseña del usuario que ejecuta `passwd`.

Opciones

`-d`

`--delete`

Elimina la contraseña de usuario para que puede abrir sesión sin una clave

`-e`

`--expire`

Expira inmediatamente la contraseña actual para forzar al usuario a cambiarla en el siguiente inicio de sesión.

-i días
--inactive días

Esta opción se usa para deshabilitar la cuenta del usuario cuando su contraseña ha expirado después de los días especificados.

-l
--lock

Bloquea la contraseña para que el usuario no pueda abrir sesión.

-n días
--mindays días

Establece el número mínimo de días entre cambios de contraseña. Un valor 0 indica que el usuario puede cambiar su contraseña cuando quiera.

-u
--unlock

Desbloquea la contraseña para que el usuario pueda volver a abrir sesión.

-w días
--warndays días

Establece el número de días de aviso para cambiar la contraseña.

-x días
--maxdays días

Establece el número máximo de días que la contraseña permanece sin cambiarse.

-S
--status

Presenta información en pantalla del estado de la cuenta.

3.4 Políticas de contraseña

Una política de contraseña es un conjunto de directrices de obligado cumplimiento que definen los requisitos que tiene que cumplir una contraseña de usuario para ser calificada como segura. Estos requisitos varían de una organización a otra, pero generalmente pertenecen a alguna de las siguientes categorías:

- ✓ La complejidad de la contraseña → Aquí se describen unos criterios para formar contraseñas complejas y, por tanto, más difíciles de averiguar.
- ✓ Vigencia de la contraseña → El periodo de tiempo que la contraseña es válida, al término del cual el usuario está obligado a cambiarla.
- ✓ Bloqueo de cuenta → Aquí indicamos límite de intentos de los usuarios para abrir sesión y periodo de tiempo del bloqueo de la cuenta si se supera el número máximo de intentos.

En los siguientes apartados vamos a abordar cada uno de los tres aspectos anteriores para definir una política de contraseña.

3.4.1 Complejidad de la contraseña. El archivo `/etc/security/pwquality.conf`

Para definir una complejidad de la contraseña Ubuntu emplea la librería

`libpam_pwquality`. Los requisitos para establecer la complejidad de las contraseñas de los usuarios podemos modificarla editando el archivo `/etc/security/pwquality.conf`.

Los requisitos de la contraseña que definimos en este archivo se aplica cuando el usuario emplea la herramienta gráfica para cambiar su contraseña, o el administrador para establecer la contraseña de cualquier usuario. Este archivo de texto puede incluir un conjunto de directivas las cuales modifican algún aspecto de la complejidad de las contraseñas. El archivo contiene únicamente líneas de comentario que explican las diferentes opciones que podemos utilizar y cuáles son sus valores por defecto. Estas son:

- ✓ `difok` → Número de caracteres en la nueva contraseña que no pueden estar la anterior contraseña. Por defecto son 5.
- ✓ `minlen` → Longitud mínima de la contraseña (más uno si los créditos no están desactivados, lo que es por defecto. Por defecto es 9 y el valor no se puede establecer por debajo de 6.
- ✓ `dcredit` → Créditos máximos por tener dígitos en la nueva contraseña. Si es negativo sería el número mínimo de dígitos en la nueva contraseña. Por defecto es 1.
- ✓ `ucredit` → Créditos máximos por tener letras mayúsculas en la nueva contraseña. Si es negativo sería el número mínimo de letras mayúsculas en la nueva contraseña. Por defecto es 1.
- ✓ `lcredit` → Créditos máximos por tener letras minúsculas en la nueva contraseña. Si es negativo sería el número mínimo de letras minúsculas en la nueva contraseña. Por defecto es 1.
- ✓ `ocredit` → Créditos máximos por tener otros caracteres (`.;#@...`) en la nueva contraseña. Si es negativo sería el número mínimo de otros caracteres en la nueva contraseña. Por defecto es 1.
- ✓ `minclass` → Número mínimo de tipos de caracteres requeridos por la nueva contraseña (dígitos numéricos, letras mayúsculas y minúsculas, y otros). Por defecto es 0.
- ✓ `maxrepeat` → Número máximo de caracteres consecutivos iguales. Por defecto está deshabilitado con 0.
- ✓ `maxclassrepeat` → Número máximo de caracteres consecutivos permitidos de la misma clase. Por defecto está deshabilitado con 0.
- ✓ `gecoscheck` → Un valor distinto de cero comprueba si las palabras más largas de 3 caracteres del campo GECOS (nombre completo del usuario) están en la nueva contraseña. La comprobación está deshabilitada con el valor por defecto de 0.
- ✓ `badwords` → Lista de palabras separadas por espacio que no debe contener la nueva contraseña. Estas palabras son adicionales a la comprobación del diccionario `cracklib`. Esta directiva también se puede ser empleada por aplicaciones que emulan la comprobación GECOS para cuentas de usuario que no se han creado todavía.

✓ `dictpath` → Path al direccionario cracklib.

A la hora de configurar las directivas anteriores para establecer una complejidad de la contraseña, debemos tener en cuenta que el cambio de contraseña desde Configuración → Detalles → Usuarios emplea el valor de la directiva `minlen` como la longitud mínima de la contraseña, pero sin emplear el sistema de créditos. Por tanto, los valores positivos en las directivas `dcredit`, `ucredit`, `lcredit` y `ocredit` no se tienen en cuenta. Sin embargo, si los valores de estas directivas son negativos, si los utiliza para forzar al usuario a introducir una cantidad mínima de caracteres del tipo indicado por la directiva en cuestión.

De lo anterior se deduce que si, por ejemplo, si queremos que los usuarios utilicen contraseñas de, como mínimo, 10 caracteres de longitud, con 1 carácter de cada clase tendríamos que establecer la siguientes directivas.

```
minlen=10
dcredit=-1
ucredit=-1
lcredit=-1
ocredit=-1
```

El sistema de créditos se emplea con la librería `pam_pwquality` que se explica a continuación.

3.4.2 Complejidad de la contraseña. Librería `pam_pwquality`

Cuando los usuarios utilizan el comando `passwd` para cambiar su contraseña, Ubuntu permite utilizar cualquier contraseña ya que no se aplica las directivas establecidas en el archivo `/etc/security/pwquality.conf`. Por tanto, en un sistema operativo sin GUI se hace necesario una política de contraseñas impida que los usuarios utilicen contraseñas débiles que harían el sistema vulnerable.

Tradicionalmente, las distribuciones GNU/Linux utilizaban, y puede que aún utilicen, la librería `pam_cracklib`, que permite establecer unas directrices para la complejidad de la contraseña parecidas a las vistas en el apartado anterior. Sin embargo, poco a poco se está migrando al emplea de la librería `pam_pwquality`, la cual añade algunas mejoras.

En esta situación, si queremos aplicar una complejidad de la contraseña debemos instalar la librería `libpam_pwquality`, la cual se encargará de que cada vez que un usuario cambia su contraseña verificar que la nueva cumple unos requisitos de complejidad, como que la nueva contraseña no es igual a una anteriormente utilizada, o si ha sido reutilizada cambiando únicamente un carácter. Otras comprobaciones que realiza es por ejemplo si es demasiado corta, o poco compleja, además, también se asegura que no tiene un único carácter introducido varias veces de forma consecutiva, etc.

Para instalar esta librería ejecutamos el siguiente comando:

```
sudo apt-get install libpam-pwquality
```

Antes de realizar cambios en la configuración vamos a realizar una copia de seguridad del fichero de configuración, esto no es obligatorio pero sí recomendable. Una vez hecho,

podemos editarlo.

```
sudo cp /etc/pam.d/common-password /root/
```

Cuando editemos el archivo veremos una línea similar a la siguiente

```
password requisite pam_pwquality.so retry=3 minlen=8 difok=3
```

Aquí es donde estableceremos la configuración de nuestra política de contraseñas. Disponemos de las siguientes opciones:

- ✓ `retry` → Número de intentos antes de que el sistema devuelva un error.
- ✓ `minlen` → Longitud mínima de contraseña. Más adelante se explica en detalle como se aplica este valor.
- ✓ `difok` → Número mínimo de caracteres que debe ser diferentes de la anterior contraseña.
- ✓ `ucredit` → Número de caracteres en mayúscula.
- ✓ `lcredit` → Número de caracteres en minúscula.
- ✓ `dcredit` → Número de dígitos numéricos.
- ✓ `ocredit` → Número de símbolos.

El significado de estas cuatro últimas opciones depende de si el valor es positivo o negativo. Más adelante se explica en detalle el sistema de créditos.

- ✓ `minclass` → El número mínimo de tipos de caracteres que se deben usar. Como tipos de caracteres nos referimos a letras mayúsculas, minúsculas, dígitos numéricos y otros caracteres.
- ✓ `maxrepeat` → El número máximo de veces que un carácter se puede repetir.
- ✓ `maxclassrepeat` → El número máximo de caracteres de la misma clase que pueden ir seguidos.
- ✓ `remember` → El número de contraseñas que el sistema recuerda y no pueden utilizarse de nuevo.
- ✓ `enforce_for_root` → Indica que el usuario `root` también está obligado a cumplir los requisitos de complejidad de la contraseña cuando cambia la contraseña de los usuarios.

Las opciones `ucredit`, `lcredit`, `dcredit` y `ocredit` pueden tener números negativos o positivos, y en cada caso tendrá un significado diferente. Si el número es positivo indica el número de créditos que se añade para alcanzar el valor de `minlen`. Si por el contrario, el número es negativo indica la cantidad mínima de caracteres de ese tipo, pero sin sumar créditos por ello. Por ejemplo si ponemos `ucredit=-3` significa que, como mínimo, la contraseña debe tener 3 caracteres en mayúscula, y no se añaden créditos a la

contraseña; pero si ponemos `ucredit=+3` significa que la contraseña sumará un crédito por cada carácter en mayúscula que incluya hasta un máximo de tres. Con el resto de opciones referente al número de caracteres de un tipo se opera de forma similar.

El parámetro `minlen` no es lo que parece, no indica la longitud de la contraseña, sino que establece la longitud mínima de la contraseña en créditos. Por tanto, `minlen` es en realidad una medida de la complejidad de la contraseña que se basa en el uso de créditos. La idea de asignar créditos a una contraseña es realmente interesante. Básicamente, los créditos de una contraseña es un valor que indica lo compleja que es dicha contraseña. Podría ocurrir que una contraseña más corta que otra fuera más compleja al tener más créditos.

Por ejemplo, una clave como `azbycdwev` podría satisfacer un parámetro `minlen=10` ya que tiene una longitud de 10 caracteres. Por otro lado, si `dcredit` se establece a 2 la clave `azbycx99` también sería aceptable, aunque solo tiene una longitud de 8 caracteres. Ello se debe a que se obtienen 2 créditos adicionales por los dígitos numéricos. Así, 8 caracteres más 2 dígitos numéricos se valora tanto como 10 caracteres. Si `dcredit` estuviera establecido a 1 los dígitos numéricos solamente suman 1 crédito y, consecuentemente, haría falta un carácter adicional. Sin embargo, podemos conseguir créditos por letras mayúsculas, minúsculas y caracteres no alfanuméricos.

Por tanto el parámetro `minlen` especifica el valor mínimo en créditos que una contraseña debe tener para ser aceptada. En una configuración donde no se exigieran forzosamente diferentes tipos de caracteres, cada carácter en una contraseña agrega un crédito a la cuenta de la complejidad de la misma. En esta situación `minlen` simplemente representaría la longitud de la contraseña pero, si algunos caracteres agregan más créditos, el cálculo es más complejo. El cálculo de los créditos de una contraseña sería así:

- ✓ Todos los caracteres de una contraseña tienen un crédito, sin importar el tipo de carácter.
- ✓ Cada letra minúscula suma un crédito, pero sólo hasta el valor positivo de `lcredit`.
- ✓ Cada letra mayúscula añade un crédito, pero sólo hasta el valor positivo de `ucredit`.
- ✓ Cada dígito numérico añade un crédito, pero sólo hasta el valor positivo de `dcredit`.
- ✓ Cada carácter no alfanumérico (como los símbolos de puntuación) añade un crédito, pero sólo hasta el valor positivo de `ocredit`.

Se deduce que si `lcredit`, `ucredit`, `dcredit` y `ocredit` se ajustan a 0, sólo la longitud de la contraseña se utiliza para determinar si es aceptable.

Una vez que sabemos las opciones que disponemos y lo que significan cada una de ellas, debemos elegir la configuración para indicar la complejidad de las contraseñas. Un ejemplo de política de contraseñas seguras sería la siguiente:

```
password requisite pam_pwquality.so retry=3 minlen=10
difok=3 ucredit=1 lcredit=1 dcredit=2 ocredit=1
enforce_for_root
```

Con la política anterior tendremos:

- ✓ Longitud mínima de la contraseña 10 créditos (parámetro `minlen=10`).
- ✓ Si cambiamos la clave, como mínimo debe ser diferente a las tres claves anteriores (parámetro `difok=3`).
- ✓ 1 crédito adicional por los caracteres en mayúscula (`ucredit=1`) sin estar obligado a introducir ningún carácter de este tipo.
- ✓ 1 crédito adicional por los caracteres en minúscula (`lcredit=1`) sin estar obligado a introducir ningún carácter de este tipo.
- ✓ 2 créditos adicionales por los dígitos numéricos (`dcredit=2`) sin estar obligado a introducir ningún carácter de este tipo.
- ✓ 1 crédito adicional por los caracteres no alfanuméricos (`ocredit=1`) sin estar obligado a introducir ningún carácter de este tipo.
- ✓ El usuario `root` está obligado a cumplir los requisitos de la contraseña.

En función de lo anterior vamos a calcular los créditos para la siguiente contraseña: `a9Z1%`.

Parámetro	Longitud	Car. May.	Car. Min.	Dígitos Num.	Otros
Valor	5	1	1	2	1
Créditos	5	1	1	2	1

La contraseña incluye el número de cada tipo de caracteres especificados en las directivas que suman créditos. Si sumamos todos los créditos tenemos en total 10 y, por tanto, la clave es aceptada ya que supera el valor de `minlen`, aunque su longitud solamente es de 5 caracteres.

Con esta configuración el usuario no está obligado a introducir caracteres de ninguno de los tipos descritos. Podría introducir solamente caracteres en minúscula o mayúscula, dígitos numéricos, otros caracteres, o una combinación cualquiera de ellos. Sin embargo, la contraseña introducida deberá alcanzar 10 créditos o será rechazada. Por ejemplo, las siguientes contraseñas serían aceptables:

```
aqswdefrg
aqswdef1
aqswdefrA
aQs12w
```

Por otro lado, si queremos obligar a los usuarios a que introduzcan caracteres de algún tipo tendremos que definir valores negativos para las directivas anteriores y, en este caso, no suman créditos. Por ejemplo, si modificamos la configuración anterior y establecemos los

misimos valores, pero en negativo.

```
password requisite pam_pwquality.so retry=3 minlen=10
difok=3 ucredit=-1 lcredit=-1 dcredit=-2 ocredit=-1
enforce_for_root
```

Ahora, el usuario está obligado a introducir una contraseña en la que haya una letra mayúscula como mínimo, una en minúscula, dos dígitos numéricos y un carácter de otro tipo. Si embargo, los caracteres que introduzca de estos tipos no añaden créditos y, por tanto, tendrá que tener si o si una longitud de 10 caracteres.

Comprobar la longitud en créditos de una contraseña y el uso de los diferentes tipos de caracteres no son las únicas comprobaciones que la librería realiza. También comprueba lo siguiente:

- ✓ Que no sea una palabra del diccionario.
- ✓ Que no sea un palíndromo.
- ✓ Que no sea una antigua en la que se ha cambiado solo un carácter.
- ✓ Que no sea igual a una antigua en la que han cambiado algunos caracteres. Esto lo controla el parámetro `difok`.
- ✓ Que no sea igual a una antigua. Esto se controla con el parámetro `remember`.
- ✓ Que no sea demasiado simple. Esto lo controlan los parámetros `ucredit`, `lcredit`, `dcredit` y `ocredit`. Además no se permiten introducir caracteres seguidos en orden como `ab` o `45`.
- ✓ Que no sea una contraseña antigua en la que hemos rotado los caracteres.
- ✓ Que no contenga el nombre del usuario.

Una vez que hayamos configurado el fichero de texto, lo guardamos y ya podremos comprobarlo cambiando las contraseñas de los usuarios.

3.4.3 Bloqueo de cuenta

Además de obligar a los usuarios a que utilicen contraseñas complejas debemos establecer, dentro de nuestra política de contraseñas, cuántos intentos tienen para abrir sesión y cuánto tiempo estará bloqueada una cuenta de usuario cuando haya superado el número de intentos permitidos.

La librería `pam_tally2` está deprecated (obsoleta) y ya no está disponible en Ubuntu partir de la versión 22.04, en su lugar está la librería `pam_faillock`.

La configuración de esta librería permitirá el bloqueo de una cuenta de usuario cuando se haya realizado un número de intentos fallidos de conexión al sistema.

Para configurar esta librería será necesario modificar los ficheros

/etc/pam.d/common-auth

```
GNU nano 6.2 /etc/pam.d/common-auth *
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.
#
# here are the per-package modules (the "Primary" block)
auth required pam_faillock.so deny=3 unlock_time=60
auth required pam_faillock.so preauth audit silent deny=5 unlock_time=120

auth [success=2 default=ignore] pam_unix.so nullok
auth [success=1 default=ignore] pam_sss.so use_first_pass

#BEGIN ANSIBLE MANAGED BLOCK
auth [default=die] pam_faillock.so authfail audit deny=5 unlock_time=120
auth sufficient pam_faillock.so authsucc audit deny=5 unlock_time=120
#END ANSIBLE MANAGED BLOCK
```

/etc/pam.d/common-account

En este fichero al final del mismo se añadirá la línea

account required pam_faillock.so

```
GNU nano 6.2 /etc/pam.d/common-account
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.
#
# here are the per-package modules (the "Primary" block)
account [success=1 new_authtok_reqd=done default=ignore] pam_unix.so
# here's the fallback if no module succeeds
account requisite pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
account required pam_permit.so
# and here are more per-package modules (the "Additional" block)
account sufficient pam_localuser.so
account [default=bad success=ok user_unknown=ignore] pam_sss.so
# end of pam-auth-update config
account required pam_faillock.so
```

El parámetro `deny=N` indica el número máximo de intentos que un usuario tiene para autenticarse antes de que su cuenta quede bloqueada. No afecta al usuario `root`. En el caso de que también queramos aplicarlo a `root` tendríamos que añadir el parámetro `even_deny_root`.

El parámetro `unlock_time=N` indica el número de segundos que transcurrirán desde el bloque de una cuenta hasta que sea desbloqueada por el sistema

automáticamente. Si no se especifica este parámetro, la cuenta solamente podrá ser desbloqueada manualmente por el administrador.

El parámetro `silent` indica que no se informará al usuario que ha sido bloqueado, se mostrará error de autenticación.

Guardamos el archivo y realizamos nuestras pruebas para verificar su correcto funcionamiento.

```
usuario@pc1:~$ su user01
Contraseña:
su: Permiso denegado
usuario@pc1:~$ su user01
Contraseña:
su: Permiso denegado
usuario@pc1:~$ su user01
Contraseña:
su: Permiso denegado
usuario@pc1:~$ su user01
Contraseña:
su: Permiso denegado
usuario@pc1:~$ su user01
Contraseña:
su: Permiso denegado
usuario@pc1:~$ su user01
Contraseña:
su: Fallo de autenticación
usuario@pc1:~$
```

A los 5 intentos bloquea el usuario, pero no informa del bloqueo, indica fallo de autenticación. Posteriormente aunque se introduzca contraseña correcta el usuario no podrá acceder y deberá esperar el tiempo de bloqueo o pedir que se reinicie el número de errores para desbloquear al usuario.

Cuando necesitemos ver el número de intentos fallidos de un usuario ejecutaremos el comando `faillock` de la siguiente forma.

Si se ejecuta el comando sin opciones muestra la información de intentos fallidos de todos los usuarios

```
usuario@pc1:~$ sudo faillock
user01:
When                Type  Source                Valid
2023-04-13 18:59:24 TTY   /dev/pts/0           V
2023-04-13 18:59:51 TTY   /dev/pts/0           V
usuario:
When                Type  Source                Valid
usuario@pc1:~$
```

Nos mostrará el número de intentos errados por el usuario. Si no indicamos la opción `--user` mostrará todos los usuarios con intentos fallidos.

```
usuario@pc1:~$ su user01
Contraseña:
su: Permiso denegado
usuario@pc1:~$ sudo faillock --user user01
user01:
When                Type  Source                Valid
2023-04-13 18:59:24 TTY   /dev/pts/0           V
2023-04-13 18:59:51 TTY   /dev/pts/0           V
usuario@pc1:~$
```

Para desbloquear a un usuario manualmente y no tener que esperar al desbloqueo automático ejecutaremos el siguiente comando.

```
usuario@pc1:~$ sudo faillock --user user01
user01:
When                Type  Source                Valid
2023-04-13 18:59:24 TTY   /dev/pts/0           V
2023-04-13 18:59:51 TTY   /dev/pts/0           V
usuario@pc1:~$ sudo faillock --user user01 --reset
usuario@pc1:~$ sudo faillock --user user01
user01:
When                Type  Source                Valid
usuario@pc1:~$
```

El número de intentos fallidos actuales los ha puesto a cero y el usuario puede volver a abrir sesión si estuviera bloqueado.

3.4.4 Vigencia de la contraseña

Otro aspecto a tener en cuenta a la hora de definir una política de contraseñas es su validez. Para establecer un período de vigencia para las contraseñas, debemos editar las siguientes opciones en el archivo `/etc/login.defs`.

PASS_MAX_DAYS	99999
PASS_MIN_DAYS	0
PASS_WARN_AGE	7

Por ejemplo, si establecemos los siguientes valores:

PASS_MAX_DAYS	100
PASS_MIN_DAYS	0
PASS_WARN_AGE	10

Esta directiva obligará a los usuarios a cambiar la contraseña cada 100 días, y enviará un mensaje 10 días antes del vencimiento de la misma.

También podemos establecer configuraciones personalizadas por usuarios. Mediante el comando `chage` podemos establecer la caducidad de las contraseñas a un usuario

especifico.

El comando `chage` cambia el número de días entre cambios de contraseña y la fecha del último cambio de contraseña. Esta información es usada por el sistema para determinar cuando un usuario debe cambiar su contraseña.

Sintaxis

```
chage [opciones] [login]
```

Parámetros

login

El login de la cuenta de usuario

Opciones

`-d, --lastday último_día`

Establece el día del último cambio de la contraseña. La fecha se puede expresar en formato YYYY-MM-DD o MM/DD/YYYY.

`-E, --expiredate fecha_expiración`

Establece la fecha de caducidad de la contraseña. La fecha se puede expresar en formato YYYY-MM-DD o el formato utilizado en cada región. El administrador del sistema debe desbloquear la cuenta para que el usuario pueda volver a abrir sesión.

Pasar el número `-1` como FECHA_EXPIRACIÓN hará que quite en la cuenta la fecha de expiración.

`-I, --inactive inactivo`

Deshabilita la cuenta después de INACTIVA días de la fecha de caducidad

Pasar el número `-1` como INACTIVO quitará la inactividad de la cuenta.

`-l, --list`

Muestra información de vigencia de la cuenta.

`-m, --mindays días_mínimo`

Establece el número mínimo de días entre cambios de contraseña. Un valor 0 indica que el usuario puede cambiar su clave cuando quiera.

`-M, --maxdays días_máximo`

Establece el número máximo de días de vigencia de la contraseña. Cuando DIAS_MÍNIMOS más ÚLTIMO_DÍA es inferior a la fecha actual, se le pedirá al usuario que cambia su contraseña antes de poder usar su cuenta.

Pasar el número `-1` como DÍAS_MÁXIMO quitará la comprobación de validez de la contraseña.

`-W, --warndays días_aviso`

Establece el número de días de aviso antes de requerir el cambio de contraseña. El valor DIAS_AVISO es el número de días que el usuario será avisado de que tiene que cambiar su contraseña antes de que ésta expire.

Si no se indican acciones, `chage` opera de forma interactiva solicitando los valores para todas las opciones. Introduciendo un valor cambia la opción y dejando el valor en blanco seguirá con el valor en curso. El valor actual se muestra entre corchetes.

Por defecto se establece que la contraseña de un usuario nunca caduque. Para cambiar esto simplemente ejecutamos el siguiente comando:

```
chage -E 2/5/2019 -m 10 -M 80 -I 30 -W 14 usuario
```

- ✓ `-E 2/5/2019` → Establece que la contraseña expirara el 5/02/2019
- ✓ `-m 10 -M 80` → Numero mínimo (10) y máximo (80) de días para cambiar la contraseña.
- ✓ `-I 30` → La cuenta se bloqueará 30 días después de que expire la contraseña.
- ✓ `-W 14` → Envía el mensaje de advertencia 14 días antes de la expiración de la contraseña.

4 Dispositivos hardware y controladores

La mayoría de los controladores de hardware se instalan automáticamente con Ubuntu, lo que reduce la necesidad de instalar manualmente el soporte de hardware y controladores para los dispositivos del equipo.

Estos controladores son actualizados, junto con otros archivos del sistema, de manera regular, asegurándose de que los dispositivos están funcionando de manera óptima. Aquellos controladores que no se instalan automáticamente en general, caen bajo el paraguas de controladores manuales, ya que pueden necesitar que el usuario instale estos controladores para funcionar correctamente. Generalmente estos controladores suelen ser propietarios.

Un controlador propietario es un controlador de hardware que no es de libre disposición o no es de código abierto. La mayoría de los dispositivos conectados al equipo deberían funcionar correctamente en Ubuntu. Estos dispositivos probablemente dispongan de controladores libres, lo que significa que los desarrolladores de Ubuntu pueden modificarlos para corregir posibles problemas con ellos.

Algunos dispositivos hardware no tienen controladores libres, normalmente porque el fabricante del hardware no ha publicado los detalles de su hardware que permiten crear dicho controlador. Esos dispositivos pueden tener una funcionalidad limitada o bien no funcionar en absoluto.

Si un controlador propietario se encuentra disponible para un dispositivo, podemos instalarlo para permitir que el dispositivo funcione apropiadamente, o para añadir una nueva función. Por ejemplo, instalar un controlador propietario para una determinada tarjeta gráfica puede permitir usar efectos visuales más avanzados.

Algunos equipos pueden no necesitar usar controladores propietarios, debido a que todos los dispositivos están soportados por controladores libres o debido a que no existen

controladores propietarios disponibles para el dispositivo. Los controladores propietarios son mantenidos por el fabricante de hardware, y no pueden ser modificados por los desarrolladores de Ubuntu si hay un problema.

Ubuntu es de código abierto, por eso el código cerrado no se ha instalado o actualizado de forma automática. Sin embargo, instalar el soporte para estos controladores y actualizaciones de mantenimiento es un proceso sencillo para que su sistema funcione en perfecta armonía con Ubuntu.

4.1 Instalación de un controlador propietario

La instalación de un controlador propietario puede variar en función del fabricante. Algunos suministran un paquete binario para su instalación, otros un script del sistema operativo, etc.

En principio el controlador hay que descargarlo de la web del fabricante. Tendremos que elegir el modelo preciso de nuestro dispositivo y versión, ya que algunos fabricantes están desarrollando controladores para diferentes distribuciones Linux. Si el fabricante no ha desarrollado un controlador específico para Ubuntu, podemos utilizar el genérico de Linux que habrá con casi toda seguridad.

Una vez descargado se instala en el sistema. Si es un paquete binario `.deb` utilizaremos la herramienta `dpkg`. Si el fabricante suministra un repositorio de instalación por Internet y el nombre del paquete, tendremos que modificar la base de datos de repositorios de aplicaciones y utilizar alguna herramienta como `apt-get` o `synaptic` para su instalación. Por último si lo que proporciona el fabricante es un script del sistema operativo, lo descargamos y ejecutamos en una ventana de terminal invocándolo por el nombre del archivo, habiéndonos asegurado antes que nuestra shell es compatible con la del script.

LINUX DISPLAY DRIVER - X86

Versión: 367.44
Fecha de publicación: 2016.8.23
Sistema operativo: Linux 32-bit
Idioma: Español (España)
Tamaño: 42.04 MB

DESCARGAR AHORA

ASPECTOS DESTACADOS DE LA	PRODUCTOS SOPORTADOS	MÁS INFORMACION
<ul style="list-style-type: none">• Added support for the following GPUs: TITAN X (Pascal) GeForce GTX 1060 6GB GeForce GTX 1060 3GB• Fixed a regression that caused applications using indirect GLX to crash.• Fixed a regression introduced in 367.35 that caused the first modeset of the X server to display blank if the features requested in the X configuration file enabled the X driver's composition pipeline. This would be triggered, e.g., by MetaMode tokens such as ForceCompositionPipeline, ForceFullCompositionPipeline, Rotation, Reflection, and Transform.		

Figura 9.- Descarga de controladores Nvidia para Linux

En la imagen anterior se ilustra la descarga de un controlador de tarjeta gráfica Nvidia GeForce GTX 1060 desde la web del fabricante. El archivo descargado es un ejecutable para la instalación del controlador desde una ventana de terminal.

4.2 Gestión de controladores adicionales

Durante la instalación de Ubuntu es posible que se haya creado una lista de controladores propietarios a disposición del usuario para su instalación. En *Software y actualizaciones* de *Configuración del Sistema* tenemos la pestaña *Controladores adicionales* la cual muestra esta lista.

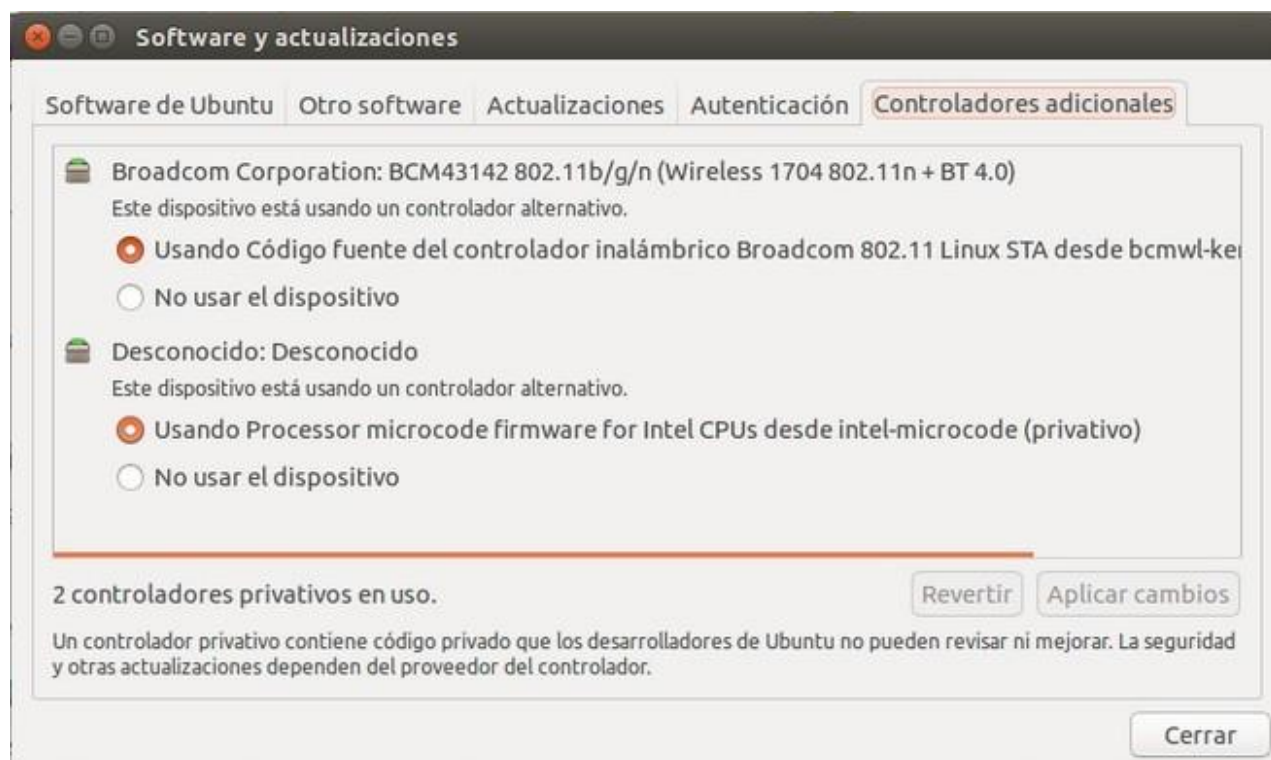


Figura 10.- Controlador Nvidia para una tarjeta de red inalámbrica Broadcom BCM43142

Aparecerá una lista de controladores. Alguno de ellos son propietarios y para utilizarlos tendremos que seleccionarlos y hacer clic en el botón *Aplicar cambios*. Al hacerlo se descargará de Internet el controlador y lo instalará. A partir de entonces se empleará el controlador para el dispositivo en lugar de utilizar el de Linux.

Si nuestro dispositivo no funciona correctamente con el controlador propietario podemos desactivarlo. Para ello se repiten los mismos pasos anteriores, pero ahora elegiremos otro controlador.

5 Instalación de aplicaciones

En Linux la mayoría de las aplicaciones son como el sistema operativo: de distribución gratuita y código abierto. Los desarrolladores de Linux están repartidos por todo el mundo creando aplicaciones de usuario que pondrán a disposición de los usuarios de forma gratuita y con el código fuente para hacer modificaciones, mejoras o nuevas versiones. Estas aplicaciones se distribuyen principalmente a través de Internet aunque también pueden distribuirse a través de soportes físicos como los CD's o los DVD's.

Si has sido usuario de Windows o Mac OS, estás probablemente acostumbrado a buscar los programas en Internet (la mayoría de veces ofrecidos en instaladores ejecutables) y teniendo que descargar e instalarlo. Estarás familiarizado probablemente con el software distribuido en CDs, DVDs, etc. los cuales normalmente tienen una característica autoejecutable desde donde puedes instalarlos. Para sistemas libres y abiertos como Ubuntu GNU/Linux hay algún software distribuido de este modo, pero la mayoría son programas propietarios y cerrados.

En Linux tenemos cuatro formas principales de instalar una aplicación:

- ✓ Mediante un ejecutable proporcionado por el distribuidor de la aplicación y descargado desde Internet.- No es muy habitual, pero algunos desarrolladores suministran aplicaciones que se instalan fácilmente ejecutan un script o programa de instalación de forma parecida a como se hace en Windows. Estas son las aplicaciones más fáciles de instalar.
- ✓ Compilando el código fuente de una aplicación.- La mayoría de las aplicaciones Linux son de código abierto, lo que significa que proporcionan el código fuente para que un usuario avanzado y con conocimientos de programación pueda modificarlo a su gusto o desarrollar a partir de ahí una mejora de sus funciones o funcionalidad añadida. El código fuente tiene que compilarse para obtener una versión ejecutable del programa. Esta forma es la más difícil y no al alcance de todos los usuarios.
- ✓ Instalando paquetes binarios.- Un paquete es una aplicación empaquetada junto con su documentación en uno o varios archivos y que se instala mediante herramientas específicas del sistema operativo. Generalmente los paquetes binarios se emplean para instalación de programas del sistema operativo.
- ✓ Mediante el centro de software.- Todas las distribuciones Linux incluyen una herramienta gráfica para instalar aplicaciones. Esta es la forma habitual de instalación de aplicaciones para usuarios en Linux.

5.1 Introducción a los paquetes binarios

En Linux las aplicaciones están empaquetadas, es decir, forman uno o varios paquetes. En sistemas como Ubuntu los paquetes vienen en la forma de ficheros con extensión `.deb` de Debian, ya que Ubuntu es una distribución Linux que deriva de Debian. Los paquetes en formato `.rpm` son propios de las distribuciones Red Hat y Fedora. Estos paquetes se descargan de Internet y se instalan con alguna herramienta propia del sistema operativo para ello.

Para instalar aplicaciones a través de paquetes binarios las distribuciones Linux disponen de varias herramientas para tal fin. Desde comandos en línea a herramientas gráficas.

5.1.1 Formato de un paquete en Debian

Un paquete Debian contiene los archivos ejecutables, bibliotecas y la documentación asociada con un programa particular o con un conjunto de programas relacionados. Normalmente, un paquete Debian tiene una extensión de archivo que termina en `.deb`. Los nombres de los paquetes Debian siguen la siguiente convención:

`nombreaplicacion_version-revision_architectura.deb`

donde generalmente `nombreaplicación` es el nombre del paquete o aplicación, `versión` es el nombre de versión, `revisión` es el número de revisión y `arquitectura` es la arquitectura. Por ejemplo el paquete `tcpdump_4.5.1-2ubuntu1.1_amd64.deb`

corresponde a la aplicación Tcpcdump (un sniffer de red), versión 4.5.1 revisión 2 para Ubuntu en la arquitectura AMD64 (64 bits).

El número de revisión es asignado por el desarrollador Debian o por quien creó el paquete. Un cambio en el número de revisión generalmente indica que ha cambiado algún aspecto del empaquetado.

5.1.2 Dependencias de paquetes

En Linux no siempre es necesario desarrollar una aplicación desde cero. Es posible que una aplicación pueda servir de punto de partida para ejecutar otra. Por ejemplo, si un desarrollador crea un navegador de Internet no es necesario programar todas las funciones de conexión de red ya que estas ya están instaladas en el sistema operativo para todas las aplicaciones que necesitan conectividad de red. Entonces, solamente se tiene que preocupar de centrarse en los detalles de su navegador y olvidarse del resto.

Para evitar duplicidades, la mayoría de las aplicaciones reutiliza la funcionalidad de otras aplicaciones o bibliotecas. Las bibliotecas sólo proporcionan funciones a otras bibliotecas o aplicaciones y no son aplicaciones por sí mismas. De esta manera, la mayoría de los paquetes dependen de otros paquetes.

Esto significa que una aplicación puede necesitar que exista otra instalada para poder funcionar y si la segunda no se encuentra instalada en el sistema, no se puede instalar la primera. A estas relaciones entre aplicaciones se las denomina dependencias entre paquetes y hay de varios tipos:

- ✓ El paquete A depende del paquete B si B debe instalarse para poder ejecutar A. En algunos casos A depende no sólo de B, sino de una versión específica de B. En este caso, la dependencia de versión constituye un límite inferior, es decir, A dependerá de cualquier versión de B más reciente que la versión especificada.
- ✓ El paquete A recomienda al paquete B si el encargado del mismo considera que la mayoría de los usuarios no querrán el paquete A sin tener también la funcionalidad proporcionada por B.
- ✓ El paquete A sugiere al paquete B si B contiene archivos que están relacionados y mejoran la funcionalidad de A. La misma relación se expresa diciendo que el paquete B mejora el paquete A.
- ✓ El paquete A está en conflicto con el paquete B cuando A no funciona si se instala B en el sistema. A menudo los “Conflictos” están relacionados con “Reemplazos”.
- ✓ El paquete A reemplaza el paquete B cuando los archivos instalados por B se eliminan o se sobrescriben por los archivos de A.
- ✓ El paquete A proporciona el paquete B cuando todos los archivos y funcionalidad de B están incorporados en A.

Generalmente los usuarios no tienen que preocuparse por las dependencias de paquetes a la hora de instalarlos ya que las herramientas de instalación de paquetes de alto

nivel se ocupan de ello. Sin embargo hay otras herramientas de instalación de paquetes que no realizan esta función.

Hay que tener en cuenta que si estamos instalando el paquete A y tiene dependencia de B, habrá que instalar el paquete B, pero éste a su vez tiene dependencia del paquete C, el cual no está instalado. Por tanto para instalar el paquete A hay que instalar previamente el paquete B y previamente a éste hay que instalar el paquete C. Así sucesivamente hasta llegar a un paquete que depende de paquetes que ya están instalados.

Cuando un usuario quiere instalar un paquete, la herramienta elegida crea un árbol de dependencias que consiste en un gráfico en forma de árbol en el que se representa las dependencias de unos paquetes con otros. Posteriormente revisa el árbol y comprueba que paquetes están instalados y cuáles no, de forma que instalará los que sean necesarios para poder instalar el paquete requerido.

Si por cualquier causa un paquete está instalado y otros de los que dependen no, entonces se dice que existen dependencias rotas y la aplicación instalada no funcionará adecuadamente. Las mismas herramientas de instalación de paquetes tienen capacidad para detectar dependencias rotas entre paquetes y resolver el problema.

5.1.3 Lista de repositorios

¿Dónde podemos conseguir aplicaciones Linux? Existen miles de aplicaciones para Linux y la mayoría de ellas son de código abierto, algunas incluso son específicas de algunas distribuciones Linux y también de un gestor de escritorio concreto. Es difícil conocerlas todas y saber para qué sirven por lo que se hace necesario tener alguna documentación que nos permita conocerlas un poco. Esta herramienta es el repositorio.

Los repositorios son servidores conectados en Internet que contienen paquetes para su instalación. Son mantenidos por los distribuidores del sistema operativo y colocan aquí las nuevas versiones de los paquetes para instalar todo el sistema operativo y las aplicaciones de usuarios. Existen dos tipos de repositorios:

- ✓ Los oficiales, que pertenece y mantiene el distribuidor.
- ✓ Los no oficiales, los cuales pertenecen a desarrolladores que no forman parte del distribuidor y generalmente se utilizan para instalar unas pocas aplicaciones concretas.

Por regla general, los repositorios son servidores ftp o http, aunque también pueden ser locales y estar disponibles en un dispositivo físico (DVD). Por todo el mundo hay repartidos servidores espejo, para no saturar los servidores principales y poder utilizar aquellos que estén más cerca de nuestra ubicación para una descarga más rápida.

En distribuciones Debian GNU/Linux y derivadas la lista de repositorios se almacena en el archivo `/etc/apt/sources.list`. Las diferentes herramientas para gestionar los paquetes toman la información de los mismos de los repositorios que se encuentran en este fichero.

De esta manera el archivo `sources.list` es una pieza importante en la

administración, por parte del usuario o administrador de sistemas, en Debian GNU/Linux y en las distribuciones derivadas. El archivo puede ser editado mediante un editor de texto, directamente desde una línea de comandos, para la modificación de la lista de repositorios, aunque es recomendable hacerlo mediante una herramienta gráfica que se verá en el siguiente apartado.

El formato de este archivo es muy simple. Cada repositorio ocupa una única línea. Las líneas que comienzan por # son comentarios. Solamente contiene texto aclaratorio sobre algún repositorio o también lo usamos para eliminar una línea de repositorio y poder recuperarla cuando necesitemos. Una línea de repositorio tiene el siguiente formato:

```
tipo uri distribución [componente1] [componente2] [...]
```

Cada campo tiene el siguiente significado:

tipo

Indica el tipo de repositorio. Puede ser:

- ✓ deb.- Es un repositorio de paquetes binarios.
- ✓ deb-src.- Es un repositorio de paquetes fuente.

uri

La URI (Uniform Resource Identifier), en este caso una localización de internet donde se pueden obtener los paquetes.

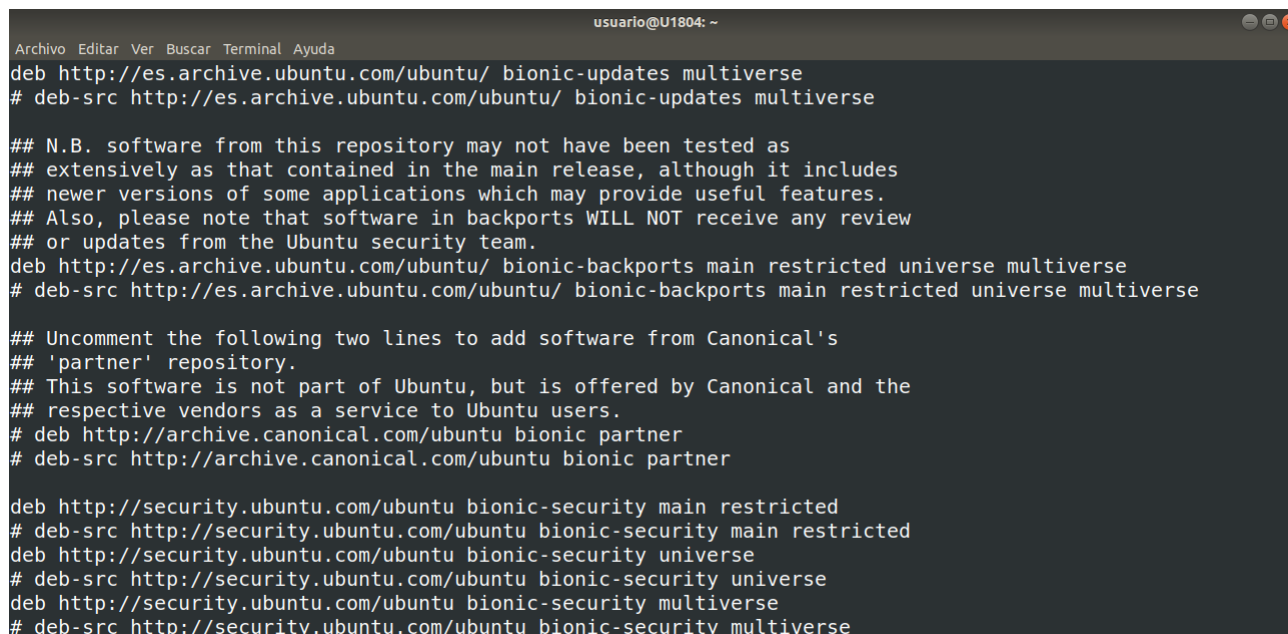
distribución

Es la versión de la distribución a la que pertenece el repositorio.

componente

Son los nombres de sección o componentes. Pueden haber varios nombres de secciones, separadas por espacios.

Por ejemplo, la siguiente imagen muestra el contenido del archivo `sources.list` en Ubuntu 20.04



```
usuario@U1804: ~
Archivo Editar Ver Buscar Terminal Ayuda
deb http://es.archive.ubuntu.com/ubuntu/ bionic-updates multiverse
# deb-src http://es.archive.ubuntu.com/ubuntu/ bionic-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://es.archive.ubuntu.com/ubuntu/ bionic-backports main restricted universe multiverse
# deb-src http://es.archive.ubuntu.com/ubuntu/ bionic-backports main restricted universe multiverse

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu bionic partner
# deb-src http://archive.canonical.com/ubuntu bionic partner

deb http://security.ubuntu.com/ubuntu bionic-security main restricted
# deb-src http://security.ubuntu.com/ubuntu bionic-security main restricted
deb http://security.ubuntu.com/ubuntu bionic-security universe
# deb-src http://security.ubuntu.com/ubuntu bionic-security universe
deb http://security.ubuntu.com/ubuntu bionic-security multiverse
# deb-src http://security.ubuntu.com/ubuntu bionic-security multiverse
```

Figura 11.- Archivo sources.list

- ✓ Las líneas que comienzan por `deb http://...` son los repositorios.
- ✓ Las líneas que comienzan por `deb-src http://...` son los repositorios del código fuente y no es necesario que estén activados.
- ✓ Las líneas que comienzan por una almohadilla (`#`) son comentarios. Para desactivar un repositorio, basta añadir la `#` delante y para volver a activarlo, basta con borrar la `#`.
- ✓ En dichas líneas aparece el servidor. En el ejemplo es el servidor de España `es.archive.ubuntu.com`. Esta parte de las líneas puede variar y dirigir los repositorios a un servidor espejo o al servidor principal.
- ✓ En todas las líneas de los repositorios debe de aparecer el mismo nombre de versión de Ubuntu antes del nombre del componente (al final de la línea). En este caso es `bionic`, nombre de Ubuntu 20.04. Si, por lo que sea, apareciera un nombre de otra versión de Ubuntu, dará error al actualizar.
- ✓ El resto de líneas que empiezan por `## ...` son comentarios para saber qué es cada apartado. Si borramos las almohadillas de estas líneas dará error al actualizar.

Los repositorios de software de Ubuntu se organizan en cuatro componentes. Son:

- ✓ Main.- Este componente contiene aplicaciones que son software libre, que pueden ser redistribuidas y que están soportadas completamente por el equipo de Ubuntu. Esto incluye las aplicaciones de código abierto más populares y más seguras disponibles, muchas de las cuales son instalados por defecto cuando instalamos Ubuntu.
- ✓ Restricted.- Está reservado para software que es utilizado muy comúnmente, y que es soportado por el equipo de Ubuntu a pesar de que no esté disponible bajo una licencia completamente libre. Debido a ello, es posible que no se pueda proporcionar

soporte completo para este software debido a que el equipo de Ubuntu no está autorizado a modificar dicho software, sino simplemente a enviar informes de problemas a los autores verdaderos.

- ✓ Universe.- Podemos encontrar casi cualquier paquete de software de código abierto de software disponible bajo una variedad de licencias menos abiertas, todo construido automáticamente a partir de una variedad de fuentes públicas. Todo este software es compilado, sometido a las bibliotecas y utilizando las herramientas que forman parte de "main", pero viene sin ninguna garantía de obtener actualizaciones de seguridad y soporte.
- ✓ Multiverse.- contiene software que es "no libre", que significa que los requisitos de licencia de este software no coinciden con la política de licencias del componente "main" de Ubuntu. Es responsabilidad del usuario que se instala software de este componente el verificar los derechos y responsabilidades que implica emplear este software. Este software no está soportado, y generalmente no puede ser corregido ni puede ser actualizado, por lo que debe ser empleado a responsabilidad del usuario.

Cada vez que modificamos este archivo es necesario realizar una actualización de la lista de paquetes de los repositorios. Aunque podemos modificar el contenido de este fichero para añadir nuevos repositorios y tener así un abanico más amplio de software para instalar en nuestro sistema es recomendable hacerlo mediante una herramienta gráfica que vamos a ver a continuación.

5.1.4 Software y actualizaciones

Esta herramienta nos permite gestionar los repositorios de software desde donde podremos instalar aplicaciones. Accedemos a ella desde *Configuración del Sistema* → *Aplicaciones*. También podemos buscarla en el lanzador.

La primera pestaña es *Software de Ubuntu*. Aquí encontramos los repositorios oficiales y los mantenidos por la comunidad que deben de estar siempre activados. No es necesario activar la casilla de *Código fuente*.

No activar la casilla de CD-ROM/DVD ya que, al actualizarse el sistema, buscará en el disco de instalación de Ubuntu y al no encontrarlo, dará error.

Tenemos la opción *Descargar desde*, donde elegiremos los servidores a utilizar (principal, España y Otro). Si seleccionamos "Otro", se abrirá una nueva ventana con un listado de servidores espejo de los distintos países y un botón para *Seleccionar el mejor servidor*.

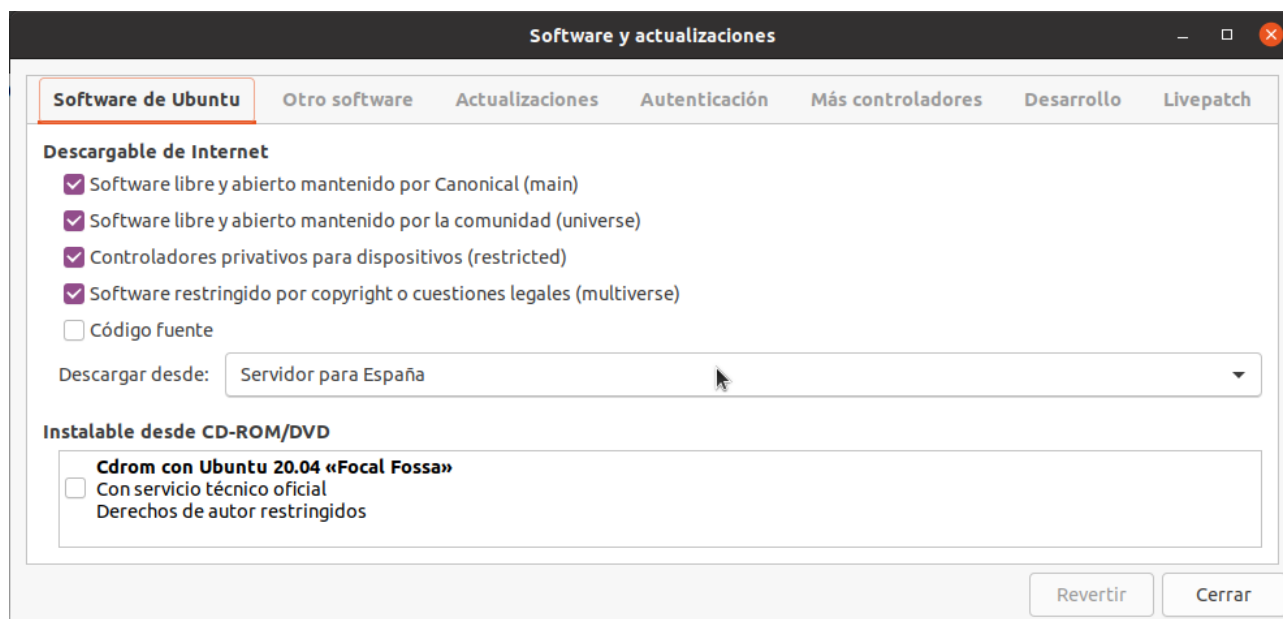


Figura 12.- Software y actualizaciones

En la pestaña *Otro Software* se almacena otro tipo de software proporcionado por Ubuntu. Inicialmente solamente aparece *Socios de Canonical* que consiste en software propietario (no libre) que ha llegado a un acuerdo con Ubuntu para tener acceso a ciertas aplicaciones. Aquí aparecerán también, los repositorios que nosotros añadamos.

En la pestaña *Más Controladores* están disponibles los controladores de dispositivo privativo. Si deseamos utilizarlos, en lugar del controlador libre que Ubuntu instala por defecto, lo seleccionamos y hacemos clic en *Aplicar cambios*.

5.2 Instalación de aplicaciones desde Ubuntu Software

Ubuntu Software es la aplicación para instalar aplicaciones. Consiste en un catálogo de miles de aplicaciones libres y otro software organizados por categorías cuya instalación se ha simplificado para que los usuarios profanos en informática puedan instalar aplicaciones de forma fácil e intuitiva. Además dispone de una potente herramienta de búsqueda de aplicaciones.

Disponemos de un icono en el lanzador para abrir el *Ubuntu Software*. La pantalla principal es la siguiente:

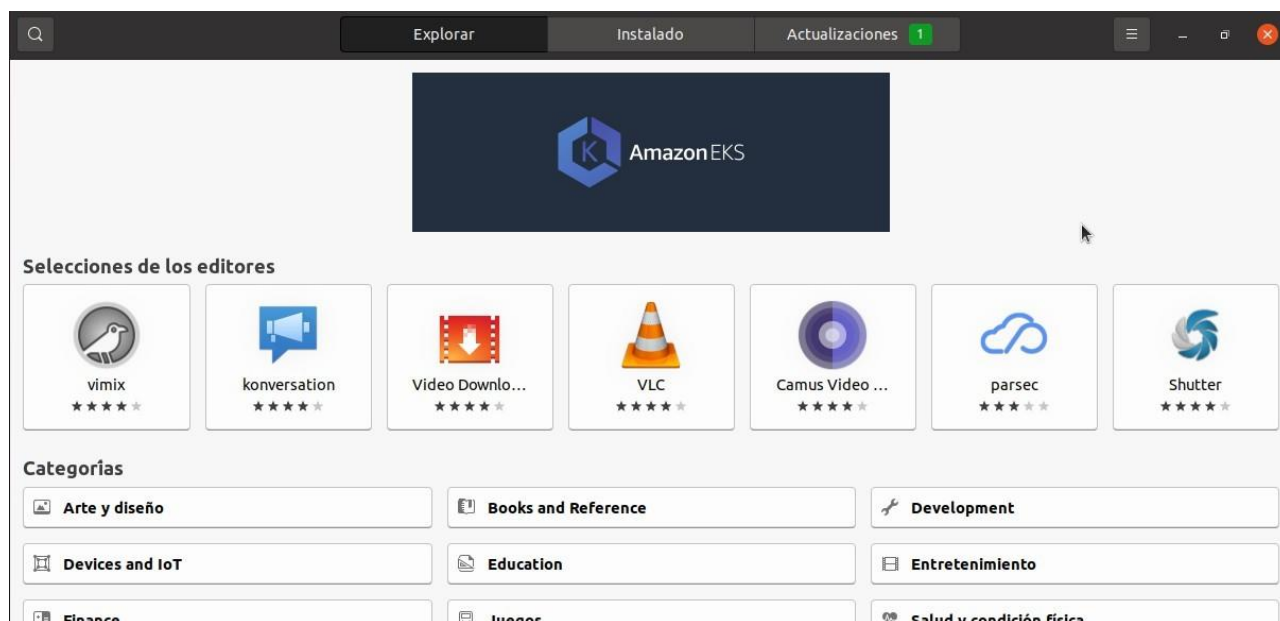


Figura 13.- Ubuntu Software

En la parte superior vemos tres botones para acceder a todas las aplicaciones, las aplicaciones instaladas y las actualizaciones disponibles. Justo debajo de esta barra de herramientas disponemos de un cuadro de búsqueda para introducir parte del nombre o la descripción de la aplicación que estamos buscando. Cada vez que tecleamos un texto muestra una lista de las aplicaciones que se ajustan al texto introducido.

En la parte central disponemos de una lista de categorías en las que se agrupan las aplicaciones. Cuando hacemos clic en una categoría se muestran las aplicaciones de esa categoría ordenadas por puntuación.

En la parte inferior nos propone un conjunto de aplicaciones recomendadas para instalar, aunque puede que algunas ya esté instalada y venga etiquetada como tal.

5.2.1 Instalar una aplicación

Para instalar una aplicación tenemos que seguir los siguientes pasos:

1. Buscar una aplicación utilizando las categorías o el cuadro de búsqueda.
2. Hacer clic sobre la aplicación elegida para mostrar información detallada sobre ella.
3. Si se quiere instalar directamente hacer clic en *Instalar*. Desde la pantalla de información también hay disponible un botón *Instalar*. Hay que autenticarse como usuario administrador.

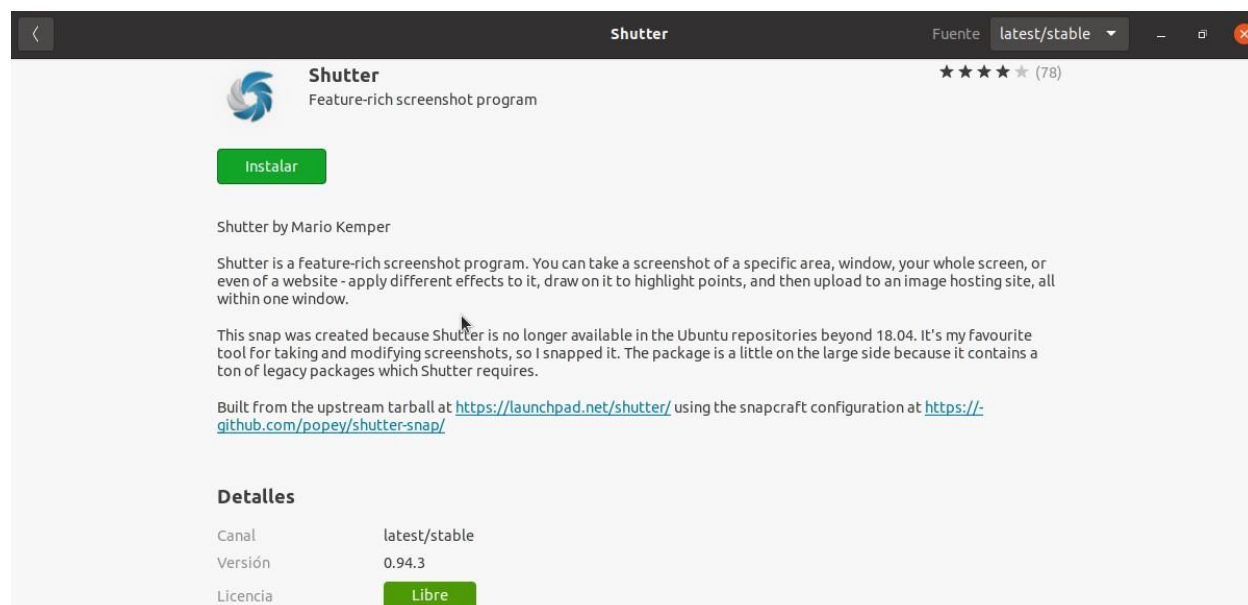


Figura 14.- Instalación de Geany

4. En la ventana de información aparece una descripción de la aplicación, su puntuación y el número de reseñas. Debajo veremos los detalles de la aplicación, como versión tipo de licencia, categoría y tamaño. Finalmente aparecen las reseñas de los usuarios que previamente instalaron esta aplicación.

La instalación comenzará. Primero descargará los paquetes necesarios, incluyendo aquellos de los que dependa la aplicación. Aparecerá una barra de progreso indicando el estado de la instalación.

5.2.2 Desinstalar una aplicación

Si hemos dejado de utilizar una aplicación podemos desinstalarla para que no ocupe espacio en el disco duro. Para ello seguir los siguientes pasos:

1. En la barra de herramientas hacer clic en *Instaladas* y buscar la aplicación que queremos desinstalar.
2. Hacer clic en el botón *Desinstalar*.
3. Autenticarse como usuario administrador.

5.2.3 Añadir una reseña

Cuando hemos usado con frecuencia una aplicación porque es muy adecuada para nuestras necesidades y la conocemos bien podemos añadirle una reseña de forma que otros usuarios interesados en instalarla puedan leerla y tener una idea más aproximada sobre el funcionamiento o las características de la aplicación. Para añadir una reseña seguir los siguientes pasos:

1. En la barra de herramientas hacer clic en *Instalado* y buscar la aplicación que queremos comentar.

2. Hacer clic en la aplicación que aparece en la lista de resultados.
3. Hacer clic en el enlace *Escribir una reseña*.
4. Escribir la valoración, el resumen y la reseña.
5. Hacer clic en el botón *Publicar*.
6. Si no tienes una cuenta de Ubuntu One introduce tu dirección de correo y crea una.
7. Si tienes una cuenta de Ubuntu One haz clic en *Ya tengo una cuenta*. Si elegiste éste último escribe tu dirección de correo y clave. Posteriormente haz clic en *Continuar*.
8. Hacer clic en el botón *Aceptar*.

A partir de entonces la valoración y la reseña aparecerán en la descripción de la aplicación y el resto de usuarios podrá verla.

Opiniones

57

7

3

2

9

78 puntuación total

[Escribir una opinión](#)

★★★★★ **buen programamas** 23 de febrero de 2021

Atreides

Rapido y Facil de usar

¿Le ha resultado útil esta opinión? ☐ Sí ☐ No [Informar...](#)

Figura 15.- Reseña de aplicación

5.3 Instalación de paquetes binarios. Comandos APT

Advanced Package Tools (APT) es un conjunto de comandos en línea que se emplean para gestión de paquetes. Entre sus principales características se encuentra manejo automático de conflictos, actualización de archivos de configuración para las aplicaciones que así lo requieran, etc.

Existen varios comandos APT los cuales se encargan de tareas específicas, todas ellas relacionadas con la gestión de paquetes.

5.3.1 Comando `apt`

El comando `apt` se emplea para la gestión de paquetes binarios. Mediante esta herramienta podemos hacer una gestión completa de los paquetes binarios, lo que incluye

su instalación, borrado, búsqueda, arreglar dependencias rotas, etc.

Sintaxis

```
apt [opciones] comando
```

Comandos

install paquete ...

Instala o actualiza a su última versión la lista de paquetes

update

Descarga desde los repositorios fuente la lista de paquetes disponibles.

upgrade [paquete ...]

Instala la versión más reciente de los paquetes indicados. Si se omite una lista de paquetes actualiza todos los paquetes instalados en el sistema

full-upgrade

Realiza la misma función que upgrade pero quitando los paquetes actualmente instalados si fuera necesario para actualizar el sistema como un conjunto.

remove paquete ...

Desinstala los paquetes indicados. Sus archivos de configuración permanecen y no son borrados.

purge paquete ...

Igual que remove pero eliminando también los archivos de configuración.

autoremove

Desinstala paquetes que se instalaron automáticamente para satisfacer las dependencias de algún paquete, pero que ya no son necesarios.

search patrón

Busca paquetes que se ajustan al patrón y los lista por pantalla.

show paquete ...

Muestra información de los paquetes indicados.

list patrón

Lista paquetes que satisfacen ciertos criterios.

dist-upgrade

Además de realizar las acciones de upgrade, trata inteligentemente los cambios de dependencias debidos a las nuevas versiones de paquetes. Apt tiene un sistema «inteligente» de resolución de conflictos, y si es necesario tratará de actualizar los paquetes más importantes a costa de los menos importantes.

Opciones

-d

--download-only

Sólo descarga los ficheros de los paquetes, no los desempaqueta ni los instala.

-f

--fix-broken

Intenta arreglar un sistema con dependencias actualmente rotas. Si se usa esta opción junto a «install»/«remove» se puede omitir cualquier paquete para permitir a APT deducir una posible solución.

-S

--simulate

No realiza ninguna acción, simula lo que hubiese ocurrido pero sin hacer cambios reales en el sistema.

--reinstall

Reinstala los paquetes ya instalados, incluso si son la última versión disponible del paquete

En aquellas opciones que haya que especificar una lista de paquetes estos irán separados por un espacio en blanco.

5.3.2 Comando apt-get

El comando `apt-get` también se emplea para la gestión de paquetes binarios. Esta herramienta estaba disponible antes de `apt` la cual combina las funciones de `apt-get` y `apt-search`. Las opciones disponibles en `apt` también se pueden usar aquí, pero además contamos con las siguientes:

Sintaxis

```
apt-get [opciones] comando
```

check

Es una herramienta de diagnóstico. Actualiza la caché de paquetes y revisa dependencias rotas.

download

Descarga los paquetes en el directorio activo.

5.3.3 Comando apt-cdrom

Herramienta APT para la gestión de paquetes en discos ópticos. `apt-cdrom` se usa para añadir un disco óptico a la lista de fuentes disponibles de APT

Sintaxis

```
apt-cdrom [opciones] comando
```

Comandos

add

Añade un disco nuevo a la lista de fuentes. Desmontará el dispositivo del disco y pedirá que se inserte un disco para iniciar el análisis y copiado de los ficheros de índice. Si el disco no contiene un directorio `disk` apropiado, se le pedirá un título descriptivo.

El comando `apt-cdrom` se emplea principalmente para añadir a la lista de

repositorios un CD-ROM con paquetes.

5.4 Snap

Ubuntu Snap es una nueva forma de instalar aplicaciones en Ubuntu que se ha introducido desde la versión 16.04. Trata de simplificar, aun más, la instalación de aplicaciones para los usuarios menos avanzados y de resolver los problemas asociados a las dependencias.

Uno de los inconvenientes que presenta la instalación de aplicaciones en Linux, mediante los anteriores métodos, es el problema de las dependencias. Es habitual que una aplicación requiera (o tenga dependencias con) ciertos paquetes. Entonces tenemos que instalar estas dependencias previamente, que a veces pueden no estar disponibles (según qué versión del sistema operativo estemos ejecutando). Estos paquetes quedan en el sistema si decidimos desinstalar la aplicación más tarde, y que por supuesto no son nada fáciles de manejar por los usuarios menos avanzados.

Se supone que los usuarios de Linux tienen un nivel más avanzado, y lidiar con dependencias e instalar paquetes es algo fácil para este tipo de usuarios. Pero si hablamos de usuarios que tienen unas nociones muy básicas de informática, esto puede ser un problema. Aquí es donde entra en juego Ubuntu Snap, para hacer todo esto más sencillo si cabe. Y sobre todo para resolver el tema de las dependencias.

Ubuntu Snap es una nueva tecnología introducida en Ubuntu 16.04 que permite empaquetar una aplicación cualquiera en lo que se denomina un “paquete snap”, que contiene la aplicación en cuestión junto a sus dependencias. Evidentemente esto tiene un inconveniente, y es el tamaño del paquete snap, que al contener las dependencias, ocupan más espacio en disco.

Sin embargo, el uso de snap tiene una serie de ventajas que contrarrestan la desventaja del espacio en disco. Estas son:

- ✓ Desaparecen los problemas de dependencias. Instalar el paquete snap es todo lo que necesitamos para ejecutar una aplicación.
- ✓ Se pueden instalar aplicaciones nuevas en versiones antiguas, ya que las dependencias (que pueden no estar disponibles en esa versión antigua del sistema operativo) ya están contenidas y vienen en el paquete snap.
- ✓ Instalar y desinstalar (por completo) aplicaciones también es más sencillo. Hasta ahora, las aplicaciones que requerían dependencias, cuando se desinstalaban, dejaban “colgando” estas dependencias, que en algunos casos no eran utilizadas por ninguna otra aplicación. Con Ubuntu Snap esto no es un problema, porque al desinstalar eliminamos todo el contenido del paquete snap, que es una entidad aislada del resto del software del sistema.
- ✓ Se aumenta la seguridad. Las aplicaciones que instalamos mediante Snap no afectan al resto de aplicaciones, en el sentido de que son completamente independientes.

- ✓ A nivel de usuario, las aplicaciones son más sencillas de instalar. Ahora sí que se cumple lo de “instalar con un click”.

Por supuesto, Ubuntu Snap es una tecnología complementaria (no excluye a los métodos hasta ahora habituales), que viene a sumarse al resto de métodos de instalación que ya conocemos (Centro de Software, paquetes deb, etc.).

5.5 Instalación de paquetes binarios con dpkg

Es una herramienta para instalar, compilar, eliminar y manipular los paquetes de Debian. Se invoca con parámetros de línea de órdenes, que consisten de una sola acción y cero o más opciones. La acción o parámetro dice a dpkg qué hacer, mientras que las opciones controlan de una manera u otra su comportamiento.

Sintaxis

```
dpkg [opciones] accion
```

Acciones

`-i paquete`

`--install paquete`

Instala el paquete. El paquete tiene que ser un archivo .deb previamente descargado.

`-r paquete`

`--remove paquete`

Elimina el paquete pero mantiene los archivos de configuración.

`-P paquete`

`--purge paquete`

Elimina el paquete y borra los archivos de configuración.

`-s paquete`

`--status paquete`

Muestra el estado del paquete.

`-l patrón`

`--list patrón`

Lista los paquetes cuyos nombres se ajustan al patrón.

`--configure paquete`

Configura un paquete que esté desempaquetado.

El comando dpkg comprueba las dependencias cuando se instala un paquete, pero no descarga ni instala los paquetes de los cuales dependa. Simplemente lo desempaqueta y no lo configura. Por tanto las dependencias tienen que resolverse manualmente. Tampoco desinstala un paquete si mantiene dependencias con otros. Debido a ello hay que utilizar esta herramienta con extremo cuidado.

5.6 Añadir repositorios de terceros

Los repositorios de terceros o PPA (*Personal Package Archive*) son depósitos de

software alojados en Launchpad que se pueden utilizar para instalar (o actualizar) paquetes que no están disponibles en los repositorios que trae por defecto Ubuntu. Estos PPAs son específicos para una versión de Ubuntu en concreto, por lo que no es recomendable usarlos en otras distribuciones ni usar los de otra versión de Ubuntu.

La nomenclatura de los PPA es: `ppa:creador/nombre`. Por ejemplo, `ppa:webupd8team/java` es el PPA para Oracle Java (nombre del PPA) de webupd8team (creador y mantenedor de dicho PPA).

5.6.1 Añadir un PPA a la lista de repositorios e instalar el software

Un PPA se pueden añadir desde la línea de comandos (terminal) o mediante una interfaz gráfica (Software y actualizaciones):

Para añadirlo desde la terminal se utiliza el comando `add-apt-repository`, seguido del PPA. Por ejemplo:

```
sudo add-apt-repository ppa:webupd8team/java
```

Nos pedirá confirmación para añadir el repositorio. Pulsamos Enter para continuar y después de añadirlo, actualizará la lista de repositorios, para que los paquetes disponibles de dicho PPA pueda ser encontrados por el sistema.

Para añadirlo desde la interfaz gráfica seguimos los siguientes pasos:

1. Buscamos en el tablero *Software y actualizaciones*, que vimos en el apartado anterior o también desde el *Centro de Software de Ubuntu*, con la opción de menú *Editar* → *Orígenes del Software*.
2. Vamos a la pestaña *Otro software* y pulsamos en el botón *Añadir*.
3. Escribimos el PPA. Por ejemplo: `ppa:webupd8team/java`.
4. Hacemos clic en el botón *Cerrar*.
5. El sistema nos pedirá actualizar la lista de software.

Una vez tenemos añadido el repositorio podemos instalar el software con un comando `apt-get` visto anteriormente.

```
sudo apt-get install oracle-java8-installer
```

5.6.2 Eliminar un PPA

Si no queremos tener un PPA en la lista de repositorios lo mejor es eliminarlo y volver a actualizar la lista de repositorios. Al igual que antes podemos realizar esta tarea desde una ventana de terminal o desde *Software y actualizaciones*.

Desde la terminal, podemos quitar el PPA y elegir si queremos mantener los paquetes instalados o actualizados desde dicho repositorio. Para ello se utiliza el comando `add-apt-repository --remove`, seguido del PPA. Ejemplo:

```
sudo add-apt-repository --remove ppa:webupd8team/java
```

Para quitar un PPA de nuestro sistema y además desinstalar los paquetes que hayamos instalado o actualizado desde dicho PPA, dejando el sistema como antes de añadirlo, se utiliza el comando `ppa-purge`, seguido del PPA. Pero no viene instalado por defecto, así que hay que instalarlo antes con:

```
sudo apt-get install ppa-purge
```

Y ya podemos eliminar el PPA con `ppa-purge` seguido del PPA. Ejemplo:

```
sudo ppa-purge ppa:webupd8team/java
```

En ambos casos, hay que actualizar la lista de paquetes del repositorio con:

```
sudo apt-get update
```

El propósito de `ppa-purge` es restaurar los paquetes originales desde los repositorios de Ubuntu en caso de que algo vaya mal, pero podemos encontrarnos con que un paquete instalado desde un PPA no existe en los repositorios oficiales de Ubuntu y no puede ser desactualizado ni tampoco eliminado automáticamente, por lo que tendremos que eliminarlo manualmente.

Si lo eliminamos desde la interfaz gráfica seguiremos los siguientes pasos:

1. Abrir *Software y actualizaciones*.
2. Vamos a la pestaña *Otro software* y seleccionamos el PPA que deseamos eliminar.
3. Hacer clic en el botón *Quitar*.

Cada PPA tiene dos líneas, una para los paquetes compilados y otra para el código fuente, por lo debemos de quitar ambas.

Al añadir un PPA, se crea dos archivos en `/etc/apt/sources.list.d`, uno con el PPA `.list` y otro de respaldo `.list.save`. Los archivos se nombran según el nombre de PPA y su versión de Ubuntu, por ejemplo: `webupd8team-ubuntu-java-bionic.list` y `webupd8team-ubuntu-java-bionic.list.save`. Para eliminar el PPA, bastaría con borrar dichos archivos y después actualizar.

5.7 Instalación de aplicaciones desde código fuente

Instalar programas o aplicaciones nuevas en Ubuntu es una de las tareas mas sencillas que podemos hacer con Linux, sobretodo cuando se usa una herramienta como el *Centro de Software de Ubuntu* que nos pone todas las aplicaciones disponibles en los repositorios de Ubuntu a un solo clic.

Sin embargo vamos a suponer que somos un usuario experimentado en Ubuntu y nos vemos en la obligación a utilizar el sistema tradicional de instalación de aplicaciones en Linux que usa el código fuente de la aplicación, disponible generalmente en los sitios web oficiales de cada una de estas, ya sea porque no podemos encontrarla en los repositorios

oficiales o simplemente porque queremos instalar una versión más reciente.

A continuación aprenderemos en cuatro sencillos pasos cómo instalar programas en Ubuntu directamente desde el código fuente provisto por los desarrolladores de estos programas.

5.7.1 Preparar el sistema para la construcción de paquetes

Ubuntu no instala por defecto algunas herramientas necesarias. Necesitamos instalar los paquetes `build-essential` para construir el paquete y `checkinstall` para meterlo en el gestor de paquetes. En una ventana de terminal ejecutamos el siguiente comando `apt-get`.

```
sudo apt-get install build-essential checkinstall
```

En el caso de que necesitemos descargar los archivos fuentes de proyectos que no disponen de una versión de lanzamiento oficial directamente desde los servidores que los almacenan, tendremos que tener las siguientes herramientas:

```
sudo apt-get install cvs subversion git-core mercurial
```

Además, debemos tener un directorio común para construir los paquetes. Se recomienda utilizar el directorio `/usr/local/src`, aunque en realidad se pueden poner en cualquier otro. Hay que asegurarse que este directorio tiene permiso de escritura para el usuario.

```
sudo chown $USER /usr/local/src  
sudo chmod u+rw /usr/local/src
```

5.7.2 Descargar el código fuente de la aplicación

La mayoría de las aplicaciones disponibles en código fuente vienen en forma de archivos empaquetados y comprimidos conocidos como *tarballs*. Estos archivos tienen extensión `.tar.gz` o `.tar.bz2` ya que se empaquetaron y comprimieron con el comando `tar`. Al descargarlos los guardaremos en el directorio que preparamos en el apartado anterior y extraer su contenido desde el Nautilus haciendo clic con el botón derecho del ratón sobre el archivo y seleccionando la opción del menú contextual *Extraer aquí* o utilizando el siguiente comando:

```
tar -xvzf aplicacion.tar.gz
```

o

```
tar -xvjf aplicacion.tar.bz2
```

Una vez se descomprimen y desempaquetan estos archivos crearán un nuevo directorio con el nombre y la versión de la aplicación a instalar. Nos ubicaremos en este directorio para comenzar el proceso de construcción del paquete.

Existe una probabilidad muy alta de que haya unas instrucciones de instalación de la aplicación en el archivo `README`, por lo que se recomienda encarecidamente leer este

archivo y seguir las instrucciones antes de empezar la construcción del paquete.

5.7.3 Resolviendo las dependencias

Las herramientas de instalación que hemos visto hasta ahora tienen la ventaja de que resuelven las dependencias instalando los paquetes necesarios junto con la aplicación. Desafortunadamente no ocurre lo mismo con las instalaciones desde código fuente por lo que tenemos que resolverlo manualmente.

Ubuntu dispone de una aplicación, `apt-file`, que permite conocer cuáles son los paquetes que contienen los archivos o dependencias necesarias para construir un paquete desde su código fuente. Primero deberemos instalarla y después hay que actualizar su base de datos. Esto descargará una lista de todos los paquetes disponibles y todos los ficheros que estos paquetes contienen. Para ello ejecutamos los siguientes comandos:

```
sudo apt-get install apt-file
sudo apt-file update
```

El programa `apt-file` tiene un par de interesantes funciones, las dos más útiles son `apt-file search`, la cual busca un fichero en particular, y `apt-file list`, que lista todos los archivos de un paquete dado.

Para comprobar las dependencias de la aplicación a instalar tenemos que cambiarnos al directorio donde hemos descargado el tarball y extraer su contenido. Esto creará un nuevo directorio con el nombre y versión de la aplicación. Este directorio contiene un archivo llamado `configure`, el cuál es un script para asegurar que la aplicación puede compilarse. Para ejecutarlo nos cambiamos al directorio de la aplicación y ejecutamos

```
./configure
```

El comando comprobará y tenemos todos los programas necesarios para instalar la aplicación. En la mayoría de los casos no los tendremos y veremos un error con un mensaje acerca del programa que falta. Este mensaje aparecerá en la última línea de salida del comando que puede ser algo como esto

```
configure: error: Library requirements (gobbletycook) not
met, ...
```

En la línea anterior listará el nombre del fichero que no puede encontrar. Lo que necesitamos hacer es ejecutar el siguiente comando

```
apt-file search fichero
```

Este comando nos informará de que paquete contiene el fichero necesario y solamente tendremos que instalar este paquete con `apt-get`.

```
sudo apt-get install paquete_necesario
```

Intentamos ejecutar de nuevo `./configure` y vemos si funciona. Si el resultado del comando acaba en una línea como la siguiente significa que no hay errores y podemos continuar con la instalación.

```
config.status: creating Makefile
```

El comando `./configure` configurará la aplicación con los valores predeterminados, pero existen un conjunto de opciones de configuración para una aplicación. Para conocer estas opciones podemos ejecutar `./configure --help` o leer el archivo `README` que se encuentra en el directorio de la aplicación.

5.7.4 Construir e instalar la aplicación

Llegados a este punto hemos hecho lo más difícil, que es resolver las dependencias. Ahora solo queda ejecutar el siguiente comando

```
make
```

Este comando realiza la compilación de la aplicación. Si el programa es muy grande o el PC es muy lento tardará bastante. Cuando termine ejecutamos el siguiente comando

```
checkinstall --fstrans=0
```

Este comando pone la aplicación en el administrador de paquetes, para que su desinstalación sea mucho más sencilla. Eso es todo, si los pasos descritos anteriormente han finalizado sin problema alguno, ya puedes ejecutar la aplicación en Ubuntu y comenzar a utilizarla.

5.7.5 Ejemplo de instalación

Vamos a ver un ejemplo práctica de instalación desde código fuente. En este caso vamos a instalar un editor HTML llamado BlueFish. Primero tengo que descargar el *tarball* con el código fuente empaquetado desde <https://www.bennewitz.com/bluefish/stable/source/> y lo guardo en `/usr/local/src` que habré creado y puesto los permisos y propietario adecuados para ello.

Posteriormente, desempaquete el *tarball* con el comando

```
tar -xzvf bluefish-1.0.1.tar.gz
```

Habrá creado el directorio `/usr/local/src/bluefish-1.0.1`. Nos cambiamos a él e intentamos resolver las dependencias. Para ello ejecutamos

```
./configure
```

Vemos en la siguiente captura de pantalla que ha dado error y que no encuentra el `pkg-config`.

```
checking for Mac OS X... no
checking for a BSD-compatible install... /usr/bin/install -c
checking how to run the C preprocessor... gcc -E
checking for egrep... grep -E
checking for ANSI C header files... yes
checking for pkg-config... no
configure: error: pkg-config not found please install pkg-config
```

Figura 16.- Resultado de ./configure

Por tanto las dependencias no están resueltas y hay que instalar los paquetes que falta. Para saber que paquete incluye dicho script hay que ejecutar una búsqueda con `apt-file`. Ejecutamos el siguiente comando

```
apt-file search pkg-config.
```

Vemos que muestra muchos paquetes, entre ellos `pkg-config`. En este ha encontrado el archivo `/usr/bin/pkg-config` que es justo el que necesitamos. Generalmente es un buen indicio instalar el paquete que contiene archivos del directorio `/usr/bin` ya que en esta ubicación se encuentran comandos y utilidades del sistema operativo. Así que instalaremos `pkg-config` y después volveremos a intentar resolver dependencias con `./configure`.

```
nant: /usr/share/doc/nant/help/functions/pkg-config.is-between-version(System.String,System.String,System.String).html
nant: /usr/share/doc/nant/help/functions/pkg-config.is-exact-version(System.String,System.String).html
nant: /usr/share/doc/nant/help/functions/pkg-config.is-max-version(System.String,System.String).html
pkg-config: /etc/dpkg/dpkg.cfg.d/pkg-config-hook-config
pkg-config: /usr/bin/pkg-config
pkg-config: /usr/bin/x86_64-linux-gnu-pkg-config
pkg-config: /usr/lib/pkg-config.multiarch
pkg-config: /usr/share/doc/pkg-config/AUTHORS
pkg-config: /usr/share/doc/pkg-config/NEWS.gz
pkg-config: /usr/share/doc/pkg-config/README
pkg-config: /usr/share/doc/pkg-config/changelog.Debian.gz
pkg-config: /usr/share/doc/pkg-config/copyright
pkg-config: /usr/share/doc/pkg-config/pkg-config-guide.html
pkg-config: /usr/share/man/man1/pkg-config.1.gz
pkg-config: /usr/share/pkg-config-crosswrapper
pkg-config: /usr/share/pkg-config-dpkghook
pkg-config-aarch64-linux-gnu: /usr/bin/aarch64-linux-gnu-pkg-config
```

Figura 17.- Resultado de apt-file

Para instalar el paquete ejecutamos el siguiente comando

```
apt-get install pkg-config
```

Ahora volvemos a ejecutar `./configure` y vuelve a dar error. En la siguiente captura de pantalla vemos que falta `libgtk2.0-dev`. Volvemos a buscar los paquetes para este archivo.

```
checking for Mac OS X... no
checking for a BSD-compatible install... /usr/bin/install -c
checking how to run the C preprocessor... gcc -E
checking for egrep... grep -E
checking for ANSI C header files... yes
checking for pkg-config... /usr/bin/pkg-config
checking for gtk+-2.0... no
configure: error: libgtk2.0-dev missing please install libgtk2.0-dev
```

Figura 18.- Resultado de ./configure

En esta ocasión solamente nos ofrece un paquete con el mismo nombre que el archivo. Por tanto instalamos dicho paquete.

```
libgtk2.0-dev: /usr/share/doc/libgtk2.0-dev/AUTHORS
libgtk2.0-dev: /usr/share/doc/libgtk2.0-dev/NEWS.gz
libgtk2.0-dev: /usr/share/doc/libgtk2.0-dev/README.gz
libgtk2.0-dev: /usr/share/doc/libgtk2.0-dev/changelog.Debian.gz
libgtk2.0-dev: /usr/share/doc/libgtk2.0-dev/copyright
```

Figura 19.- Resultado apt-file

Pudiera ocurrir que los resultados de `apt-file search` fueran múltiples. Se recomienda dar prioridad a aquellos paquetes que contienen el nombre del archivo buscado acabado en `.pc`. También es buena idea dar prioridad a aquellos paquetes cuyo nombre acaba en `dev`, ya que hace referencia a herramientas y bibliotecas empleadas en la construcción de programas. Es muy posible que al instalar un paquete, este incluya archivos que son necesarios y no están

```
configure: creating ./config.status
config.status: creating Makefile
config.status: creating icons/Makefile
config.status: creating src/Makefile
config.status: creating po/Makefile
config.status: creating data/Makefile
config.status: creating src/config.h
config.status: executing default-1 commands
-----
If you like this program, please let me know and send me
a postcard and tell me what city/country you're from:

Olivier Sessink
Thorbeckestraat 470
6702 CJ
Wageningen
The Netherlands
-----
```

Figura 20:- Última ejecución de `./configure` sin dependencias sin cumplir

Después de instalarlos volvemos a ejecutar `./configure` y en esta ocasión no vuelve a dar error. Por tanto, el siguiente paso es construir la aplicación ejecutando el siguiente comando

```
make
```

Cuando termine ejecutamos el siguiente comando

```
sudo checkinstall --fstrans=0
```

Nos hará varias preguntas. La primera nos indica si creamos una descripción de la aplicación a modo de documentación. Contestamos si (Y) e introducimos una descripción breve. Posteriormente nos presenta un conjunto de valores que aceptaremos por defecto pulsando Intro. Cuando termine la instalación veremos que ha tenido éxito.

```
===== Instalación exitosa =====  
Copying documentation directory...  
./  
./README  
./COPYING  
./NEWS  
./ChangeLog  
./TODO  
./AUTHORS  
./INSTALL
```

Figura 21.- Fin de la construcción de la aplicación

Podremos ejecutarla la aplicación escribiendo en la misma ventana de terminal `bluefish &`. El resultado será el siguiente

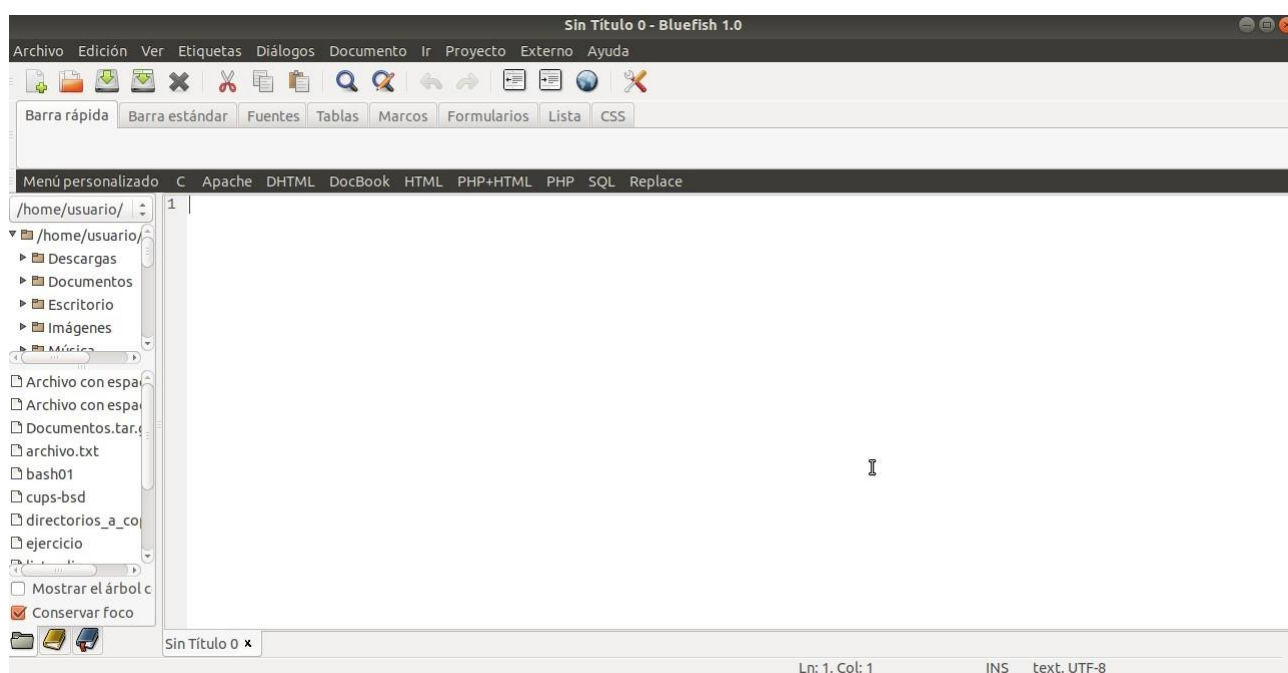


Figura 22.- Ejecución de Bluefish

En el icono que ha aparecido en el lanzador hacemos clic con el botón derecho del ratón y seleccionamos la opción *Añadir a los favoritos*. Así tendremos un icono permanente en el lanzador a la hora de ejecutar la aplicación.

Con el último comando ejecutado para su instalación se ha generado un paquete `.deb` en `/usr/local/src`. Para desinstalarla solo hay que ejecutar el siguiente comando

```
dpkg -r bluefish-1.0.1
```

5.8 Actualizaciones automáticas

Cuando se instala Ubuntu una de las primeras cosas que debemos hacer es actualizar nuestro sistema. Por defecto, las actualizaciones automáticas están activadas y al encender

el ordenador es posible que aparezca una ventana de *Actualización de software* avisándonos de que hay disponibles actualizaciones disponibles para descargar e instalar.



Figura 23.- Actualización de software

Si hacemos clic en el botón *Instalar ahora* comenzará el proceso de descarga y actualización de paquetes. Si hacemos clic en el botón *Recordármelo más tarde* volverá a ejecutarse la próxima vez que encendamos el ordenador. De todas formas, en cualquier momento podemos ejecutar *Actualización de software* buscándolo en la barra de búsqueda del tablero.

Además, también disponemos de *Software de Ubuntu* para ver las actualizaciones que hay disponibles en cualquier momento. En el botón *Actualizaciones* veremos el número de aplicaciones pendientes de actualizar. Si hacemos clic en este botón aparece esta lista de aplicaciones y podemos actualizarlas individualmente haciendo clic en el botón *Instalar* que hay al lado de cada una de ellas o todas a la vez haciendo clic en el botón *Instalar* de la barra de herramientas.

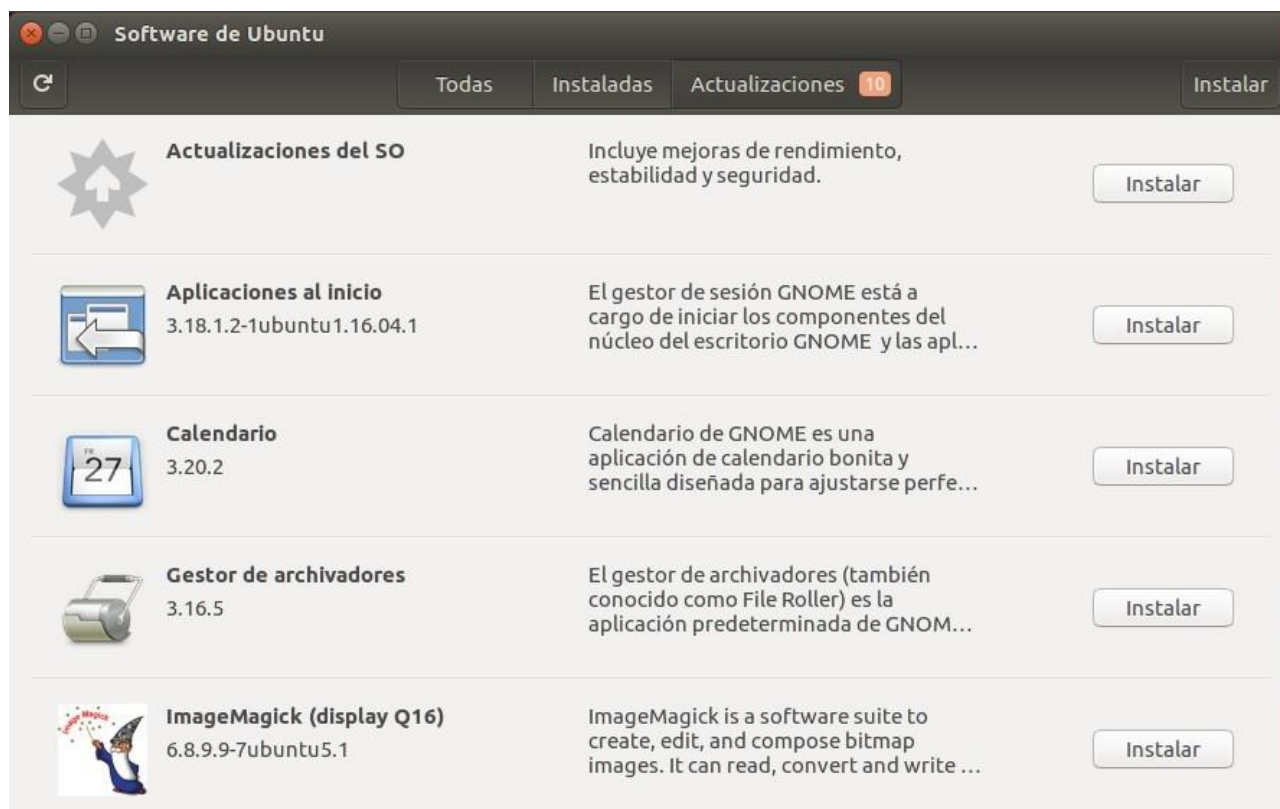


Figura 24.- Lista de actualizaciones

Si buscamos en el tablero *Software y actualizaciones* y vamos a la pestaña *Actualizaciones* podemos configurar las actualizaciones automáticas. Aquí configuramos todo lo relacionado con las actualizaciones (tipo, tiempo, versiones):

- ✓ Actualizaciones importantes de seguridad (xenial-security). Siempre debe de estar activada.
- ✓ Actualizaciones recomendadas (xenial-updates) por los desarrolladores de los paquetes. Siempre activada.
- ✓ Actualizaciones sin asistencia técnica (xenial-backports) da acceso a las últimas versiones de paquetes no soportados por Ubuntu. Si añadimos repositorios de terceros es recomendable mantenerla activada. Si quieres utilizar solo paquetes oficiales y anteponer la estabilidad a nuevas versiones, puedes dejarla desactivada.

También podemos configurar cuando y como se realizan las actualizaciones y si queremos que nos avise cuando salga una versión nueva de Ubuntu.

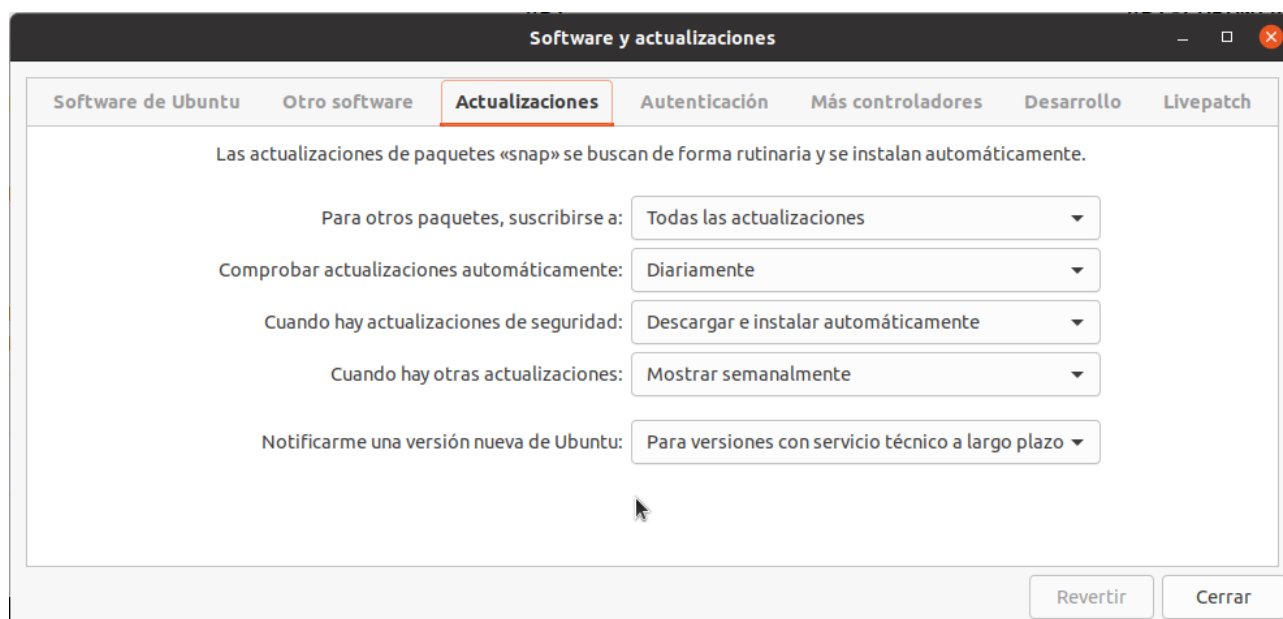


Figura 25.- Actualizaciones automáticas

6 Procesos

La más simple definición de un proceso podría ser que es una instancia de un programa en ejecución. A los procesos frecuentemente se les refiere como tareas. El contexto de un programa que está en ejecución es lo que se llama un proceso. Este contexto puede ser mas procesos hijos que se hayan generado del principal (proceso padre), los recursos del sistema que este consumiendo, sus atributos de seguridad (tales como su propietario y permisos de archivos así como roles y demás de Linux), etc.

Linux, como se sabe, es un sistema operativo multitarea y multiusuario. Esto quiere decir que múltiples procesos pueden operar simultáneamente sin interferirse unos con los otros. Cada proceso tiene la "ilusión" que es el único proceso en el sistema y que tiene acceso exclusivo a todos los servicios del sistema operativo.

Programas y procesos son entidades distintas. En un sistema operativo multitarea, múltiples instancias de un programa pueden ejecutarse simultáneamente. Cada instancia es un proceso separado.

Cada proceso que se inicia es referenciado con un número de identificación único conocido como Process ID o PID, que es siempre un entero positivo. Prácticamente todo lo que se está ejecutando en el sistema en cualquier momento es un proceso, incluyendo el shell, el servicio gráfico que puede tener múltiples procesos, etc. La excepción a lo anterior es el kernel en sí, el cual es un conjunto de rutinas que residen en memoria y a los cuales los procesos a través de llamadas al sistema pueden tener acceso.

6.1 Listado de procesos

Podemos tener una idea de la actividad del sistema mirando un listado de procesos en el que podemos ver su estado y los recursos que están consumiendo. Disponemos de dos comandos que nos permiten hacer un listado de los procesos actualmente en el sistema: `ps`

y top.

6.1.1 Comando ps

El comando `ps` es el que permite informar sobre el estado de los procesos. Tiene una gran cantidad de opciones, incluso estas opciones varían dependiendo del estilo en que se use el comando. Estas variaciones sobre el uso de `ps` son las siguientes:

- ✓ Estilo UNIX, donde las opciones van precedidas por un guión.
- ✓ Estilo BSD, donde las opciones no llevan guión.
- ✓ Estilo GNU, donde se utilizan nombres de opciones largas y van precedidas por doble guión.

Sea cual sea el estilo utilizado, dependiendo de las opciones indicadas, varias columnas se mostrarán en el listado de procesos que resulte, estas columnas pueden ser entre muchas otras, las siguientes (y principales):

- ✓ `p` o `PID`.- Process ID, número único o de identificación del proceso.
- ✓ `P` o `PPID`.- Parent Process ID, identificación del proceso padre
- ✓ `U` o `UID`.- User ID, usuario propietario del proceso.
- ✓ `t` o `TT` o `TTY`.- Terminal asociada al proceso, si no hay terminal aparece entonces un '?'.
- ✓ `T` o `TIME`.- Tiempo de uso de CPU acumulado por el proceso.
- ✓ `c` o `CMD`.- El nombre del programa o comando que inició el proceso.
- ✓ `RSS`.- *Resident Size*, tamaño de la parte residente en memoria en kilobytes.
- ✓ `SZ` o `SIZE`.- Tamaño virtual de la imagen del proceso.
- ✓ `NI`.- *Nice*, valor *nice* (prioridad) del proceso, un número positivo significa menos tiempo de procesador y negativo más tiempo (-19 a 19).
- ✓ `C` o `PCPU`.- Porcentaje de CPU utilizado por el proceso.
- ✓ `STIME`.- *Starting Time*, hora de inicio del proceso.
- ✓ `S` o `STAT`.- Estado del proceso, estos pueden ser los siguientes:
 - ✗ `R` *runnable*, en ejecución, corriendo o ejecutándose.
 - ✗ `S` *sleeping*, proceso en ejecución pero sin actividad por el momento, o esperando por algún evento para continuar.
 - ✗ `T` *stopped*, proceso detenido totalmente, pero puede ser reiniciado.
 - ✗ `Z` *zombie*, difunto, proceso que por alguna razón no terminó de manera correcta y por tanto no ha liberado los recursos que tenía asignados, no debe haber

procesos zombis.

- ✗ *D uninterruptible sleep*, son procesos generalmente asociados a acciones de IO del sistema.
- ✗ *X dead*, muerto, proceso terminado pero que sigue apareciendo, igual que los Z no deberían verse nunca.

Sintaxis

```
ps [opciones]
```

Opciones

-e

Lista todos los procesos.

-u `lista_usuarios`

Lista los procesos de los usuarios de la lista.

-g `lista_grupos`

Lista los procesos de los grupos de la lista.

-f

Hace un listado completo. Los campos que incluyen son UID, PID, PPID, C, STIME, TTY, TIME, CMD.

Para un listado completo de todas las opciones teclear `ps -help`.

6.1.2 Comando top

Una herramienta muy usada y muy útil para la monitorización en tiempo real del estado de los procesos y de otras variantes del sistema es el comando `top`, es interactivo y por defecto actualiza la lista de procesos cada 3 segundos.

```
top - 17:03:32 up 2 min,  2 users,  load average: 4,69, 2,43, 0,93
Tareas: 153 total,   1 ejecutar, 151 hibernar,   0 detener,   1 zombie
%Cpu(s):  0,7 usuario,  1,0 sist,  0,0 adecuado, 98,0 inact,  0,3 en espera,  0,
KiB Mem:  1545148 total,  717056 used,  828092 free,   64304 buffers
KiB Swap: 1998844 total,    0 used, 1998844 free. 357452 cached Mem
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
1115	root	20	0	299412	50188	15052	S	1,0	3,2	0:03.64	Xorg
1732	rll	20	0	290740	62748	32116	S	0,7	4,1	0:04.03	compiz
23	root	20	0	0	0	0	S	0,3	0,0	0:00.23	kworker/0:1
1019	root	20	0	2196	632	492	S	0,3	0,0	0:00.04	acpid
1190	root	20	0	9808	1012	720	S	0,3	0,1	0:00.16	VBoxService
2180	rll	20	0	129828	19256	12724	S	0,3	1,2	0:01.11	gnome-termi+
2234	rll	20	0	6736	1484	1092	R	0,3	0,1	0:00.14	top
1	root	20	0	4584	2560	1444	S	0,0	0,2	0:05.64	init
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.11	ksoftirqd/0
4	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0H
6	root	20	0	0	0	0	S	0,0	0,0	0:00.18	kworker/u2:0
7	root	20	0	0	0	0	S	0,0	0,0	0:02.08	rcu_sched
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
10	root	rt	0	0	0	0	S	0,0	0,0	0:00.04	watchdog/0

Figura 26.- Salida del comando top

Sintaxis

```
top [opciones]
```

Opciones

-d intervalo

Actualiza la lista cada segundos ss.dd indicado en el intervalo.

-u usuario

Solo monitoriza los procesos del usuario.

-p proceso

Solo monitoriza el proceso indicado.

Los campos que puede listar son los siguientes:

- ✓ PID.- Id de proceso.
- ✓ PPID.- ID del proceso padre
- ✓ RUSER.- Nombre de usuario real propietario de la tarea.
- ✓ UID.- Id de usuario
- ✓ USER.- Nombre de usuario
- ✓ GROUP.- Nombre de grupo
- ✓ TTY.- Tty desde donde se invocó el proceso.

- ✓ PR.- Prioridad
- ✓ NI.- Valor Nice
- ✓ P.- Último número de procesador utilizado (en entorno multiprocesador).
- ✓ %CPU.- Uso de la CPU
- ✓ TIME.- Tiempo de CPU.
- ✓ TIME+.- Igual que TIME pero con centésimas de segundo.
- ✓ %MEM.- Uso de memoria.
- ✓ VIRT.- Tamaño de memoria virtual usado por el proceso en kb.
- ✓ SWAP.- Tamaño intercambiado en kb.
- ✓ RES.- Memoria residente en kb.
- ✓ CODE.- Tamaño del código en kb.
- ✓ DATA.- Tamaño de datos en kb.
- ✓ SHR.- Tamaño de la memoria compartida en kb.
- ✓ nFLT.- Número de fallos de página.
- ✓ nDRT.- Número de páginas sucias.
- ✓ S.- Estado del proceso.
- ✓ Command.- Comando en línea o nombre del programa.

Además, `top` es interactivo. Admite un conjunto de comandos para personalizar la salida. Si pulsamos la letra `h` veremos la ayuda de estos comandos interactivos. Para salir de `top` pulsamos la tecla `q`.

6.2 Enviar señales a procesos

El comando `kill`, que literalmente quiere decir matar, sirve no solo para matar o terminar procesos sino principalmente para enviar señales (signals) a los procesos. La señal por defecto es terminar o matar el proceso.

Sintaxis

```
kill [opciones] [pid ...]
```

Parámetros

`pid...`

Lista de procesos a los que se les envía la señal

Opciones

-señal

Señal que se envía al proceso.

-L

Lista las señales que se pueden enviar a un proceso.

Podemos obtener el PID de un proceso mediante el comando `ps` o `top`. Generalmente usamos el comando `kill` para matar los procesos que se han quedado bloqueados sin tener que reiniciar el sistema.

El comando `killall`, que funciona de manera similar a `kill`, pero con la diferencia de en vez de indicar un PID se indica el nombre del programa, lo que afectará a todos los procesos que tengan ese nombre. Así por ejemplo si se tienen varias instancias ejecutándose del servidor proxy `squid`, con `killall squid` eliminará todos los procesos que se estén ejecutando con el nombre `squid`.

Sintaxis

```
killall [opciones] programa
```

Parámetros

programa

Nombre del programa al que se le envía la señal

Opciones

-s señal

Señal que se envía al proceso. Por defecto es `SIGTERM`. También pueden especificarse la señal con el número o por nombre sin necesidad de `s`.

-I

No es sensible a minúsculas o mayúsculas en el nombre del proceso.

-r

--regex

Interpreta el nombre de proceso como una expresión regular

-u usuario

--user usuario

Envía la señal solo a los procesos que pertenecen al usuario

6.3 Ejecutar procesos en 2º plano

Cuando se trata ejecutar procesos en background (segundo plano) se utiliza el comando `nohup` o el operador `&`. Aunque realizan una función similar, no son lo mismo.

Si se desea liberar la terminal de un programa que se espera durará un tiempo considerable ejecutándose, entonces se usa `&`. Esto funciona mejor cuando el resultado del proceso no es necesario mandarlo a la salida estándar, como por ejemplo cuando se ejecuta una copia de seguridad o se abre una aplicación gráfica desde la consola o terminal. Para lograr esto basta con escribir el comando en cuestión y agregar al final el símbolo `&`.

Por ejemplo, si queremos abrir el navegador web desde un terminal de texto ejecutaríamos el siguiente comando

```
firefox &
```

Sin embargo lo anterior produce que el padre del proceso que se invocó sea el proceso de la terminal en sí, por lo que si cerramos la terminal o salimos de la sesión también se terminaran los procesos hijos que dependan de la terminal, no muy convenientes si se desea que el proceso continúe en ejecución.

Para superar este inconveniente se usa el comando `nohup` que permite al igual que `&` mandar el proceso en background y que este quede inmune a los bloqueos o a la terminación del terminal o consola desde la cual se ejecutó el proceso.

Sintaxis

```
nohup [comando]
```

Parámetros

comando

Nombre del programa que se va a ejecutar en segundo plano

Si el programa envía algo a la salida estándar, se guardará en el fichero `nohup.out` si es posible. Si no es posible se enviará al fichero `$HOME/nohup.out`. La salida de error por defecto es el terminal, así que resulta conveniente que se envíe a un fichero utilizando un redireccionamiento.

6.4 Detención e inicio del sistema

Disponemos de varios comandos para detener el sistema, reiniciarlo o cambiar su nivel de ejecución.

6.4.1 Comando `systemctl`

El comando `systemctl` visto en la gestión de servicios también puede usarse para la detención y reinicio del sistema.

Sintaxis

```
systemctl {poweroff | reboot | rescue }
```

Parámetros

poweroff

Apagado del sistema

reboot

Reinicio del sistema

rescue

Reinicia el sistema en modo recuperación

6.4.2 Comando shutdown

Apaga o reinicia el sistema.

Sintaxis

```
shutdown [opciones] tiempo [mensaje]
```

Parámetros

tiempo

Tiempo que espera antes de realizar la parada. Puede ser:

- ✓ now.- Inmediatamente
- ✓ +m.- Número de minutos que espera antes de la parada.
- ✓ hh:mm.- Hora en formato 24h en la que se realiza la parada.

mensaje

Mensaje que se envía a los usuarios de que el sistema va a ser parado.

Opciones

-r

Realiza un reinicio del sistema.

-h

Apaga el sistema.

-c

Cancela el apagado del sistema. No toma tiempo, el siguiente argumento sería el mensaje.

-k

No apaga el sistema, solo envía el mensaje de advertencia a los usuarios y no permite que nadie vuelva a abrir sesión.

6.5 Tareas programadas

Todas las distribuciones GNU/Linux, cuentan con la posibilidad de ejecutar tareas programadas, de forma que cualquier usuario puede hacer que el sistema realice una tarea en la fecha y hora indicada o con la periodicidad señalada.

6.5.1 Comando at

El comando `at` ejecuta un comando, que lee de la entrada estándar, a la hora indicada.

Sintaxis

```
at [opciones] fecha_hora
```

Parámetros

fecha_hora

Momento en el que se ejecuta el trabajo. Puede tener el siguiente formato:

- ✓ hh:mm.- Hora y minutos en formato 24h. Si ha pasado se asume el día siguiente.
- ✓ hh:mm mmdd[yy]yy.- Fecha y hora en la que se ejecuta el trabajo.
- ✓ hh:mm mm/dd/[yy]yy.- Igual que el anterior.
- ✓ hh:mm dd.mm.[yy]yy.- Igual que el anterior.
- ✓ hh:mm [yy]yy-mm-dd.- Igual que el anterior.
- ✓ hh:mm[am|pm].- Hora en formato 12h indicando AM o PM.
- ✓ Para especificar horas se admite midnight, noon, now y teatime (4pm).
- ✓ Para especificar fechas se admite today, tomorrow y nombre-mes dia.
- ✓ Se pueden aumentar periodos de tiempo a la hora y fecha indicada con +[minutes|hours|days|weeks]

Opciones

-f fichero

Fichero de texto que contiene el comando a ejecutar. Si se omite toma el comando de la entrada estándar.

-l

Lista los trabajos pendientes. Si lo ejecuta root lista los trabajos de todos los usuarios.

-d nº

Elimina el trabajo con el número indicado.

-m

Envía un correo electrónico al usuario cuando la tarea finaliza.

6.5.2 Cron y crontab

Cron es un servicio que se encarga de ejecutar tareas en segundo plano a una determinada fecha y hora. Las tareas a ejecutar se encuentran en el fichero de texto `crontab`. Este fichero es editado por una utilidad en línea de comando con el mismo nombre y cada usuario tiene uno. También hay un fichero `crontab` de sistema para tareas que necesitan privilegios administrativos.

Cada línea del fichero `crontab` corresponde a una tarea a ejecutar y esta contiene cinco campos de fecha y hora seguidos por el comando. El `crontab` del sistema tiene un campo adicional: el usuario que ejecuta la tarea después de los campos de fecha y hora. Cada campo se separa del siguiente por un espacio o tabulador. Las líneas que comienzan por `#` son comentarios y se ignoran.

Los comandos se ejecutan por el servicio `crond` cuando el minuto, hora y mes del año coinciden con la hora actual y cuando al menos uno de los dos campos de día (día del mes o día de la semana) coinciden con la hora actual. El servicio `crond` examina los ficheros `crontab` cada minuto.

El formato de la línea es el siguiente

minuto	hora	día_mes	mes	día_semana	comando
--------	------	---------	-----	------------	---------

Los valores que se admiten son:

- ✓ Minuto.- 0-59
- ✓ Hora.- 0-23
- ✓ Día del mes.- 1-31
- ✓ Mes.- 1-12 o nombres de mes
- ✓ Día de la semana.- 0-7, 0 o 7 es domingo.

Puede indicarse en cada campo un asterisco (*) el cual es un comodín que significa cualquiera. También pueden indicarse:

- ✓ Rangos.- Dos números separados por un guión. El rango es inclusivo, es decir, si se pone 8-11 en el campo hora, la tarea se ejecuta a las 8, 9, 10 y 11 horas.
- ✓ Listas.- Conjunto de números separados por comas. Por ejemplo: 2,5,7,8 en el campo día del mes se ejecutará la tarea el día 2, 5, 7 y 8.
- ✓ Saltos en rangos.- Consiste en poner un rango y un salto separados por una barra de dividir /. Por ejemplo, si ponemos en la hora 12-23/2 significa que se ejecutará cada dos horas comenzando a las 12 y terminando a las 23. Si se quiere se puede usar * para el rango y después el salto, por ejemplo */4 en horas significa que se ejecutará la tarea cada 4 horas.

El último campo es el comando y se entiende hasta el final de la línea.

En lugar de los cinco primeros campos pueden usarse las siguientes cadenas especiales:

Cadena	Significado
@reboot	Se ejecuta una vez al arranque
@yearly	Se ejecuta una vez al año, equivale a 0 0 1 1 *
@annually	Igual al anterior
@monthly	Se ejecuta una vez al mes, equivale a 0 0 1 * *
@weekly	Se ejecuta una vez a la semana, equivale a 0 0 * * 0
@daily	Se ejecuta una vez al día, equivale a 0 0 * * *
@midnight	Igual al anterior
@hourly	Se ejecuta una vez a la hora, equivale a 0 * * * *

Por defecto los comandos se ejecutan con el intérprete de comandos /bin/sh. Si se quiere un intérprete de comandos diferente hay que indicarlo con la variable SHELL. Por ejemplo, para indicar que se desean ejecutar los comandos con el intérprete de comandos Bash pondríamos lo siguiente.

```
SHELL=/bin/bash
```

El archivo crontab de cada usuario se encuentra en

`/var/spool/cron/crontabs/login` y para editarlo se utiliza la utilidad `crontab`.

Sintaxis

```
crontab [opciones]
```

Opciones

`-u usuario`

Usuario del que se gestiona su fichero `crontab`.

`-l`

Lista los procesos actualmente en el `crontab`

`-r`

Borra el contenido del fichero `crontab`

`-e`

Edita el fichero `crontab` con el editor por defecto del sistema. Normalmente es `nano`.

El `crontab` del sistema se edita directamente por el usuario `root` y es el fichero `/etc/crontab`.

6.6 Monitor del sistema

El monitor del sistema es una herramienta gráfica que permite monitorizar el uso de los recursos del equipo por parte de los usuarios y los procesos del sistema. Se abre desde el tablero.

La información que proporciona se encuentra dividida en varias pestañas. La primera, *Procesos*, muestra la lista de los procesos actuales en ejecución.

Nombre del proceso	Usuario	% CPU	ID	Memoria	Prioridad
at-spi2-registryd	usuario	0	1359	640,0 KiB	Normal
at-spi-bus-launcher	usuario	0	1348	2,8 MiB	Normal
bamfdaemon	usuario	0	1272	8,6 MiB	Normal
compiz	usuario	0	1390	127,8 MiB	Normal
dbus-daemon	usuario	0	1219	1,3 MiB	Normal
dbus-daemon	usuario	0	1356	480,0 KiB	Normal
dconf-service	usuario	0	1412	560,0 KiB	Normal
deja-dup-monitor	usuario	0	1860	3,1 MiB	Normal
evolution-addressbook-factor	usuario	0	1641	4,8 MiB	Normal
evolution-addressbook-factor	usuario	0	1671	4,5 MiB	Normal
evolution-calendar-factory	usuario	0	1562	36,9 MiB	Normal
evolution-calendar-factory-sul	usuario	0	1593	37,4 MiB	Normal
evolution-calendar-factory-sul	usuario	0	1631	39,4 MiB	Normal
evolution-source-registry	usuario	0	1515	5,7 MiB	Normal
gnome-keyring-daemon	usuario	0	1250	736,0 KiB	Normal
gnome-session-binary	usuario	0	1394	2,0 MiB	Normal
gnome-software	usuario	0	1579	59,7 MiB	Normal
gnome-system-monitor	usuario	1	2171	14,5 MiB	Normal
gpg-agent	usuario	0	1279	268,0 KiB	Normal
gvfsafc-volume-monitor	usuario	0	1665	1,0 MiB	Normal

Figura 27.- Monitor de procesos

Si seleccionamos un proceso podemos gestionarlo haciendo clic con el botón derecho del ratón sobre él y seleccionando algunas de las operaciones que podemos ejecutar sobre un procesos:

- ✓ Detener.- Parar un proceso.
- ✓ Continuar.- Reanudar un proceso parado.
- ✓ Finalizar.- Enviar señal para finalizar el proceso.
- ✓ Matar.- Abortar un proceso.
- ✓ Cambiar la prioridad.- Modificar el valor nice del proceso.
- ✓ Mapa de memoria.- Podemos ver la memoria ocupada por el proceso.
- ✓ Archivos abiertos.- Muestra una lista con los archivos actualmente en uso por el proceso.

También disponemos en la parte inferior de la ventana el botón *Finalizar proceso* para enviarle la señal de finalización al proceso.

La siguiente pestaña nos muestra el uso de los recursos principales del sistema: CPU,

memoria y red. Para cada uno muestra el porcentaje medio de uso en el último minuto junto con un gráfico de la actividad de cada recurso también en el último minuto.

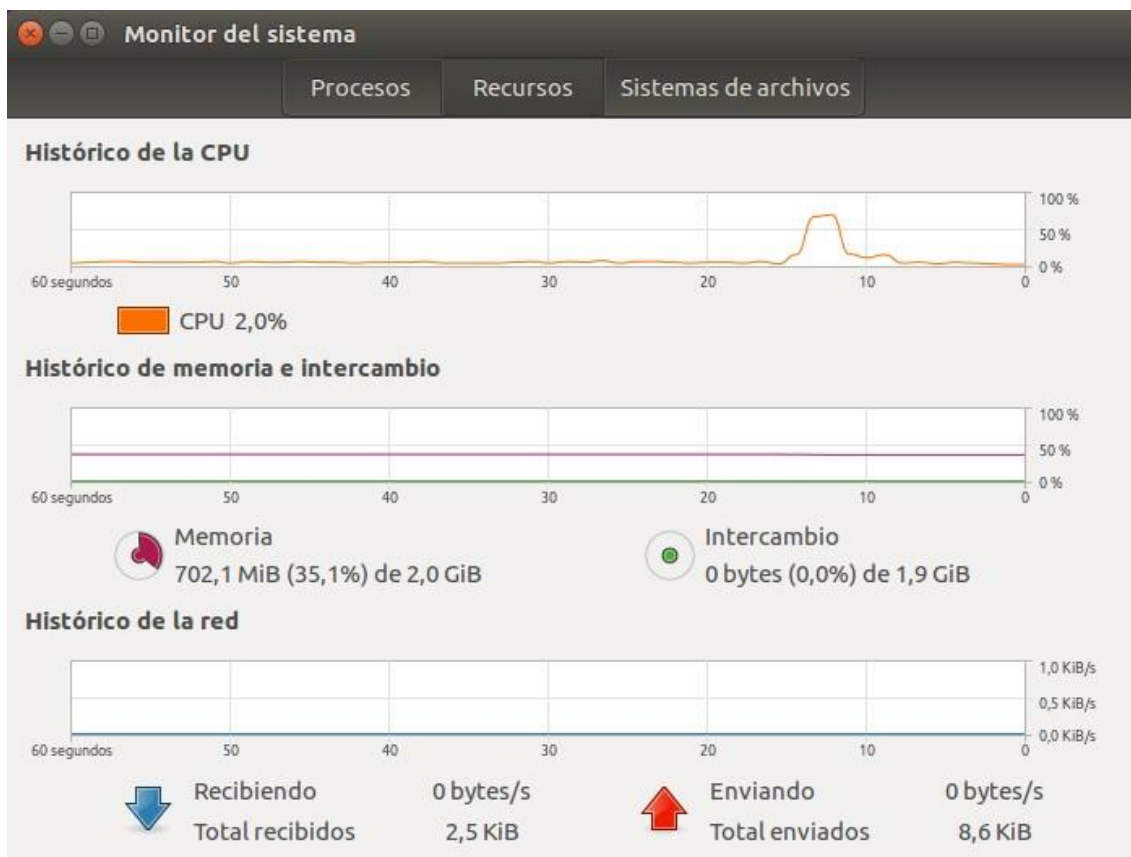


Figura 28.- Monitor de recursos

Por último, la pestaña Sistema de archivos muestra las particiones de los discos en el sistema, junto con la cantidad de espacio libre y ocupado de los mismos.

7 Servicios

Un servicio es un programa que se inicia durante el arranque del sistema operativo y que se ejecuta en segundo plano sin intervención del usuario. Su misión es proporcionar algún tipo de funcionalidad a los usuarios del sistema o a otras aplicaciones. Por ejemplo el servicio de red permite conectividad con Internet a los usuarios y programas que necesitan Internet, como descargas de archivos, juegos en red, etc. El servicio de impresión permite imprimir documentos desde cualquier aplicación.

En Linux/UNIX los servicios se prestan con un proceso **demonio**. Un demonio es un programa especial que se ejecuta en segundo plano sin interacción con el usuario. Solo se puede comunicar con él a través de canales de comunicación entre procesos.

El modelo de gestión de los servicios en un sistema operativo GNU/Linux ha sufrido cambios a lo largo de su historia. En esencia un servicio sigue siendo lo mismo: una aplicación ejecutada en segundo plano que ofrece alguna funcionalidad bajo demanda. Lo

que ha cambiado ha sido la forma de gestionarlos.

- ✓ System V → Al principio, en las primeras versiones de Ubuntu y otras distribuciones GNU/Linux se empleaba System V, en el que los servicios venían en forma de scripts ubicados en `/etc/init.d/` y se ejecutaban automáticamente invocados por el proceso `init`, el cual era el primer proceso que se ejecutaba tras la carga del kernel. Este método era heredado del sistema operativo Unix. Los scripts de inicio deservicio se invocaban a diferentes niveles de ejecución, para lo cual se tenían los enlaces simbólicos a cada script en los directorios `/etc/rc#.d`, donde `#` correspondía al número del nivel de ejecución. Así, cuando el sistema arranca en el nivel de ejecución 2, el proceso `init` invocaba los scripts que iniciaban los servicios mediante los enlaces que había en `/etc/rc2.d`.
- ✓ Upstart → A partir de la versión 6.10, Ubuntu cambia el sistema de gestión de servicios anterior por *Upstart*, aunque manteniendo la compatibilidad con System V. Este modelo es generado por eventos, lo que permite superar algunas desventajas de System V, como la conexión en caliente de ciertos dispositivos. *Upstart* permite responder a ciertos eventos de forma asíncrona cuando se generan, lo cual no era posible antes.
- ✓ Systemd → A partir de la versión 15.04, Ubuntu ha adoptado un nuevo modelo de gestión de servicios: *Systemd*. Consiste en una suite de herramientas que proveen un modelo de inicio rápido y flexible para gestionar la máquina desde el arranque hasta su apagado. *Systemd* se encarga del inicio de las tareas y servicios durante el arranque del sistema o la parada de estos servicios durante el apagado del sistema. Comienza montando todos los sistemas de archivos para posteriormente comenzar el inicio de los servicios.

7.1 Arranque del sistema

La secuencia de arranque varía de un sistema a otro pero se puede dividir básicamente en los siguientes pasos:

- ✓ Arranque del hardware.- Después de pulsar el botón de encendido se pasa el control a la BIOS. Este programa normalmente hace una comprobación básica de equipo y accede a la CMOS para leer parámetros adicionales. Los parámetros almacenados en la CMOS varían entre sistemas, pero como mínimo el programa de arranque debe saber cuál es el dispositivo de arranque, o qué dispositivos probar como posibles dispositivos de arranque. Después se accede al dispositivo de arranque, se trae a memoria el cargador del SO, que está localizado en una posición fija de este dispositivo y se le transfiere el control a éste.
- ✓ Cargador del SO.- En el PC, el cargador del SO está localizado en el primer sector del dispositivo de arranque, es el llamado MBR (*Master Boot Record*). En la mayoría de los sistemas, este cargador primario está limitado a 512 bytes incluyendo la tabla de particiones lo que hace casi imposible introducir un gestor de arranque completo dentro de él. Además, la mayoría de sistemas operativos hacen que el cargador primario llame a un cargador secundario que puede estar localizado en

una partición del disco especificada. En Linux el gestor de arranque es normalmente GRUB, el cual es un cargador en dos partes que proporciona un MBR especial que contiene el código de arranque necesario para cargar la segunda parte del cargador desde la partición raíz. La principal tarea del gestor de arranque es localizar el núcleo en el disco, cargarlo y ejecutarlo. La mayoría de gestores de arranque permiten un uso interactivo, para poder especificar un núcleo alternativo (posiblemente una copia de seguridad en caso de que el último núcleo compilado no funcione) y para pasar parámetros opcionales al núcleo.

- ✓ Puesta en marcha del núcleo.- Una vez que se carga el núcleo, éste inicializa los dispositivos (a través de sus drivers), arranca el intercambiador o swapper (es un "proceso del núcleo", llamado kswapd en los núcleos Linux modernos) y monta el sistema de ficheros raíz (/). Sólo después el núcleo crea el primer proceso llamado `init` en espacio de usuario al que asigna el PID 1. A continuación comienza el inicio de tareas básicas en el sistema.
- ✓ Tareas básicas del sistema.- Una vez el núcleo crea el primer proceso en el espacio de usuario, cede el control a éste el cual se encargará de montar el resto de los sistemas de archivos en diferentes dispositivos de almacenamiento configurados para montarse durante el arranque y pone en marcha los servicios del sistema. Para cada servicio hay un archivo ejecutable, generalmente scripts ubicados en `/etc/init.d` en la mayoría de distribuciones de Linux, aunque pueden estar en otra localización. Cada uno de estos scripts acepta como argumento la palabra `start`, que provoca el inicio del servicio durante el inicio del sistema, o la palabra `stop`, que provoca que la parada del servicio al apagar el sistema.

7.2 Systemd

Systemd es un conjunto de demonios de administración de sistema, bibliotecas y herramientas diseñados como una plataforma de administración y configuración central para interactuar con el núcleo del sistema operativo GNU/Linux. Se utiliza como un sistema de inicio de GNU/Linux, el proceso `init` llamado por el núcleo para inicializar el espacio de usuario durante el proceso de arranque y gestionar posteriormente todos los demás procesos.

El elemento básico sobre el que *systemd* actúa es conocido como *unidad* la cual se refiere a cualquier recurso que *systemd* gestiona. Este es el objeto primario con el que las herramientas de *systemd* tratan. Las unidades pueden ser, por ejemplo, servicios (`.service`), puntos de montaje (`.mount`), dispositivos (`.device`) o sockets (`.socket`).

Los *ficheros de unidad* pueden estar en diferentes ubicaciones, cada una de las cuales tienen diferentes prioridades e implicaciones. Una copia de los ficheros de unidad está en el directorio `/lib/systemd/system`. Cuando el software instala ficheros de unidad se colocan aquí por defecto. El fichero de unidad colocado aquí es el genérico que funciona en cualquier implementación de *systemd*. No es aconsejable editar estos ficheros aquí, en su lugar, si es necesario hacer alguna modificación a algún fichero de unidad, es mejor hacerlo en el que se encuentra en `/etc/systemd/system`. Los ficheros de unidad en este directorio tienen precedencia sobre cualquier otra ubicación. Cuando se necesita modificar

una copia del fichero de unidad, es mejor hacerlo sobre una copia ubicada en este directorio.

Si necesitamos sobrescribir solamente directivas específicas en un fichero de unidad, podemos utilizar un subconjunto de las mismas en un archivo dentro de un subdirectorio con el nombre del fichero de la unidad acabado en `.d`. Por ejemplo, una unidad llamada `example.service` podría tener un subdirectorio `example.service.d` dentro del cual un archivo con extensión `.conf` puede contener o sobrescribir los atributos del fichero de unidad.

Para gestionar las unidades disponemos del comando `systemctl`. Aunque las posibilidades de `systemd` son muy numerosas, vamos a estudiarlo en los siguientes apartados para la gestión de los servicios.

7.3 Arranque, parada y reinicio de servicios

La siguiente sintaxis del comando `systemctl` se emplea para gestionar los servicios

Sintaxis

```
systemctl {start | stop | restart | reload } servicio...
```

Parámetros

start

Inicia el servicio.

stop

Para el servicio.

reload

Recarga la configuración del servicio sin pararlo.

restart

Reinicia el servicio.

servicio ...

Nombre de los servicios. Pueden ser varios separados por espacios

7.4 Estado de un servicio

Además, si queremos comprobar el estado en que se encuentra un servicio podemos utilizar las siguientes acciones del comando `systemctl`.

Sintaxis

```
systemctl {is-active | is-failed | status } servicio...
```

Parámetros

is-active

Comprueba si el servicio está en ejecución

is-failed

Comprueba si el servicio falló

status <unidad>

Muestra información de estado del servicio

servicio ...

Nombre de los servicios. Pueden ser varios separados por espacios

7.5 Habilitar o deshabilitar un servicio

Si queremos habilitar o deshabilitar un servicio en el arranque del sistema podemos utilizar la siguiente forma del comando `systemctl`.

Sintaxis

```
systemctl {enable | disable } servicio...
```

Parámetros

enable

Habilita fichero de unidad o instancias de fichero de unidad. En la práctica crea los enlaces simbólicos a las unidades para que el servicio arranque al inicio del sistema.

disable

Deshabilita una unidad. Quita los enlaces simbólicos a los ficheros de unidad del directorio de configuración para que el servicio no arranque durante el inicio del sistema.

servicio ...

Nombre de los servicios. Pueden ser varios separados por espacios

Hay que recordar que `systemctl` trabaja con los ficheros `.service`, los cuales contienen la configuración para la gestión de un servicio. Cada servicio tiene un script para arrancarlo y/o pararlo, y todos los scripts de los servicios se encuentran en el directorio `/etc/init.d`. Cada fichero de `.service` tendrá una directiva que indica el script que lo inicia o para en el directorio `/etc/init.d` aunque también podría encontrarse en otro lugar.

7.6 Información de un servicio

Podemos obtener diversa información de un servicio mediante los siguientes comandos de `systemctl`.

Sintaxis

```
systemctl {list-dependencies | show | list-units | list-unit-files -type=service } servicio...
```

Parámetros

list-dependencies

Muestra el árbol de dependencias del servicio. Si se utiliza la opción `--reverse` mostrará los

servicios que dependen del especificado.

show

Muestra información completa del servicio.

list-units

Muestra una lista de unidades que se ajustan al patrón indicado. Si en lugar de un nombre de servicio como último argumento del comando indicamos el patrón `*.service` entonces solamente muestra una lista de los servicios.

list-unit-files --type=service

Muestra una lista de las unidades que son servicios. Equivale al anterior

servicio ...

Nombre de los servicios. Pueden ser varios separados por espacios

7.7 Objetivos (niveles de ejecución)

Con el modelo de gestión de servicios de *systemd* una de sus funciones es la transición del servidor entre diferentes estados. En los sistemas de inicio tradicionales estos estados eran conocidos como *niveles de ejecución* que representaba un modo específico de operación. Un sistema solo puede estar en un nivel de ejecución en un momento específico.

En *systemd* el nivel de ejecución es conocido como objetivo (*target*). Básicamente lo que hace es sincronizar los puntos que el servidor puede utilizar para estar en un estado específico. Los servicios pueden unirse a un objetivo y múltiples objetivos pueden activarse a la vez.

Sintaxis

```
systemctl {list-unit-files --type=target | get-default |  
set-default | list-dependencies | isolate } objetivo
```

Parámetros

list-unit-files --type=target

Lista las unidades del tipo objetivo.

get-default

Muestra el objetivo por defecto

set-default objetivo

Establece el objetivo por defecto

list-dependencies objetivo

Muestra las unidades vinculadas a un objetivo

isolate objetivo

Cambia el estado al objetivo indicado. Esto parará cualquier unidad que no esté unidad al objetivo especificado. Hay que asegurarse que el aislamiento no para un servicio esencial.

8 Configuración de la red

Generalmente, al configurar un PC para que pueda acceder a los recursos y servicios de una red, se comienza por la identificación del PC dentro de la red. Básicamente, la configuración de red de un ordenador en una red local con acceso a Internet implica los siguientes pasos:

1. Asignar la dirección IP al host, de forma estática o dinámica.
2. Indicar la dirección IP de la puerta de enlace o host al que se envían los paquetes que están destinados fuera de la red local.
3. Establecer un método de resolución de nombres para traducir nombres de ordenadores y/o dominio a direcciones IP.

Con estos tres pasos se dispone de un ordenador conectado a la red. Para añadir mayor funcionalidad al mismo (compartir archivos, impresoras, etc) se necesitará añadir otras herramientas.

En las siguientes secciones se discutirá la forma de configurar el adaptador de red y el resto de los parámetros de red en GNU/Linux, describiendo los pasos necesarios y desde dos enfoques distintos: con herramientas gráficas o con órdenes en línea de comando.

8.1 Interfaces de red

Los dispositivos de red en el sistema operativo se crean de forma dinámica. En la mayoría de los casos los dispositivos de red son creados automáticamente por el controlador de dispositivos mientras se inicia y localiza el hardware.

Hasta la versión 15.04 de Ubuntu, el controlador Ethernet crea interfaces `eth0`, `eth1`, `eth2` ... secuencialmente conforme el kernel las detecta. La primera interfaz que encuentra es `eth0`, la segunda `eth1`, etc. Si las tarjetas son inalámbricas entonces las nombraba como `wlan0`, `wlan1`, etc.

Sin embargo, a partir de la versión 15.10 Ubuntu utiliza una nomenclatura diferente para nombrar las interfaces de red. Consiste en un método llamado *Predictable Network Interface Names*, el cual asigna nombres a las interfaces de red con uno de los siguientes métodos y en el siguiente orden:

1. El número de índice que proporciona la BIOS/UEFI para dispositivos en placa, como por ejemplo `eno1` (Ethernet On-board 1).
2. El número de índice que proporciona la BIOS/UEFI para dispositivos en ranuras PCI Express, como por ejemplo `ens1` (Ethernet Slot 1).
3. Localización física del conector hardware, como por ejemplo `enp2s0` (Ethernet Bus n.º 2, Slot n.º 0).
4. La dirección MAC de la interfaz, como por ejemplo `enx78e7d1ea46da`. (Ethernet, la x significa que lo que viene a continuación es una dirección MAC).

5. El clásico `ethX` que proporciona el kernel, como por ejemplo `eth0`. Este método está en desuso.

¿Cómo podemos saber el nombre de dispositivo de nuestra tarjeta de red? Si ejecutamos el comando `lshw -class network` veremos todos los dispositivos hardware de nuestro PC. De cada dispositivo de red mostrará información detallada estructurada en campos. Debemos fijarnos en el campo *nombre lógico* para obtener el nombre de la interfaz de red. También podemos ejecutar el mismo comando `lshw -short` donde mostrará información de todo el hardware del PC con sus correspondientes nombres de dispositivo. Buscamos en la lista todos los que sean de clase *network* y en el campo *Dispositivo* tendremos su nombre.

En Ubuntu, podemos realizar la configuración de red mediante los siguientes métodos:

- ✓ NetPlan, que consiste en una utilidad para facilitar la configuración de las interfaces de red. Esta utilidad crea una descripción de las interfaces de red para que, posteriormente, se realice la configuración de la red mediante uno de los dos siguientes servicios:
 - Servicio `network-manager` que toma los datos de configuración de la herramienta gráfica *Network Manager*.
 - Servicio `systemd-networkd`, el cual toma los datos de un archivo de configuración de red `.network` en el directorio `/etc/systemd/network`
- ✓ Servicio `networking` que toma la configuración de la red del archivo `/etc/network/interfaces`. Hasta hace poco era el método tradicional de configuración de red.

8.2 NetPlan

Netplan es una utilidad para facilitar la configuración de la red en un sistema GNU/Linux. La descripción de las interfaces de red se crean en un archivo con formato `.yaml` en el directorio `/etc/netplan`. El archivo en sí no tiene un nombre preestablecido, sino que el propio administrador elige el nombre. En esta descripción también se establece que servicio se empleará para establecer la configuración de la red: `Network-Manager` o `systemd-networkd`.

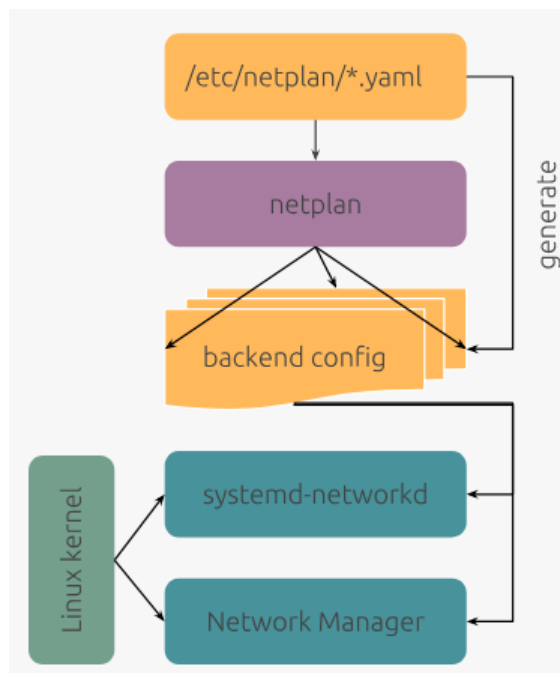


Figura 29.- Diagrama de funcionamiento de NetPlan

Vamos a ver como deben ser los archivos `.yaml` para configurar la red en función del servicio elegido para ello.

8.2.1 Configuración con Network Manager

Cuando hablamos de configurar una interfaz de red nos referimos al proceso de asignar direcciones apropiadas a un dispositivo de red y asignar valores adecuados a otros parámetros IP configurables. Existe una utilidad en formato gráfico que permite configurar la red: *Network Manager*.

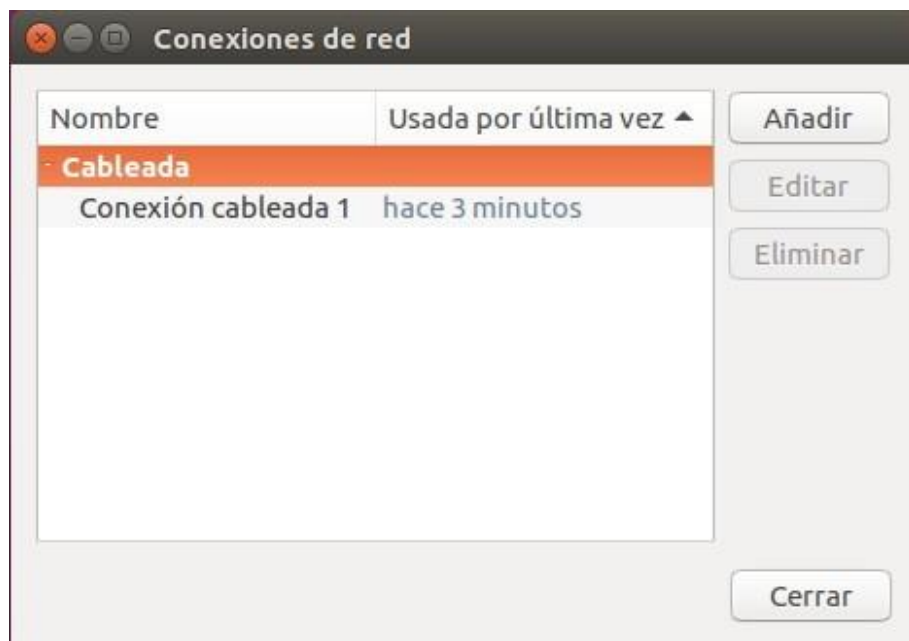


Figura 30.- Conexiones de red

Por defecto, Netplan está configurado para que la configuración de red se realice mediante NetworkManager en versiones de Ubuntu con escritorio gráfico. Esto es así porque el archivo `/etc/netplan/01-network-manager-all.yaml` tiene el siguiente contenido:

```
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
```

Como se puede apreciar, la opción `renderer` indica que NetworkManager es el encargado final de establecer la configuración de la red. Podemos acceder a la herramienta gráfica de configuración de la red mediante dos métodos. El primero de ellos consiste en hacer clic en el icono de red del panel superior del escritorio y en el menú que aparece seleccionar la opción *Cableada conectada* → *Configuración de red cableada*. El otro consiste en acceder al panel de actividades y buscar *Configuración* → *Red*.

Como se puede apreciar en la imagen aparecen todas las interfaces de red, cableadas e inalámbricas. Existirá una conexión de red por cada interfaz de red. Si queremos configurar alguna de ellas la tendremos que seleccionar y hacer clic en el botón *Editar*.



Figura 31.- Configuración de una interfaz de red

A la derecha de la interfaz de red *Cableado* tenemos el botón para configurarla. En la pestaña *Detalles* veremos la configuración actual IP junto a otros parámetros como la velocidad de conexión.

En la pestaña *IPv4* se emplea para configurar los parámetros de red. En la lista método indicaremos si la dirección IP del ordenador es *Manual* (asignamos los parámetros de red IP manualmente) o *Automático (DHCP)* (en la red hay funcionando un servidor DHCP que configura los PCs automáticamente). Si se escoge la primera opción habrá que hacer clic en el botón *Añadir* y rellenar los campos siguientes:

- ✓ Dirección IP.- Dirección IP del host.
- ✓ Máscara de subred.- Para indicar que parte de la dirección corresponde a la red y cuál al host.
- ✓ Puerta de enlace.- Dirección IP del dispositivo que se empleará para enrutar los paquetes cuyo destino está fuera de la red. Si no se indica este dato el ordenador solamente se podrá comunicar únicamente con los ordenadores de la misma red.
- ✓ Servidores DNS.- Direcciones IP de los servidores DNS que permitirán traducir nombres de dominio a direcciones IP.

The screenshot shows the 'Cableada' (Cabled) window in the Ubuntu Network Manager. The 'IPv4' tab is selected. Under 'Método IPv4', the 'Manual' radio button is chosen. The 'Direcciones' (Addresses) table contains one entry with IP '192.168.0.10', mask '255.255.255.0', and gateway '192.168.0.1'. The 'DNS' section has a toggle for 'Automático' which is turned on, and a text field containing '192.168.0.1'. The 'Rutas' (Routes) section also has an 'Automático' toggle turned on. Buttons for 'Cancelar', 'Cableada', and 'Aplicar' are at the top.

Dirección	Máscara de red	Puerta de enlace
192.168.0.10	255.255.255.0	192.168.0.1

Cuando finalicemos de configurar la conexión de red haremos clic en el botón *Aplicar...*

Si la dirección IP se va a obtener dinámicamente entonces hay que seleccionar *Automático (DHCP)* y los datos anteriores se dejan en blanco, ya que durante el arranque, el ordenador los obtiene desde un servidor DHCP.

Una vez establecida la configuración podemos verla en la pestaña *Detalles* del propio Network Manager.

8.2.2 Configuración con `systemd-networkd`

Otra opción para configurar la red es mediante el empleo del servicio `systemd-networkd` en lugar del servicio `network-manager` y su frontend Network Manager. Esta opción es la empleada en versiones de Ubuntu Server, las cuales al no tener escritorio gráfico carecen del servicio `network-manager`. Por tanto, no hay más remedio que utilizar este método.

Para ello primero debemos crear un archivo `.yaml` en `/etc/netplan` que establezca que la configuración mediante este servicio. El archivo `/etc/netplan/00-systemd-networkd.yaml` debería tener el siguiente contenido:

```
# La configuración de la red se realiza mediante systemd-networkd
```



```
network:
  version: 2
  renderer: networkd
```

Como se puede observar la opción `renderer` a cambiado a `networkd`. También podemos omitir esta opción ya que `networkd` es el valor por defecto.

A continuación debemos crear el archivo de configuración de las interfaces de red para que el servicio `systemd-networkd` tome los parámetros de configuración. Este archivo debe tener extensión `.network`. Por ejemplo, vamos a crear el archivo `/etc/systemd/network/00-red.network` en el que configuraremos la interfaz de red `enp0s3` de forma estática. Este archivo debería tener el siguiente contenido.

```
[Match]
Name=enp0s3

[Network]
Description=Tarjeta de red cableada
Address=10.0.2.35/24
Gateway=10.0.2.2
DNS=10.0.2.3
```

Como se puede ver, el archivo se divide en secciones cuyo título se encierra entre corchetes. Cada sección contiene un conjunto de opciones para establecer la configuración de la interfaz de red.

La sección `[Match]` determina la interfaz de red a la que se aplicará la configuración de la sección `[Network]`. El primer archivo en orden alfabético que contenga una interfaz de red será la configuración aplicada, si hay más archivos posteriormente se ignoran.

Podemos identificar la interfaz de red en la sección `[Match]` mediante un nombre (`Name`) o su dirección MAC (`MACAddress`) entre otros parámetros.

En la sección `[Network]` establecemos opciones para la configuración de la interfaz de red indicada en la sección `[Match]`. En el ejemplo anterior establecíamos una configuración estática para la interfaz de red. Si por el contrario, quisiéramos establecer una configuración dinámica por DHCP tendríamos que poner el siguiente contenido en el archivo.

```
[Match]
Name=enp0s3

[Network]
Description=Tarjeta de red cableada
DHCP=yes
```

Una vez hemos establecido la configuración con `NetPlan` y `systemd-networkd` la aplicamos ejecutando los siguientes comandos.

```
sudo netplan generate
```

```
sudo netplan apply
sudo systemctl restart systemd-networkd
```

En versiones Desktop de Ubuntu es necesario habilitar el servicio `systemd-networkd` para la configuración tenga efecto entre reinicios del sistema, ya que en esta versión de Ubuntu se emplea únicamente NetworkManager. Para ello ejecutamos el siguiente comando

```
sudo systemctl enable systemd-networkd
```

Prestar atención a la nomenclatura de archivos `.yaml` de NetPlan y `.network` de `systemd-networkd`. Estos archivos se aplican en orden alfabético, de ahí que se empleen un número de dos dígitos en el nombre para establecer el orden.

8.2.3 Configuración con Netplan

Por último, también podemos indicar la configuración de las interfaces de red únicamente con NetPlan. En este caso, el servicio que se empleará será `systemd-networkd` para establecer la configuración final, pero los parámetros de configuración no se establecen en un archivo `.network` de `/etc/systemd/network`, sino en el propio archivo `.yaml` de NetPlan.

Por ejemplo, podemos crear el archivo `/etc/netplan/00-red.yaml` con el siguiente contenido para establecer una configuración estática de la interfaz de red cableada.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      addresses: [10.0.2.15/24]
      gateway4: 10.0.2.2
      nameservers:
        addresses: [10.0.2.3, 8.8.8.8]
```

Es importante tener en cuenta que no puede haber un archivo `.network` en `/etc/systemd/network` ya que prevalece sobre la configuración del archivo de NetPlan.

Para una configuración dinámica podemos establecer la siguiente configuración en el mismo archivo anterior.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: true
```

Como se puede observar, si establecemos la configuración de la red directamente en NetPlan se emplea el servicio `systemd-networkd`. Por último, aplicamos la configuración

con el siguiente comando

```
sudo netplan generate
sudo netplan apply
sudo systemctl restart systemd-networkd
```

8.2.4 Precedencia en la configuración

De los métodos vistos anteriormente para configurar la red, ¿cuál elijo?. Dependerá de la versión de sistema operativo que tengamos. En general, los sistemas operativos con escritorio gráfico emplean NetworkManager por su facilidad. Por tanto no es necesario utilizar `systemd-networkd` y de hecho puede que este servicio esté deshabilitado.

Por otro lado, en sistemas operativos para servidor, los cuales no suelen tener un escritorio gráfico, tendremos que emplear `systemd-networkd`.

Hay ocasiones en que podremos encontrarnos ambos servicios funcionando. En este caso deberemos tener presente que `systemd-networkd` tiene precedencia. Por ello, si hay un archivo de configuración de red en `/etc/systemd/network` se aplicará antes de la configuración establecida con NetworkManager.

8.3 Servicio `networking`

Antes de la incorporación de NetPlan en Ubuntu 18.04 LTS Bionic, tradicionalmente, la configuración de red se realiza mediante NetworkManager, en entornos con escritorio gráfico, y el archivo `/etc/network/interfaces`. Este último ha sido reemplazado por NetPlan, pero todavía se mantiene por compatibilidad. Por tanto los administradores de red que prefieran continuar con el método antiguo de configuración de la red pueden instalar el paquete `ifupdown` y utilizar el archivo `/etc/network/interfaces`. Si este archivo tiene contenido con la configuración de red, prevalece sobre la configuración de red establecida en NetworkManager. El servicio `networking` es el encargado en este caso de gestionar la configuración de red.

8.3.1 El archivo `/etc/network/interfaces`

El servicio de red `networking` se encarga de configurar y activar las interfaces de red durante el arranque del sistema. Sin embargo, este servicio necesita información acerca de la configuración como el método de asignar la dirección IP, la dirección de la puerta de enlace, etc. Toda esta información debe estar disponible para el servicio de red en el momento de arrancar y de esta forma poder realizar la configuración automáticamente.

El archivo que contiene la información necesaria para configurar las interfaces de red es `/etc/network/interfaces`. Este archivo lo emplean los comandos `ifup` y `ifdown` para activar o desactivar las interfaces de red durante una sesión o al encender y apagar el ordenador. Este archivo tiene un formato específico que divide la configuración de la red en secciones. Cada sección hace referencia a una interfaz de red a configurar. Hay secciones para realizar la configuración y también para indicar su comportamiento durante el arranque y parada del sistema. En el caso de la red cableada las interfaces de red se identifican con los nombres establecidos por el controlador de red y que se han visto

anteriormente, es decir, mediante `eno1`, `ens1`, `enp2s0`, `eth0`, Las secciones pueden ser:

- ✓ **auto.-** Se emplea para que el script de arranque del servicio de red active la interfaz con el comando `ifup`. Las interfaces se activarán en el mismo orden en el que se establezcan.
- ✓ **iface.** Establece el método de configuración de la interfaz de red. Puede ser estático o dinámico. Si es el primero hay que indicar opciones para asignar los parámetros IP.

Sintaxis

```
auto <interfaz_de_red>  
iface <interfaz_de_red> <aftype> <metodo>
```

Parámetros

interfaz_de_red

Nombre de la interfaz. Generalmente es el nombre de controlador seguido por un número. Para la tarjeta de red Ethernet será `eno1`, `ens1`, `enp2s0`, `eth0`, etc.

aftype

Tipo de protocolo. Los posibles valores son los mismos que en el comando `ifconfig`, pero habitualmente es `inet`

metodo

Método de asignación de la dirección a la interfaz. Puede ser `static` para asignar la dirección manualmente o `dhcp` para asignar la dirección dinámicamente. Se emplea `dhcp` cuando la red se configura automáticamente por medio de un servidor DHCP.

Opciones

Las opciones a emplear dependerán del método elegido para asignar la dirección IP. Si se eligió `static` hay que indicar los siguientes parámetros:

- ✓ **address** dirección_IP (dirección IP de la máquina en la LAN): cada interfaz de red conectada a una red IP es identificada por una IP única de cuatro bytes (32 bits).
- ✓ **netmask** máscara_red (máscara de red de la LAN): es un número que establece qué parte de la IP de un host corresponde a la red y qué parte corresponde a la máquina.
- ✓ **network** dirección_red (dirección IP de la LAN): es la parte de la IP de la máquina común a todas las máquinas de la red.
- ✓ **broadcast** dirección_difusión (dirección de difusión): es la IP a la que se mandan los paquetes que deben recibir todas las máquinas de la LAN. Todas las máquinas de la red escuchan la dirección de broadcast además de la suya propia.
- ✓ **gateway** puerta_de_enlace (dirección de pasarela o puerta de enlace): es la IP del host en nuestra LAN al que se envían los paquetes destinados fuera de nuestra red. Un gateway es una máquina que tiene dos interfaces de red (una tarjeta conectada a nuestra LAN y la otra conectada a una red exterior), cada una de ellas con una IP, la que le corresponda a esa red. La IP del gateway suele ser la primera IP disponible en la red o la última.

Si el método de asignación de la dirección IP es `dhcp` no es necesario establecer ningún parámetro más, ya que estos se obtienen automáticamente. Veamos cómo se configuran las interfaces de red más habituales.

Para la interfaz de red local o `loopback`: dirección IP `127.0.0.1`. El archivo `/etc/network/interfaces` incluirá la sección:

```
auto lo
iface lo inet loopback
```

Para Ethernet o red por cable. Si la dirección IP se asigna estáticamente, el archivo `/etc/network/interfaces` incluirá líneas similares a:

```
auto enp2s0
iface enp2s0 inet static
    address 192.168.0.10
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
```

Comentemos detenidamente la configuración anterior.

```
auto enp2s0
```

La interfaz `enp2s0` se tiene que activar durante el arranque del sistema. De esta forma la red estará disponible para el usuario desde que enciende el ordenador.

```
iface enp2s0 inet static
```

Interfaz `eth0` con dirección estática (`static`) y configurada para el protocolo TCP/IP (`inet`)

```
address 192.168.0.10
```

Dirección IP `192.168.0.2`

```
netmask 255.255.255.0
```

Mascara de red `255.255.255.0`

En el caso de que la interfaz de red se configure automáticamente mediante un servidor DHCP la configuración sería como la siguiente.

```
auto enp2s0
iface enp2s0 inet dhcp
```

Como se puede apreciar, no es necesario indicar ningún parámetro de red ya que el propio servidor DHCP lo hace por nosotros.

8.3.2 Activar y desactivar las interfaces de red

Existen dos comandos que se emplean para activar y desactivar las interfaces de red.

Estos son `ifup` y `ifdown`. No es habitual desactivar una interfaz de red y la única razón para hacerlo es cambiar su configuración para posteriormente volver a activarla. Estos comandos también son utilizados por el servicio de red para activar las interfaces de red durante el arranque del sistema y desactivarlas al apagarse. Para activar todas las interfaces de red el servicio de red ejecuta lo siguiente:

```
ifup -a
```

Para modificar la configuración de alguna interfaz de red, haremos lo siguiente:

1. Desactivamos la interfaz de red (por ejemplo `enp2s0`):

```
ifdown enp2s0
```

2. Editamos `/etc/network/interfaces` y realizamos los cambios que se necesiten.
3. Activamos la interfaz de red de nuevo

```
ifup enp2s0
```

8.3.3 El comando `ifconfig`

Este comando se emplea para configurar temporalmente las interfaces de red desde un terminal de texto. Durante el arranque se configuran las interfaces de red y posteriormente solamente se emplea para cambiar esta configuración. Si no se indican parámetros visualiza el estado actual de las interfaces de red, generalmente para comprobar la configuración actual. Este comando no se encuentra por defecto y se incluye en el paquete `net-tools`.

Sintaxis

```
ifconfig [-v] [-a] [-s] [interfaz]  
ifconfig [-v] interfaz [aftype] opciones | dirección ...
```

Parámetros

interfaz

Nombre de la interfaz. Generalmente es el nombre de controlador seguido por un número. Para la tarjeta de red Ethernet será `eno1`, `ens1`, `enp2s0`, `eth0`, etc.

aftype

Tipo de protocolo. Los posibles valores son:

- ✓ `inet`.– IPv4. Este es el tipo por defecto cuando se omite este parámetro
- ✓ `inet6`.– IPv6. Versión 6 del protocolo IP
- ✓ `ipx`.– Novell IPX
- ✓ `ddp`.– Appletalk Phase 2

dirección

Dirección IP de la interfaz en notación decimal punteado

Opciones

`[-]promisc`

Habilita o deshabilita el modo promiscuo, lo que hará que todos los paquetes de la red se reciban por el interfaz

`netmask mascara`

Establece la máscara de red para la interfaz

`broadcast direccion`

Establece la dirección de broadcast para la interfaz

`up`

Activa la interfaz. Por defecto la interfaz se activa si se le asigna una dirección IP

`down`

Desactiva la interfaz. No se puede enviar ni recibir tráfico por esta interfaz

`-a`

Visualiza todas las interfaces que están actualmente disponibles, aunque no estén activadas.

8.4 Interfaz inalámbrica

Cuando disponemos en el equipo de una interfaz de red inalámbrica debemos configurarla de forma algo diferente. En principio si el equipo cuenta con escritorio gráfico la configuración se haría a través del applet de Network Manager, sin embargo, para un servidor deberemos utilizar Netplan y el servicio `systemd-networkd`. En este caso hay que utilizar también el servicio `wpa-suppllicant` que proporciona acceso a redes inalámbricas con seguridad WPA.

Inicialmente debemos instalar los siguientes paquetes:

```
sudo apt install wpasupplicant wireless-tools
```

Ahora debemos configurar el servicio `wpasupplicant` para la interfaz de red inalámbrica. Supongamos que disponemos de una interfaz `wlp3s0` y que vamos a conectarnos a una red inalámbrica con SSID `miwifi` y clave `clavewifi`. Primero creamos el archivo `/etc/wpa_supplicant/wpa_supplicant-wlp3s0.conf`. Este archivo debe tener en su nombre el nombre de la interfaz inalámbrica que queremos configurar y lo podemos generar con el siguiente comando.

```
sudo bash -c "wpa_passphrase miwifi clavewifi >>
/etc/wpa_supplicant/wpa_supplicant-wlp3s0.conf"
```

Si vemos el contenido de este archivo veremos lo siguiente:

```
network={
    ssid="miwifi"
    #psk="clavewifi"

    psk=7933c9e4b083d00937c418b1e498d573cbd3baca4829cc67c7a1085c
```

```
6ca06942
}
```

El comando anterior deberemos repetirlo por cada red inalámbrica conocida a la que nos podemos conectar, creando así en el archivo `/etc/wpa_supplicant/wpa_supplicant-wlp3s0.conf` todas las posibles redes inalámbricas a las que nos podemos conectar.

Ahora debemos generar el archivo de configuración del servicio `/lib/systemd/system/wpa_supplicant@wlp3s0.service`, el cuál tendrá el siguiente contenido.

```
[Unit]
Description=WPA supplicant daemon (interface-specific
version)
Requires=sys-subsystem-net-devices-%i.device
After=sys-subsystem-net-devices-%i.device

[Service]
Type=simple
ExecStart=/sbin/wpa_supplicant
-c/etc/wpa_supplicant/wpa_supplicant-%I.conf -i%i

[Install]
Alias=multi-user.target.wants/wpa_supplicant@%i.service
```

Finalmente hay que habilitar el servicio `wpa_supplicant` para que se inicie durante el arranque del sistema operativo.

```
sudo systemctl enable wpa_supplicant@wlp3s0.service
```

Una vez creado el servicio podemos ya incluir la configuración de la interfaz de red inalámbrica en nuestro servicio de red. La forma de hacerlo dependerá del servicio que estemos utilizando y se verá en los siguientes apartados.

8.4.1 Netplan y servicio `systemd-networkd`

A continuación vamos a ver como configurar la interfaz de red inalámbrica si disponemos del servicio `systemd-networkd`. Inicialmente consideraremos que no vamos a utilizar el backend Netplan, por lo que tendremos que generar un archivo de configuración `.network`.

Si empleamos un archivo de configuración `.network` debemos crear un archivo `/etc/systemd/network/10-wifi.network` con el siguiente contenido:

```
[Match]
Name=wlp3s0

[Network]
DHCP=ipv4
```



```
[DHCP]
RouteMetric=20
```

En este caso la interfaz obtendrá su dirección IP mediante DHCP. En el caso de tener también un archivo de configuración `.network` para la interfaz de red cableada establecer el parámetro `RouteMetric` en la sección `DHCP` con valor 10. Si queremos que sea asignada estáticamente el contenido del archivo anterior sería el siguiente:

```
[Match]
Name=wlp3s0

[Network]
Description=Tarjeta de red inalámbrica
Address=10.0.2.35/24
Gateway=10.0.2.2
DNS=10.0.2.3
```

Si queremos utilizar Netplan, entonces el archivo de configuración `/etc/netplan/10-wifi.yaml` podría tener el siguiente aspecto.

```
network:
  version: 2
  renderer: networkd
  wifis:
    wlp3s0:
      dhcp4: yes
      dhcp6: no
      access-points:
        "miwifi":
          password: "clavewifi"
```

En el caso de que la dirección IP se configure estáticamente, el archivo será el siguiente:

```
network:
  version: 2
  renderer: networkd
  wifis:
    wlp3s0:
      dhcp4: no
      dhcp6: no
      addresses: [192.168.0.21/24]
      gateway4: 192.168.0.1
      nameservers:
        addresses: [192.168.0.1, 8.8.8.8]
      access-points:
        "miwifi":
          password: "clavewifi"
```

8.4.2 Servicio networking

Generalmente los PCs conectados a una red inalámbrica necesitan una clave para asociarse a un punto de acceso inalámbrico. En este caso hay que añadir algunos parámetros adicionales para indicar la red inalámbrica a la que nos conectamos y la clave utilizada. Actualmente, las interfaces de red inalámbricas se configuran con seguridad WPA, habiéndose abandonado la seguridad WEP debido a su fragilidad.

Primero tendremos que averiguar la interfaz de red que el sistema ha reconocido. Según la nueva nomenclatura para las interfaces de red comenzará por `wl` seguido del número de dispositivo en placa o en ranura de expansión. Ejecutando el comando `sudo lshw -c network` podremos saberlo con seguridad. En el siguiente ejemplo supondremos que es `wlp3s0`. Si necesitamos establecer los parámetros DHCP el fichero `/etc/network/interfaces` tendrá una sección como la siguiente.

```
auto wlp3s0
iface wlp3s0 inet dhcp
    wpa-ssid NOMBRE_RED
    wpa-psk CLAVE
```

Como se puede observar hay que indicar el nombre de la red inalámbrica a la que nos conectamos (ESSID) con el parámetro `wpa-ssid` y la clave con `wpa-psk`. Conviene que a partir de entonces el fichero `/etc/network/interfaces` tenga máscara de permisos 600 para evitar que cualquier usuario puede leerlo y ver la clave.

En el caso de que la configuración de la red sea estática, entonces hay que añadir el resto de parámetros de red. Siguiendo con nuestro ejemplo:

```
auto wlp3s0
iface wlp3s0 inet static
    wpa-ssid NOMBRE_RED
    wpa-psk CLAVE
    address 192.168.0.10
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
```

8.5 Comando ip

Durante muchísimo tiempo se ha utilizado el comando `ifconfig` para realizar tareas relacionadas con la red, como verificar interfaces de red o configurarlas. Pero `ifconfig` ya no se mantendrá y ha quedado en desuso en las versiones recientes de Linux. Siendo remplazado por el comando `ip`.

El comando `ip` es bastante similar al comando `ifconfig` pero es mucho más potente, con muchas más funcionalidades asociadas. Puede realizar varias tareas que no eran posibles de realizar con el comando `ifconfig`.

Sintaxis

```
ip [ OPCIONES ] OBJETO { COMANDO | help }
```

Opciones

-V, -version

Muestra la versión de ip.

-s, -stats, -statistics

Muestra información estadística. Esta opción puede aparecer varias veces para incrementar la cantidad de información a mostrar.

-f, -family

Seguida por un protocolo: inet, inet6 o link, fuerza a utilizar dicho protocolo. Si no se indica, el protocolo se obtiene a través de otros argumentos.

-4

Abreviatura para -family inet.

-6

Abreviatura para -family inet6.

-0

Abreviatura para -family link.

-o, -oneline

Muestra cada registro en una línea simple.

-r, -resolve

Utiliza el sistema DNS para visualizar nombres en lugar de direcciones.

Parámetros

OBJETO

Especifica sobre que elemento de red actúa el comando. Puede ser:

- ✓ link → Para configurar dispositivo de red.
- ✓ addr → Para configurar las direcciones IPv4 e IPv6.
- ✓ addrlabel → Para gestionar una etiqueta a la dirección IPv6
- ✓ route → Para gestión de la tabla de enrutamiento.
- ✓ rule → Para gestión de la política de enrutamiento.
- ✓ neighbour → Para gestión de la tabla ARP
- ✓ tunnel → Para gestión de tráfico encapsulado.
- ✓ maddr → Gestión de direcciones multicast.
- ✓ mroute → Gestión del enrutamiento multicast.
- ✓ monitor → Para monitorización de dispositivos, direcciones y rutas.

COMANDO

Especifica la acción a realizar sobre el objeto. El conjunto de posibles acciones depende del tipo de objeto. Como regla general, es posible añadir (add), borrar (delete) y mostrar (show) o listar (list) objetos, pero algunos objetos no permiten todas estas operaciones y

tienen comandos adicionales. La ayuda del comando (help) está disponible para todos los objetos. Visualiza una lista de comandos disponibles y argumentos.

Si no se incluye ningún comando, algún comando por defecto se asuma. Normalmente list y si el tipo de objeto no admite list, entonces help.

El número de objetos y comandos hace crecer exponencialmente las posibilidades del comando `ip`. Para realizar alguna opción concreta se recomienda consultar la página de manual del comando. Aquí describiremos brevemente las más habituales.

8.5.1 Comprobar información de las interfaces de red

Para verificar la información de red como la dirección IP, subred, etc. para las interfaces, usamos el siguiente comando.

```
usuario@U2004:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ea:6a:46 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic
noprofixroute enp0s3
    valid_lft 81598sec preferred_lft 81598sec
    inet6 fe80::1187:b3ad:8ef:9dc2/64 scope link
noprofixroute
    valid_lft forever preferred_lft forever
```

Vemos en la salida del comando que numera las interfaces de red y por cada una muestra información de la configuración como su dirección IPv4, dirección MAC, dirección IPv6, etc.

En el caso de que quisiéramos ver esta misma información sobre una interfaz de red la podemos añadir como argumento.

```
usuario@U2004:~$ ip addr show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ea:6a:46 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic
noprofixroute enp0s3
    valid_lft 81459sec preferred_lft 81459sec
    inet6 fe80::1187:b3ad:8ef:9dc2/64 scope link
noprofixroute
    valid_lft forever preferred_lft forever
```

8.5.2 Habilitar o deshabilitar las interfaces de red

Para habilitar una interfaz de red deshabilitada usamos el siguiente comando

```
usuario@U2004:~$ sudo ip link set enp0s3 up
```

Para desactivar la interfaz de red usaremos el mismo comando anterior con el argumento down.

```
usuario@U2004:~$ sudo ip link set enp0s3 down
```

8.5.3 Configurar las interfaces de red

Para configurar los parámetros IP a una interfaz de red utilizaremos varios comandos para establecer cada parámetro IP.

```
usuario@U2004:~$ sudo ip address add 10.0.2.25/24 dev enp0s3
```

La máscara de red se podría haber puesto en notación decimal punteado. También podemos configurar la dirección de difusión a la interfaz. De manera predeterminada, no se establece ninguna dirección de difusión, por lo que debemos ejecutar el siguiente comando:

```
usuario@U2004:~$ sudo ip address add broadcast 10.0.2.255  
dev enp0s3
```

También podemos establecer la dirección de difusión estándar junto con la dirección IP utilizando el siguiente comando:

```
usuario@U2004:~$ sudo ip address add 10.0.2.25/24 broadcast  
+ dev enp0s3
```

Los comandos para añadir direcciones IPv4 hacen que se asigne a una interfaz más de una dirección IP. Para eliminar la dirección IPv4 asignada en la interfaz ejecutamos el siguiente comando.

```
usuario@U2004:~$ sudo ip address delete 10.0.2.26/24 dev  
enp0s3
```

Para añadir una puerta de enlace ejecutamos el siguiente comando:

```
sudo ip route add default via 10.0.2.1 dev enp0s3
```

8.6 Resolución de nombres

Se conoce como resolución de nombres al proceso de traducir un nombre a una dirección IP con el objetivo de acceder a hosts en la red de manera más fácil por un nombre en lugar de por su dirección IP.

Ya sabemos que el nombre asignado a un dispositivo se conoce como *hostname*. En GNU/Linux no se contempla el uso de nombre NetBIOS, salvo en aquellas herramientas y utilidades basadas en el protocolo SMB (Samba). Cada PC tiene un nombre que se puede visualizar ejecutando el comando `hostname` sin argumentos. Este nombre se almacena

además en el fichero `/etc/hostname`.

El término resolución de nombres no hace referencia a una aplicación específica, sino a un conjunto de funciones que emplean las aplicaciones cuando necesitan traducir nombres. Cuando se las invoca, las funciones de resolución de nombres leen los archivos de configuración. A partir de estos archivos, determinan qué bases de datos de nombres deben consultar, fichero de resolución local o sistema DNS, en qué orden y otros detalles relevantes para la configuración de la resolución de nombres.

Básicamente, el sistema de resolución de nombres en un host de GNU/Linux está formado por:

- ✓ Resolución de nombres local → El archivo `/etc/hosts` contiene una lista con hostnames y sus correspondientes direcciones IP para traducir cualquier consulta de estos hostnames a su dirección IP rápidamente.
- ✓ Resolución por DNS → El servicio `systemd-resolved` forma parte del sistema `systemd` y es el encargado de enviar las solicitudes de traducción de las aplicaciones a los servidores DNS configurados en el equipo.
- ✓ Archivo `/etc/nsswitch.conf` → Este archivo permite configurar el orden de uso de las dos métodos de resolución de nombres anteriores.

Vamos a ver en detalle cada uno de ellos.

8.6.1 Archivo `/etc/hosts`

Este archivo contiene habitualmente información para resolver nombres de hosts dentro de la red local. No es habitual encontrarnos información con nombres de servidores en Internet, aunque podría haberla. En muchas ocasiones se opta por este tipo de resolución debido a su rapidez ya que la información reside en el mismo host que está solicitando la resolución, sin embargo, una de las desventajas que presenta este archivo es que debe ser mantenido manualmente, y en redes de tamaño mediano o grande el mantener este archivo actualizado en todos los PC's de la red presenta una carga administrativa excesiva.

Este archivo es de texto y mantiene una tabla con los hostnames y sus correspondientes direcciones IP. Tiene una línea por cada host y su formato es muy simple. Consiste en la dirección IP del host y el hostname separados por un espacio o tabulación. El fichero puede tener el siguiente aspecto:

```
127.0.0.1    localhost
192.168.1.11 pc01.dominio.es  pc01
```

Las líneas anteriores tienen el siguiente formato:

```
direccion_IP nombre_del_host_completo  [alias ...]
```

El alias como su nombre indica es para evitar utilizar el nombre completo del host (FQDN *Fully Qualified Domain Name*), es decir, en el ejemplo anterior `pc01.dominio.es` es lo mismo que `pc01`.

Para que este sistema funcione se tendrá en cuenta que conviene tener direcciones IP estáticas en los hosts, ya que si son dinámicas (asignadas por un servidor DHCP) podrían cambiar y la resolución de nombres estaría corrupta.

8.6.2 Servicio `systemd-resolved`

Cuando se desea acceder a hosts que están fuera de la red local hay que complementar el sistema anterior con otro que permita resolver su nombre. El método habitual es emplear un servidor DNS (*Domain Name System*) que es un servicio ejecutándose en un host cuya tarea es traducir nombres a direcciones IP.

Desde la versión 16.10 de Ubuntu hay un servicio denominado `systemd-resolved`, que se encarga de realizar la resolución de nombres en DNS. Este servicio forma parte del sistema de gestión de servicios `systemd`, el cual vimos anteriormente.

Este servicio utiliza el archivo `/run/systemd/resolve/stub-resolv.conf`, el cual reemplaza al tradicional archivo `/etc/resolv.conf`. De hecho, éste último es ahora un enlace simbólico a `stub-resolv.conf`.

Este fichero contiene las direcciones IP de los servidores DNS que se emplearán para traducir los nombres de dominio a direcciones IP. Este archivo no se debe editar manualmente ya que se genera automáticamente por el servicio `systemd-resolved` a partir de la configuración DNS que obtiene del correspondiente parámetro en el archivo de configuración de red que puede ser:

- ✓ Archivo `.network` si la configuración es con `systemd-networkd`
- ✓ Archivo `.yaml` si la configuración es con `Netplan`
- ✓ Desde el applet gráfico de `Network-Manager`).

Podemos remitirnos a los epígrafes anteriores para ver como configurar los servidores DNS en una interfaz de red.

Independientemente del método elegido para configurar la red y de las direcciones IP de los servidores DNS configurados por ende, el servicio `systemd-networkd` configura como dirección IP del servidor DNS 127.0.0.53. Todas las consultas DNS se envían por tanto al propio host, donde posteriormente se derivará a la dirección del servidor DNS configurado. Para saber cuál será este podemos ejecutar el siguiente comando

```
systemd-resolve --status
...
Link 2 (enp0s3)
    Current Scopes: DNS
    LLMNR setting: yes
MulticastDNS setting: no
    DNSSEC setting: no
    DNSSEC supported: no
    DNS Servers: 10.0.2.3
    DNS Domain: homestation
```

Al final de la salida del comando aparece la configuración DNS para cada interfaz de red.

En el caso de que queramos una configuración manual estática podemos generar un archivo `resolv.conf` propio y modificar el enlace simbólico `/etc/resolv.conf` para que apunte a nuestro archivo. En este caso debemos de conocer las opciones y sintaxis de este archivo para realizar correctamente la configuración de la resolución de nombres.

Introduciremos la dirección IP de hasta tres servidores DNS, pero nunca menos de dos, ya que si uno de ellos falla se realizaría la consulta al otro. Cada servidor se indica con la opción `nameserver`, que indica la dirección IP del servidor de nombres a usar. Si especificamos varios servidores de nombres usamos varias veces la opción `nameserver`. Por tanto, deberíamos colocar el servidor más fiable en primer lugar.

Otras dos opciones, `domain` y `search`, nos permiten usar nombres abreviados para los equipos de nuestro dominio local. Normalmente, cuando sólo contactamos con otro equipo de nuestro dominio local, no queremos escribir el nombre de host completamente cualificado, sino usar un nombre en la línea de comandos y hacer que el sistema de resolución añada el dominio.

Para esto sirve la opción `domain`. Nos permite especificar el nombre de dominio predeterminado que se va a añadir cuando DNS no consiga encontrar un nombre de host. Por ejemplo, cuando se proporcione el nombre `pc01`, el sistema de resolución de nombres intentará primero encontrar `pc01` en DNS y no lo conseguirá porque no existe este dominio de nivel superior. Cuando se proporciona `dominio.es` como dominio predeterminado, el sistema de resolución de nombres repetirá la consulta para `pc01`, añadiendo el dominio predeterminado, y ahora buscará `pc01.dominio.es`.

Aunque este sistema parece perfecto, tan pronto como salimos del dominio `dominio.es` volveremos a necesitar los nombres de dominio completos. Supongamos que manejamos más de un dominio en nuestra red. Aquí es donde entra en juego la lista de búsqueda. Podemos especificar una lista de búsqueda usando la opción `search`, que es una generalización de la instrucción `domain`. Mientras que esta última proporciona un solo dominio predeterminado, la primera especifica toda una lista de ellos, y cada uno se probará por orden hasta que se consiga un resultado para la búsqueda. La lista debe estar separada por espacios o tabulaciones.

Las instrucciones `search` y `domain` son mutuamente exclusivas y no pueden aparecer más de una vez. Si no se proporciona ninguna de estas opciones, el sistema de resolución de nombres intentará averiguar el dominio predeterminado a partir del nombre de host local.

Por ejemplo, podría tener el siguiente aspecto

```
search dominio.es
nameserver 195.235.96.90
nameserver 195.235.113.3
```

En este archivo se indican dos servidores DNS con la opción `nameserver` y una

cadena de búsqueda `dominio.es`.

8.6.3 Archivo `/etc/nsswitch.conf`

El archivo `/etc/nsswitch.conf` permite al administrador del sistema configurar una gran variedad de bases de datos. Nos limitaremos a las opciones relacionadas con la resolución de nombres. Las opciones para `nsswitch.conf` deben aparecer en líneas separadas. Los campos pueden estar separados por espacios en blanco o tabuladores. Las líneas de comentario comienzan con `#`. Cada línea describe un servicio determinado; la resolución de nombres de hosts es uno de ellos. El primer campo de cada línea es el nombre de la base de datos, finalizando con el signo de dos puntos. El nombre de la base de datos asociada a la resolución de nombres de máquinas es `hosts`. El resto de cada línea almacena opciones que determinan cómo se realizan las búsquedas en esa base de datos.

El orden en el que aparecen los servicios que se van a consultar determina el orden en el que se consultarán cuando se intente resolver un nombre. Los servicios se consultan de izquierda a derecha y, por defecto, la búsqueda se detiene cuando se consigue una resolución.

Para la resolución de hosts disponemos de la siguiente línea

```
hosts:      files mdns4_minimal [NOTFOUND=return] dns
```

- ✓ `files` → Primero intenta resolver nombre mediante el archivo `/etc/hosts`.
- ✓ `mdns4_minimal` → Intenta resolver el nombre mediante Multicast DNS.
- ✓ `[NOTFOUND=return]` → Significa que cualquier respuesta `notfound` devuelta por el método precedente `mdns4_minimal` debería ser tratada como autoritativa y el sistema no debería intentar continuar buscando una respuesta.
- ✓ `dns` → Representa una consulta unicast DNS.

Para modificar el orden de los métodos de resolución solamente hay que cambiar las opciones anteriores en `host`. Por ejemplo, si preferimos usar consultas unicast DNS antes que consultas multicast DNS debemos escribir la línea anterior como sigue:

```
hosts:      files dns [NOTFOUND=return] mdns4_minimal  
mdns4
```

8.6.4 Resolución de nombres en `/etc/network/interfaces`

Ya hemos visto anteriormente que el archivo `/etc/network/interfaces` para configurar las interfaces de red cuando empleamos el servicio `networking` en lugar de los más actualizados `NetPlan` y `systemd-networkd`. Si la interfaz de red se configura estáticamente podemos editar manualmente el archivo `/etc/resolv.conf` para establecer la información de los servidores DNS para resolver nombres. Sin embargo, resulta más adecuado establecer esta información directamente en el archivo de configuración de red. Consiste en añadir las siguientes opciones a la configuración de una interfaz de red:

```
dns-search dominio.es  
dns-nameservers 195.235.96.90 195.235.113.3
```

Como se puede deducir, el primero es equivalente la opción `search nombre_dominio` en el archivo `/etc/resolv.conf` mientras que el segundo equivale a la opción `nameserver` del mismo archivo.

Estas opciones son ignoradas cuando la configuración de la interfaz de red es dinámica, por servidor DHCP, ya que es habitual que este tipo de servidores además de establecer la dirección IP de la interfaz también establece la puerta de enlace y los servidores DNS.

9 Bibliografía

GONZALEZ Durán, S. *Manual Básico de Administración de Procesos* [acceso julio 2010]. Disponible en <http://www.linuxtotal.com.mx/index.php?cont=info_admon_012>

JUANETABITEL, *Software y actualizaciones en Ubuntu 14.04*, [accedido el 1 de Enero de 2015]. Disponible en <http://www.ubuntu-guia.com/2014/04/software-y-actualizaciones-en-ubuntu.html>

LALLINAHU, P. *CompileEasyHowTo* [acceso enero 2015]. Disponible en <<https://help.ubuntu.com/community/CompilingEasyHowTo>>

ELLINGWOOD, J. *How To Edit the Sudoers File on Ubuntu and CentOS* [acceso marzo 2019]. Disponible en <<https://www.digitalocean.com/community/tutorials/how-to-edit-the-sudoers-file-on-ubuntu-and-centos>>

DE LA LUZ, S. *Configura la política de contraseñas en Debian y Ubuntu para proteger más tu equipo* [acceso marzo 2019]. Disponible en <<https://www.redeszone.net/2016/09/13/configura-la-politica-contrasenas-debian-ubuntu-proteger-mas-equipo/>>

HENRY-STOCKE, S. *The complexity of password complexity* [acceso marzo 2019]. Disponible en <<https://www.networkworld.com/article/3198444/the-complexity-of-password-complexity.html>>

BOELEN, M. *Locking users after X failed login attempts with pam_tally2* [acceso marzo 2019]. Disponible en <https://linux-audit.com/locking-users-after-failed-login-attempts-with-pam_tally2/>

YERALDINE *Cómo usar el comando IP (el sustituto de ifconfig en Debian 9)* [acceso marzo 2019]. Disponible en <<https://ayudalinux.com/comando-ip/>>