

# Gestión de archivos y directorios en Ubuntu 20.04 LTS

---

En este capítulo veremos en detalle la gestión del almacenamiento en Ubuntu. Comenzaremos por introducir la estructura del sistema de archivos en Linux. A continuación veremos el montaje de los dispositivos de almacenamiento y después el gestor de archivos en el entorno gráfico. Posteriormente veremos los comandos de gestión de archivos y acabaremos con los permisos y ACLs de archivos.

---



*Gestión de archivos y dispositivos 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 al almacenamiento en GNU/Linux.....	1
1.1	Estructura.....	2
1.2	Extensión en los nombres de archivo.....	4
2	Introducción al uso de comandos.....	5
2.1	El intérprete de comandos.....	5
2.2	Consolas de texto.....	5
2.3	Ventana de terminal.....	6
2.4	Usuario administrador y usuario root.....	6
2.5	Documentación de comandos.....	7
2.5.1	Documentación de Ubuntu.....	7
2.5.2	Manual universal de comandos: man.....	8
2.5.3	Descripción de comandos.....	12
2.5.4	Búsqueda de comandos.....	12
2.6	Comandos iniciales.....	13
2.6.1	Mostrar identificador de usuario.....	13
2.6.2	Mostrar directorio activo.....	14
2.6.3	Mostrar el histórico de comandos.....	15
2.6.4	Limpiar la pantalla.....	15
2.6.5	Información de la conexión de red.....	15
2.6.6	Mostrar la fecha del sistema.....	16
2.6.7	Muestra información del sistema.....	17
2.6.8	Mostrar nombre del equipo.....	17
2.7	Histórico de comandos.....	17
3	Montaje de dispositivos de almacenamiento.....	18
3.1	Identificación de dispositivos.....	19
3.2	Montaje y desmontaje con la GUI.....	20
3.2.1	Unidades ópticas.....	20
3.2.2	Memorias flash.....	20
3.3	Montaje y desmontaje en CLI. Comandos mount y umount.....	21
3.3.1	Opciones de montaje.....	22
3.3.2	Montaje de dispositivos de almacenamiento externos.....	22
3.3.3	Desmontaje.....	24
3.3.4	Configuración estática del montaje. Fichero /etc/fstab.....	25
4	Archivos.....	27
4.1	La ventana de Archivos.....	27
4.1.1	Botón menú.....	29
4.1.2	La barra de herramientas.....	29
4.1.3	El panel lateral.....	30
4.1.4	Barra de lugar.....	31
4.1.5	Panel de contenido.....	31
4.2	Gestión de archivos y carpetas.....	32

4.2.1	Crear una carpeta.....	32
4.2.2	Abrir un archivo.....	32
4.2.3	Selección de archivos.....	33
4.2.4	Copiar archivos.....	34
4.2.5	Mover archivos.....	36
4.2.6	Eliminar archivos.....	37
4.2.7	Renombrar archivos.....	38
4.2.8	Propiedades de archivos y carpetas.....	38
4.3	Archivos ocultos.....	40
4.3.1	Ocultar un archivo o carpeta.....	40
5	Búsquedas de archivos.....	40
5.1	Búsqueda con Archivos.....	40
5.2	Búsqueda con el botón aplicaciones.....	42
6	La papelera de reciclaje.....	42
6.1	Restaurar elementos eliminados.....	44
6.2	Eliminar permanentemente.....	44
6.3	Vaciar la papelera.....	44
7	Almacenamiento en la nube.....	45
7.1	Google Drive.....	45
7.2	DropBox.....	47
8	Gestión de archivos y directorios en CLI.....	47
8.1	Metacaracteres en los nombres de archivo.....	49
8.2	Crear directorios. Comando mkdir.....	49
8.3	Cambiar el directorio activo. Comando cd.....	50
8.4	Eliminar directorios. Comando rmdir.....	51
8.5	Listar el contenido de un directorio. Comando ls.....	51
8.6	Copiar archivos y directorios. Comando cp.....	52
8.7	Mover archivos y directorios. Comando mv.....	53
8.8	Eliminar archivos y directorios. Comando rm.....	54
8.9	Mostrar el contenido de archivos de texto. Comandos cat y more.....	54
8.10	Búsqueda de archivos y directorios. Comando find.....	55
8.11	Compresión de archivos. Utilidad gzip.....	56
8.12	Empaquetado de archivos. Comando tar.....	57
8.13	Enlaces simbólicos. Comando ln.....	58
8.14	Alias.....	59
9	Control de acceso.....	60
9.1	Usuario propietario y grupo propietario de un archivo.....	60
9.2	Tipos de permisos.....	61
9.2.1	Permisos de lectura.....	62
9.2.2	Permiso de escritura.....	63
9.2.3	Permiso de ejecución.....	63
9.2.4	¿A quién se puede otorgar permisos?.....	64
9.2.5	Visualizar los permisos de archivos y carpetas.....	65
9.3	Cambio de permisos. Comando chmod.....	66

9.4 Permisos especiales: el bit Sticky, SUID, SGID.....	69
9.4.1 El bit Sticky.....	69
9.4.2 El bit SUID.....	69
9.4.3 El bit SGID.....	69
9.4.4 Asignar bit Sticky, SUID, SGID.....	70
9.5 Gestión de permisos en formato gráfico.....	70
9.6 Cambio de propietario. Comando chown.....	72
9.7 Máscara de permisos. Comando umask.....	73
9.8 ACLs.....	73
9.8.1 Habilitar ACL en un sistema de archivos.....	74
9.8.2 Listar permisos. Comando getfacl.....	74
9.8.3 Asignar permisos en una ACL. Comando setfacl.....	77
10 Bibliografía.....	85

# Gestión de archivos en Ubuntu 20.04 LTS

## 1 Introducción al almacenamiento en GNU/Linux

---

GNU/Linux emplea el sistema de archivos ext4 para organizar el almacenamiento de la información. Además, también puede reconocer y utilizar multitud de sistema de archivos como FAT32 y NTFS, tanto para particiones del disco duro como para dispositivos extraíbles.

Linux utiliza una estructura de árbol para organizar el almacenamiento. Esta estructura mantiene algunas diferencias respecto a Windows. En lugar de ver un volumen por partición con su árbol de almacenamiento asociado, Linux ve todo el almacenamiento con un único árbol de almacenamiento a partir del cual se puede acceder a toda la información almacenada en todos los dispositivos de almacenamiento del equipo: particiones de disco duro, medios extraíbles y carpetas de red.

En las distribuciones Linux el sistema de archivos que se emplea es el ext4, una mejora del ext3, el cual sigue siendo compatible. Es un sistema de archivos transaccional, lo que significa que tiene la posibilidad de recuperarse ante paradas abruptas de un sistema informático por corte eléctrico, avería o fallo del sistema operativo. En lo que respecta al sistema de archivos, solamente se trata de mantener en estado consistente la estructura que permite almacenar y recuperar la información, principalmente lo que se refiere a directorios y descriptores de archivos, pero no la información almacenada en los archivos. De esta forma se evita el engorroso y largo chequeo de disco que realizan los sistemas operativos al reiniciarse el sistema después de apagarse bruscamente.

Algo común a todos los sistemas de archivos son los elementos en los que se basa y que se emplean tanto para almacenar como para organizar y recuperar la información. Estos elementos son los directorios (carpeta) y los archivos

## 1.1 Estructura

En GNU/Linux y sistemas tipo UNIX, la información se guarda en archivos y los archivos se organizan en directorios. Todos los archivos y directorios se distribuyen en un gran árbol formando la jerarquía de almacenamiento comenzando al más alto nivel por la raíz `/`. Los archivos y directorios pueden encontrarse en distintos dispositivos. Antes de poder acceder a ellos ha de realizarse el montaje del dispositivo que es la operación de asociar un dispositivo a un directorio. Los fundamentos sobre el uso de los archivos en Linux son:

- ✓ Los nombres de archivo diferencian mayúsculas de minúsculas.
- ✓ El directorio raíz se referencia simplemente mediante `/`.
- ✓ Cada directorio tiene un nombre que puede incluir letra o símbolo excepto la barra de dividir `/`. El directorio raíz constituye una excepción; su nombre es `/` y no puede modificarse su nombre.
- ✓ Cada archivo o directorio se designa por su nombre de archivo completo, nombre de archivo absoluto o ruta, dando la secuencia de directorios que deben recorrerse para localizarlo. Los tres términos son sinónimos. Todos los nombres de archivo absolutos comienzan con el directorio `/` y hay una `/` entre cada directorio o carpeta en el nombre de archivo. La primera `/` es el nombre del directorio raíz pero las restantes son simplemente separadores para distinguir las partes del nombre de archivo. Por ejemplo: `/usr/share/keytables/us.map.gz` Este es un nombre de archivo completo; aunque se suele conocer como ruta. No obstante, la mayoría considera el nombre de archivo únicamente a `us.map.gz` y el resto como su ruta o path absoluto.
- ✓ El nombre de archivo admite todos los caracteres, excepto `/`. La longitud máxima del nombre del archivo es de 255 caracteres, sin incluir su ruta.
- ✓ El directorio raíz tiene un cierto número de ramas o subdirectorios, que a su vez se ramifican en más subdirectorios. Todo el conjunto se denomina árbol de directorios.
- ✓ En cada directorio existen dos directorios especiales: El punto `.` hace referencia al propio directorio y los dos puntos `..` hacen referencia al directorio anterior en la ruta, es decir, al directorio padre. El directorio raíz `/` no tiene directorio padre. Los dos puntos referencia a sí mismo, igual que el punto simple.
- ✓ No existe un árbol de directorios por cada dispositivo de almacenamiento, como ocurre en Windows, donde tenemos una raíz en cada unidad de volumen como por ejemplo `C:\`. Solo hay uno que cuelga de `/` y desde aquí se accede a cualquier archivos en cualquier dispositivo de almacenamiento. Cada uno de ellos estará montado en un directorio al cual podremos acceder desde la raíz.

En Linux/Unix todo es un archivo: los archivos de datos, las particiones, los sockets y los dispositivos hardware (teclado, ratón, impresora, discos duros...). Durante la instalación del sistema se ha creado una jerarquía de directorios y archivos necesarios por el sistema operativo. Estos son:

- ✓ `/` Directorio raíz.
- ✓ `/bin` Comandos esenciales del sistema.
- ✓ `/boot` Archivos del cargador de arranque.
- ✓ `/dev` Archivos de dispositivo.
- ✓ `/etc` Archivos de configuración de sistema específicos del ordenador.
- ✓ `/lib` Bibliotecas compartidas precisas para la ejecución del sistema local.
- ✓ `/mnt` Puntos de montaje temporal.
- ✓ `/opt` Paquetes de software añadidos.
- ✓ `/proc` Configuración e instantáneas del núcleo en vivo.
- ✓ `/sbin` Comandos de administración del sistema.
- ✓ `/tmp` Archivos temporales. Un sistema que se comporte bien los desechará en cada inicio.
- ✓ `/usr` Datos compartidos.
- ✓ `/var` Archivos de tamaño variable, como buzones de correo y archivos de registro.
- ✓ `/home` Archivos personales de los usuarios.
- ✓ `/root` Archivos personales del superusuario

Hay algunos detalles interesantes a tener en cuenta respecto a algunos de los directorios del sistema:

Como medida de seguridad, los directorios `/tmp` y `/var` pueden alojarse en sus propias particiones. Si algo se tuerce y provoca que se llenen de manera incontrolada, estarán aislados del resto del sistema.

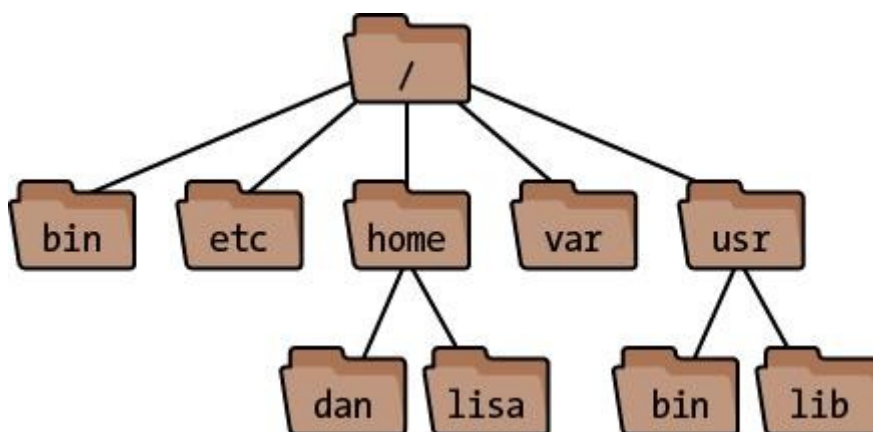


Figura 1.- Árbol de almacenamiento en GNU/Linux

El directorio `/home` puede alojarse en su propia partición, facilitando la ejecución de copias de seguridad y protegiéndolo de actualizaciones del sistema. De esta manera

podríamos volver a instalar el sistema operativo, o incluso instalar una distribución diferente, mientras `/home` permanece inalterado.

El nombre de un archivo y la ruta a la carpeta del archivo lo identifica de manera unívoca entre todos los demás archivos del sistema, no puede haber dos archivos con el mismo nombre y ruta. La ruta de un archivo puede ser de dos tipos:

- ✓ Ruta completa o absoluta.- Es la ruta desde la raíz hasta la carpeta o archivo al que nos referimos. Por ejemplo: `/home/usuario/documentos/informe.odt` sería la ruta absoluta hasta el archivo `informe.odt`.
- ✓ Ruta relativa.- Es la ruta que se forma desde el directorio dónde estemos situados. Se obvia la información desde la raíz hasta el directorio donde estamos situados. Por ejemplo, si en un momento determinado estamos situados en la carpeta `/home/usuario` sería más sencillo que hagamos referencia al archivo anterior con su ruta relativa `documentos/informe.odt` que indicar la ruta completa.

## 1.2 Extensión en los nombres de archivo

En el antiguo sistema operativo MS-DOS de Microsoft, antecesor de los actuales Windows, el nombre de archivo estaba formado por ocho caracteres, un punto y tres caracteres. A estos tres caracteres finales se les conocía como la extensión del archivo y se empleaba para identificar el tipo de archivo. Así, un archivo con extensión `txt` significa que es de texto, `doc` es un documento, `exe` una aplicación, etc. El nombre de archivo no podía tener un punto ya que era el carácter que se empleaba para separar el nombre de la extensión.

En los sistemas de archivos actuales la extensión en el nombre de archivo no existe y se admite el punto como carácter para el nombre de archivo. Sin embargo, se sigue manteniendo la norma de que los últimos tres o cuatro caracteres definan el tipo de archivo y sean precedidos por un punto. Esta norma no es obligatoria, pero si resulta muy conveniente seguirla. Así, la mayoría de las aplicaciones que crean archivos añaden un punto y tres caracteres al nombre de archivo para identificar su tipo y las aplicaciones para gestionar archivos utilizan estos caracteres para asignarles el icono correspondiente al tipo.

Por tanto será habitual encontrarse con archivos cuyos nombres acaban en un punto y tres o cuatro caracteres para identificar su tipo. Los más habituales son:

- ✓ `txt`.- Texto
- ✓ `doc`.- Documento
- ✓ `jpg`.- Imagen comprimida
- ✓ `zip`.- Archivo comprimido
- ✓ `png`.- Imagen
- ✓ `gif`.- Dibujo



## 2 Introducción al uso de comandos

---

En su origen Linux no tenía una interfaz gráfica. Las primeras versiones de Linux solamente incluían una interfaz texto similar a la de Unix. Esto significa que las tareas administrativas tenían que hacerse mediante comandos en línea. De ahí que este tipo de interfaz se denominara CLI (*Command Line Interface*).

Posteriormente aparecieron los gestores de escritorio, pero estos estaban orientados a los usuarios más que a los administradores. Consecuentemente, las tareas administrativas seguían realizándose mediante comandos. Conforme pasó el tiempo y fueron apareciendo nuevos gestores de escritorio y nuevas versiones de los mismos, estos fueron incorporando nuevas aplicaciones que permitían realizar algunas tareas administrativas en el sistema. En realidad, estas aplicaciones eran front-ends las cuales internamente se encargaban de traducir las órdenes dadas por los usuarios mediante la interfaz gráfica a comandos que se ejecutaban en segundo plano. Por supuesto, el uso de comandos para realizar tareas administrativas se mantuvo y ha permanecido hasta nuestros días.

Actualmente gran parte de la administración de un sistema Linux es mediante comandos del sistema operativo, con lo que para administrar un sistema Linux es imprescindible tener conocimientos de los comandos más habituales además de la estructura interna del sistema operativo en lo que se refiere a sus funciones principales: gestión de archivos y dispositivos, gestión de procesos, etc.

### 2.1 El intérprete de comandos

Una *Shell* es el término usado en informática para referirse a un intérprete de comandos, el cual consiste en la interfaz de usuario tradicional de los sistemas operativos basados en Unix y similares como GNU/Linux. Mientras que Windows posee dos intérpretes de comandos para trabajar en modo comando (el símbolo del sistema y Power Shell), Linux dispone de varios. En Linux los usuarios pueden elegir trabajar con distintos Shell, entre los que se encuentran la C Shell (*csh*), la Shell de Korn (*ksh*) y la Shell de Bourne (*bash*). Este último se ha convertido en un estándar de facto, ya que es el más utilizado en las diferentes distribuciones Linux. En Ubuntu se emplea *Bash* como intérprete de comandos y será el que utilicemos en este documento.

### 2.2 Consolas de texto

Cuando Linux arranca lo hace por defecto en la consola gráfica. Sin embargo, también se activan las consolas de texto para el caso de que el usuario quiera abrir una sesión texto. Por defecto hay seis consolas de texto a las que se puede acceder mediante la combinación de teclas **Ctrl+Alt+Fn**, siendo **n** un número entre 1 y 6.

Para abrir una sesión en modo texto solamente tenemos que introducir el nombre de usuario y la contraseña. Si la autenticación tuvo éxito entonces aparecerá el indicador de órdenes o *prompt*. Cuando finalizamos la sesión podemos cerrarla escribiendo el comando `exit`.

Es decir, podemos tener abiertas hasta 6 sesiones texto simultáneamente las cuales

requieren pocos recursos del equipo. La sesión gráfica se encuentra después de las consolas texto y se puede regresar a ella en cualquier momento tecleando `Ctrl+Alt+F7`. Naturalmente, podemos simultanear sesiones texto con sesión gráfica e incluso podemos disponer de más sesiones gráficas, cosa que no se recomienda ya que una sesión gráfica consume muchos recursos del equipo, sobre todo memoria.

Como se mencionó anteriormente, la consola texto muestra en pantalla un indicador de línea de comandos, denominada *prompt*, esperando que el usuario introduzca un comando. Este indicador finaliza generalmente por un carácter \$, cuando eres un usuario normal, o # cuando eres un súper usuario. El *prompt* se puede personalizar y en Ubuntu muestra el siguiente aspecto por defecto:

```
usuario@equipo:directorio_actual$
```

El carácter ~ que usualmente verás al iniciar una consola texto indica que estás en el directorio personal del usuario. Es una especie de abreviatura de esta carpeta.

## 2.3 Ventana de terminal

Desde una sesión gráfica también podemos trabajar con comandos del sistema operativo. Disponemos de una aplicación, el terminal, el cual equivale al símbolo del sistema de Windows. La encontramos en accesorios y podemos invocarla desde el tablero.

1. Abrir Aplicaciones.
2. En la barra de búsqueda de aplicaciones escribir *Terminal*.
3. Hacer clic sobre el icono del terminal.

El terminal ofrece una serie de ventajas sobre una consola de texto. Algunas de ellas son:

- ✓ Barra de desplazamiento para recorrer la actividad del usuario.
- ✓ Posibilidad de abrir varias solapas para disponer de varios terminales.
- ✓ Personalizar el aspecto como el color de fondo, la fuente, color de texto, tamaño, etc.
- ✓ Podemos editar y trabajar con el contenido del terminal.

## 2.4 Usuario administrador y usuario root

En los sistemas GNU/Linux el usuario administrador es el que se crea durante la instalación y puede realizar tareas administrativas. Sin embargo, el superusuario que tiene acceso a todo sin límites se denomina `root` y en Ubuntu está deshabilitado por defecto.

Cada vez que utiliza una aplicación que modifique la configuración del equipo aparecerá un cuadro de autenticación para que introduzca la contraseña del usuario administrador. Si está trabajando mediante comando del sistema operativo y tiene que acceder a partes del sistema a las cuales solamente está autorizado el usuario `root`, entonces también tendrá que autenticarse como usuario administrador.

La forma de hacerlo es algo diferente ya que tendrá que ejecutar el comando precediéndolo de `sudo`. La sintaxis sería así:

```
sudo comando
```

Le pedirá la contraseña del usuario administrador y si la autenticación tiene éxito el comando se ejecutará. Anteponer el prefijo `sudo` a la ejecución de un comando indica que se quiere ejecutar el comando como si fuera el usuario `root`.

También podríamos habilitar al usuario `root` y abrir sesión con él. Sin embargo, por defecto, en Ubuntu el usuario `root` no puede abrir sesiones gráficas, únicamente sesiones texto, aunque también puede cambiarse la configuración para que pueda abrir sesiones gráficas.

## 2.5 Documentación de comandos

La documentación sobre programas Linux es abundante. No obstante, hallarla puede ser algo difícil. No hay montones de relucientes y brillantes manuales impresos, porque la mayor parte del software para Linux se distribuye por Internet, en lugar de empaquetarlo en resplandecientes cajas.

Además existe otra dificultad: Linux sigue la antigua tradición de Unix de tener pequeños programas especializados cooperando entre sí, razón por la cual cualquier distribución Linux contiene un gran número de programas individuales. Una distribución de propósito general contiene varios miles de programas además existen unos 15.000 paquetes en los repositorios de Debian. A pesar de que organizar y mantener una biblioteca impresa presenta algunas dificultades, la buena noticia es que todos esos programas están documentados. Independientemente de lo que quiera saber, hay una forma de encontrar documentación.

Además, el espíritu cooperativo que originó Linux es lo que lo mantiene, y no solo para crear nuevas versiones de distribuciones y/o programas. La documentación que se confecciona se comparte a través de Internet y existen multitud de foros y sitios web dedicados al mundo Linux en los que se puede encontrar de todo.

### 2.5.1 Documentación de Ubuntu

El sitio web <http://www.ubuntu.com> permite acceder al portal de Ubuntu donde podemos encontrar documentación del sistema operativo. El enlace *Community → Ubuntu Documentation* nos lleva a la documentación oficial de Ubuntu. Aquí podemos elegir la versión de Ubuntu con la que estamos trabajando y acceder a la ayuda de muchas tareas administrativas que se encuentran categorizadas.



## Official Ubuntu Documentation

Documentation developed and maintained by the Ubuntu Documentation Project.

Release	Desktop	Server	Installation Guide
Ubuntu 20.10 (Groovy Gorilla)	<a href="#">HTML</a>	<i>unpublished</i>	<i>unpublished</i>
Ubuntu 20.04 LTS (Focal Fossa)	<a href="#">HTML</a>	<a href="#">HTML and PDF</a>	<a href="#">per architecture</a>
Ubuntu 18.04 LTS (Bionic Beaver)	<a href="#">HTML</a>	<a href="#">HTML and PDF</a>	<a href="#">per architecture</a>
Ubuntu 16.04 LTS (Xenial Xerus)	<a href="#">HTML</a>	<a href="#">HTML and PDF</a>	<a href="#">per architecture</a>

### Notes:

- As of the Ubuntu LTS release in 2020, the server documentation has moved to [a different site](#), and will automatically update when changes are made to the [discourse source code pages](#).
- The Desktop and 16.04 Serverguide 'HTML' links above are available in many different languages. They will be displayed in the preferred language specified by your browser. If the preferred language is not available they will be displayed in English. The 18.04 'HTML' and all 'PDF' Serverguides and the Installation Guide are available solely in English.
- Unsupported releases are not shown here. See [instructions for upgrading to a supported version](#).
- The Server Guide and the Installation Guide are published only for Long Term Support (LTS) releases.
- Useful information: [LTS details](#) and [release & end-of-life dates](#) (for all versions).

Figura 2.- Documentación oficial de Ubuntu

Disponemos de la documentación en versiones de escritorio y server. en formatos HTML y PDF.

### 2.5.2 Manual universal de comandos: man

Prácticamente, cada programa escrito para Linux tiene su página [man](#)<sup>1</sup>. Normalmente es la mejor herramienta de aprendizaje para administradores de sistemas Linux que comienzan. La mayor parte de las tareas de administración y configuración de un sistema Linux se realiza a través de comandos del SO que se ejecutan en un terminal. El propósito de las páginas [man](#) es documentar la sintaxis del comando y cada opción con que cuenta.

Siempre habrá páginas [man](#) para los programas que se tenga instalados. Dado que las páginas de manual se incorporan a los propios programas a los que pertenecen, comprobará que únicamente aquellos programas que tiene instalados cuentan con páginas de manual y que la versión de esas páginas es la que corresponde a la versión de ese programa o comando.

Las páginas de información tienden a ser más explicativas que las páginas de manual y además contienen hipervínculos. Éstos llevan a los distintos nodos o capítulos, existentes en la documentación y a referencias cruzadas a otras páginas de información. Muchas páginas de información son simplemente páginas de manual a las cuales se ha vuelto a dar formato, pero en algunos casos, principalmente en programas del proyecto GNU, las páginas de información son más detalladas, conteniendo guías didácticas y extensos ejemplos.

Linux ve todas las páginas de manual de un sistema como si fueran parte de un único

<sup>1</sup> Las páginas de manual son las que pueden consultarse mediante el comando man, denominadas en la propia documentación Linux como man pages

manual el cual está dividido en secciones:

1. Programas ejecutables o comandos del intérprete de comandos (shell).
2. Llamadas de sistema.
3. Llamadas de bibliotecas.
4. Archivos especiales (alojados normalmente en `/dev`).
5. Formato de archivo y convenciones.
6. Juegos.
7. Miscelánea.
8. Comandos de administración del sistema.
9. Rutinas no estándar del núcleo
10. Documentación nueva, que podría ser trasladada posteriormente.
11. Documentación local, específica de su sistema.

Cada uno de los programas, utilidades o funciones tiene su propia página en este manual, como una página en un libro. Encontrar la página de manual de un programa o comando normalmente es tan fácil como escribir en una ventana de terminal o consola de texto `man concepto`, donde `concepto` es el nombre del programa, comando, fichero, etc.

Es posible que haya dos o más páginas de manual para un mismo concepto. Esto es porque hay una página de manual para un comando y otra para un programa o fichero con el mismo nombre. Sin embargo, el manual está dividido en secciones y cada uno de ellos pertenecería a una sección diferente. En este caso si escribimos en la ventana de terminal o consola de texto `man concepto` nos llevará a la primera página del manual. Si queremos ver la página de manual en una sección concreta podemos indicar el número de la sección antes del concepto. Por ejemplo, si escribimos lo siguiente

```
man crontab
```

Nos aparecerá la página de manual del comando `crontab`, el cual se emplea para programar la ejecución de tareas. Al ser un comando está en la sección nº 1. Sin embargo, existe un fichero con el mismo nombre, el cual se emplea también para gestionar la ejecución de tareas programadas y cuya página de manual se encuentra en la sección nº 5. En este caso, si quisiéramos ver su página de manual tendríamos que ejecutar lo siguiente

```
man 5 crontab
```

¿Cómo sabemos que sección estamos consultando en las páginas del manual? En la primera línea de las páginas de manual viene el nombre del comando o fichero que estamos consultando y entre paréntesis aparece el número de sección de la página del manual.

Cada página del manual está dividida en secciones. Los nombres de éstas varían, pero normalmente serán los siguientes:

- ✓ **NOMBRE.-** Nombre del programa, archivo o comando que se está consultado su documentación.
- ✓ **SINOPSIS.-** Sintaxis de uso. Puede haber más de una sintaxis para un mismo comando, utilizando una u otra dependiendo de la tarea concreta que se pretenda realizar.
- ✓ **DESCRIPCIÓN.-** Explicación breve de lo qué hace o para qué se utiliza.
- ✓ **OPCIONES.-** Parámetros o argumentos que pueden usarse.
- ✓ **ARCHIVOS.-** Archivos relacionados con el programa o comando.
- ✓ **EJEMPLOS.-** Ejemplo de uso.
- ✓ **VEA TAMBIÉN.-** Otros conceptos relacionados con el actual y que conviene también consultar.
- ✓ **BUGS.-** Errores detectados en la versión actualmente instalada.
- ✓ **AUTOR/TRADUCCIÓN.-** Autor del programa y de la traducción de la página de manual.

La sintaxis de los comandos es fundamental ya que indica como hay que escribir correctamente un comando para su ejecución. La notación utilizada para mostrar la sintaxis de los comandos, que se encuentra en la sinopsis de las páginas de manual es:

- ✓ **Texto en negrita.-** Escriba esto exactamente como se muestra.
- ✓ Texto subrayado.- El texto en cursiva denota un argumento, lo cual significa que debe reemplazarlo con el valor que se desee. Dependiendo del visor que esté utilizando, puede ser que no vea subrayado sino en cursiva.
- ✓ **Opciones.-** Las opciones del comando pueden ser de dos tipos:
  - ✗ **Corta.-** Generalmente es un guión medio con una o dos letras. Por ejemplo `-a -h -j`. Una opción se separa de la siguiente por espacio. Se pueden combinar varias utilizando solo un guión y poniendo las letras de las opciones sin espacios. Siguiendo con el ejemplo anterior sería `-ahj`. Es posible que la opción necesite un valor, para lo cual se indicaría a continuación de la opción y separada de esta por un espacio, por ejemplo `-f fichero.tar.gz`.
  - ✗ **Larga.-** En este caso la opción esta formada por un conjunto de palabras. Siempre comienza por doble guión medio. Si tiene más de una palabra, estas se separan con guiones medio simples. Por ejemplo `--update --text-label --force --decompress`. Es muy posible que existan opciones en versión corta y larga, pudiendo utilizar la que mejor convenga. Por general, las largas son más autoexplicativas al usar una palabra que ofrece una pista sobre su función en el

comando.

- ✓ Corchetes `[opción ... ] [argumento...]` .- Podemos ver corchetes encerrando opciones y/o argumentos. Indica que la opción y argumento es opcional y puede omitirse. Cualquier argumento u opción que no esté encerrado entre corchetes es obligatorio o la ejecución del comando fallará.
- ✓ Llaves y barra vertical `{ opción1 | opción2 | opción3 | ... argumento1 | argumento2 | ... }` . Las opciones u argumentos separados por el símbolo tubería `|` no pueden combinarse entre sí, son excluyentes mutuamente. Hay que usar uno y solo uno. Si además están encerrados entre corchetes entonces indica opcionalidad además de exclusividad.
- ✓ Puntos suspensivos `...` .- Los puntos suspensivos indican que puede introducir varios argumentos. Observe cuáles son los delimitadores entre los argumentos, normalmente serán espacios pero en ocasiones se usan comas.

Como hemos visto anteriormente existen dos tipos de opciones: las cortas y las largas. Las opciones cortas suelen estar compuestas por un guión y una o dos letras. Pueden escribirse de dos maneras, la primera es combinando varias opciones con un solo guión.

```
-abc
```

o bien, poniendo las opciones cada una con su guión:

```
-a -b -c
```

Las opciones largas están formadas por un doble guión seguido de una palabra o término. Únicamente pueden introducirse individualmente:

```
--opcion1 --opcion2 -opcion3
```

Es habitual que una opción corta tenga una equivalente en larga y ambas vienen agrupadas en la página de manual para explicar su funcionamiento.

Por ejemplo, la página de manual del comando `update-rc.d` tiene la siguiente sección sinopsis. Este comando se emplea para gestionar los enlaces a los scripts de inicio y parada de servicios que se encuentran en `/etc/init.d/`.

```
SYNOPSIS
update-rc.d [-n] [-f] name remove
update-rc.d [-n] name defaults [NN | SS KK]
update-rc.d [-n] name start|stop NN runlevel [runlevel]...
.
start|stop NN runlevel [runlevel]... . ...
update-rc.d [-n] name disable|enable [ S|2|3|4|5 ]
```

Este comando tiene cuatro sintaxis diferentes. Veamos con detenimiento la primera:

```
update-rc.d [-n] [-f] name remove
```

Tiene dos opciones: `-n` y `-f`, las cuales son opcionales. A continuación hay que escribir un nombre de servicio y al final la palabra `remove`.

La segunda sintaxis que aparece en la página del manual es la siguiente:

```
update-rc.d [-n] name defaults [NN | SS KK]
```

Tiene la opción `-n`, que es opcional. A continuación hay que escribir un nombre de servicio y después la palabra `defaults`. Finalmente tenemos opción de escribir un número (`NN`) o dos números (`SS KK`).

La tercera se complica algo ya que tiene más opciones. Es esta:

```
update-rc.d [-n] name start | stop NN runlevel [runlevel]...
.
start | stop NN runlevel [runlevel]... . ...
```

La primera es `-n`, la cuál es de nuevo opcional. Después hay que escribir la palabra `start` o `stop`, pero no ambas. Después hay que escribir un número y uno o varios valores `runlevel`. Después del último hay que poner un punto. A continuación hay que escribir de nuevo `start` o `stop` y otro número seguido de una lista de valores `runlevel` separados por un espacio y que acaban en punto. Así sucesivamente.

La última sintaxis de este comando es este:

```
update-rc.d [-n] name disable | enable [ S | 2 | 3 | 4 | 5 ]
```

Tiene la opción `-n`, que es opcional. A continuación hay que escribir un nombre de servicio y después la palabra `disable` o `enable`. Opcionalmente podemos indicar la letra `S` o un número entre `2` y `5`.

### 2.5.3 Descripción de comandos

Al comienzo de cada página de manual aparece la descripción del comando. Es un breve texto que indica qué hace el comando. Si solamente necesitamos ver esta descripción podemos utilizar el comando `whatis`.

#### Sintaxis

```
whatis comando.
```

Por ejemplo, si queremos ver la descripción del comando `tar` ejecutamos lo siguiente:

```
whatis tar
tar (1)          - an archiving utility
```

### 2.5.4 Búsqueda de comandos

Hay ocasiones en las que no conocemos cuál es el programa, comando o utilidad para llevar a cabo una tarea, por ejemplo contar las palabras contenidas en un archivo, ya que no sabemos el comando que realiza dicha tarea. Tenemos dos formas de solucionar este problema: usando el comando `apropos` o utilizando `man -k`.



### Sintaxis

```
apropos lista de palabras clave  
man -k lista de palabras clave
```

Ambos son idénticos. Estos comandos efectúan búsquedas de palabras clave en las secciones de descripción de las páginas de manual. Podemos utilizar cualquier número de términos de búsqueda, pero cuantos más utilicemos más resultados habrá, porque busca cada palabra dada en la secuencia.

Por ejemplo, podemos buscar todos los comandos que tienen que ver con la gestión de usuarios. Ejecutaríamos el siguiente comando

```
apropos user
```

O también

```
man -k user
```

## 2.6 Comandos iniciales

Para acostumbrarnos al manejo de la interfaz de comandos en línea vamos a ver una serie de comandos empleados para obtener diversos tipos de información del sistema. Para cada uno de ellos se mostrará su sintaxis con los argumentos y opciones, pero debido al número alto de opciones que un comando puede tener solamente mostraremos los más habituales o interesantes.

### 2.6.1 Mostrar identificador de usuario

El comando `whoami` muestra el nombre del usuario que ha abierto sesión actualmente.

#### Sintaxis

```
whoami [OPTION]...
```

#### Opciones

`--help`

Muestra la ayuda del comando

`--version`

Muestra la versión del comando

También disponemos del comando `id`, el cual nos muestra más información, como el identificador de usuario y los grupos a los que pertenece.

#### Sintaxis

```
id [OPTION]... [USERNAME]
```

## Argumentos

### USERNAME

Nombre de usuario del que se quiere mostrar la información

## Opciones

### -g, --group

Visualiza solo el identificador de grupo principal

### -G, --groups

Visualiza los identificadores de todos los grupos del usuario

### -n, --name

Visualiza un nombre en lugar de un número. En combinación con -ugG

### -u, --user

Visualiza el identificador del usuario

Si lo ejecutamos sin argumentos ni opciones mostrará la información anterior del usuario que ha abierto sesión.

El comando `groups` se emplea para mostrar los grupos de un usuario.

## Sintaxis

```
groups [OPTION]... [USERNAME]...
```

## Argumentos

### USERNAME

Nombre de usuario del que se quiere mostrar la información

## Opciones

### --help

Muestra la ayuda del comando

### --version

Muestra la versión del comando

Si lo ejecutamos sin argumentos ni opciones mostrará los grupos a los que pertenece el usuario que ha abierto sesión.

## 2.6.2 Mostrar directorio activo

Si queremos mostrar en pantalla el directorio donde nos encontramos actualmente tenemos el comando `pwd`.

## Sintaxis

```
pwd [OPTION]... [USERNAME]...
```

## Opciones

`--help`

Muestra la ayuda del comando

`--version`

Muestra la versión del comando

## 2.6.3 Mostrar el histórico de comandos

En el siguiente apartado veremos con mas detenimiento el histórico de comandos, pero ahora podemos consultar la lista de comandos ejecutados con el comando `history`.

## 2.6.4 Limpiar la pantalla

Conforme vayamos ejecutando comandos y el resultado de los mismos se presenten por pantalla está se llenará de texto. Es posible que después de un tiempo la pantalla del terminal aparezca algo confusa y resulte incómodo buscar visualmente un dato concreto. En momentos como estos un borrado de la pantalla viene bien y podemos hacerlo con el comando `clear` que no tiene argumentos ni opciones.

## 2.6.5 Información de la conexión de red

El comando `ifconfig` sirve para configurar y mostrar información de las interfaces de red.

## Sintaxis

```
ifconfig [-v] [-a] [-s] [interface]
ifconfig [-v] interface [atype] options | address ...
```

La segunda sintaxis se emplea para configurar la tarjeta de red de forma estática, algo no muy habitual.

## Argumentos

`interface`

Interfaz de red a configurar o de la que muestra información

## Opciones

`-a`

Muestra información de todas las interfaces de red.

`-s`

Muestra información estadística de las interfaces de red.

## 2.6.6 Mostrar la fecha del sistema

Con el comando `date` podemos establecer y mostrar la fecha y la hora del sistema.

### Sintaxis

```
date [OPTION]... [+FORMAT]
date [-u|--utc|--universal] [MMDDhhmm[ [CC]YY] [.ss]]
```

### Argumentos

#### +FORMAT

Formato de salida de la fecha. Para saber como poner una cadena de formato de impresión de la fecha consultar la página de manual del comando `date`.

`[-u|--utc|--universal]`

Muestra la hora UTC que equivale a la hora GMT.

`[MMDDhhmm[ [CC]YY] [.ss]]`

Fecha a establecer. MM corresponde al mes, DD al día hh a la hora, mm a los minutos, CCYY al año y .ss a los segundos.

Relacionado con la fecha tenemos también el comando `cal`, el cual muestra un calendario con el mes actual. Este comando es muy completo y solamente veremos su sintaxis más básica.

### Sintaxis

```
cal [-3h jy] [-A number] [-B number] [[month] year]
```

### Argumentos

`[[month] year]`

Mes y año que toma como punto de referencia.

### Opciones

`[-A number]`

Número de meses que muestra después del mes de referencia.

`[-B number]`

Número de meses que muestra antes del mes de referencia.

`-3`

Muestra el mes actual, el anterior y el siguiente.

`-h`

No resalta el día actual

`-j`

Muestra el número de días desde el 1 de Enero en lugar del día del mes.

`-y`

Muestra el calendario del año actual

## 2.6.7 Muestra información del sistema

El comando `uname` muestra diversa información del sistema.

### Sintaxis

```
uname [OPTION] ...
```

### Opciones

`[-n]`

Nombre del equipo en la red.

`[-r]`

Versión del kernel

`-s`

Nombre del kernel

`-p`

Tipo de procesador.

`-i`

Tipo de plataforma hardware.

`-o`

Nombre sistema operativo.

## 2.6.8 Mostrar nombre del equipo

El comando `hostname` muestra el nombre del equipo. También se puede emplear para asignar un nombre al equipo, aunque solamente tiene efecto hasta el reinicio del equipo. Para poner un nombre al equipo es mejor modificar directamente el archivo `/etc/hostname`.

## 2.7 Histórico de comandos

Escribir comandos puede ser una tarea tediosa y propensa a errores. Hay comandos largos con múltiples argumentos y/u opciones que hay que escribir cuidadosamente siguiendo la sintaxis para evitar errores y conseguir el objetivo que se persigue con la

ejecución del comando.

La bash almacena el histórico de comandos ejecutados, 500 por defecto, incluso de aquellos que hemos escrito incorrectamente y su resultado fue un error. Es probable que necesitemos repetir la ejecución de algún comando que hemos ejecutado previamente. Para ello podemos utilizar algunos comandos del histórico. Los más habituales son:

Comando	Descripción
Flecha arriba o <code>Ctrl+p</code>	Muestra el comando anterior
Flecha abajo o <code>Ctrl+n</code>	Muestra el comando siguiente cuando previamente fuimos hacia atrás
<code>!n</code>	Ejecuta el comando nº n en el histórico
<code>!-n</code>	Ejecuta el comando -n desde el último
<code>!!</code>	Ejecuta el comando recién ejecutado
<code>!cadena</code>	Ejecuta el comando más reciente cuyo nombre comienza por cadena
<code>Alt+&lt;</code>	Muestra el primer comando del histórico
<code>Alt+&gt;</code>	Muestra el último comando del histórico
<code>Ctrl+R</code>	Realiza una búsqueda hacia atrás con la cadena introducida

Cuando se nos muestra un comando del histórico podemos editarlo desplazándonos por él para realizar algún cambio con las teclas de desplazamiento flecha izquierda y derecha.

Además, si alguno de los argumentos es el nombre de un archivo podemos utilizar el tabulador para completar su nombre. Conforme vayamos tecleando el path del archivo podemos pulsar la tecla tabulador y el bash autocompletará el nombre de la carpeta hasta la cadena común de nombres de carpetas más larga de todas las que existan. Si solo hay una, completará el nombre de la carpeta entera y añadirá el carácter / para seguir escribiendo nombres de carpeta. Así sucesivamente hasta que se consiga el path absoluto del archivo.

### 3 Montaje de dispositivos de almacenamiento

Ya hemos visto anteriormente que los archivos de dispositivos se encuentra en `/dev` y tienen que ser asociados a un directorio o carpeta mediante una operación denominada montaje. No se podrá acceder a los archivos de un dispositivo hasta que no se haya realizado el montaje del dispositivo que consiste en asociar una carpeta o directorio al dispositivo.

Cuando introducimos un soporte de almacenamiento óptico en la unidad óptica del ordenador o conectamos un pendrive en un puerto USB, el sistema operativo ejecuta automáticamente una operación de montaje en la cual asigna una carpeta al contenido de dicho soporte. A partir de entonces podremos acceder al contenido del dispositivo de almacenamiento a través de esta carpeta.

Una vez que terminamos de utilizar el dispositivo, hay que desmontar el soporte de almacenamiento externo, lo que significa que se está eliminando la asignación de la carpeta con este soporte. A partir de entonces la carpeta donde estaba montado el dispositivo se borra automáticamente.

Las carpetas que se emplean para el montaje de los soportes externos se crean automáticamente y todas ellas suelen estar en la carpeta `/media/nombre_usuario`. Veamos ahora como se comportan los habituales soportes de almacenamiento externo: los discos CD o DVD y los pendrive.

### 3.1 Identificación de dispositivos

Los archivos se encuentran almacenados en un dispositivo de almacenamiento, como una partición en el disco duro, un disquete, CD-ROM, pendrive, etc. En Linux, cada dispositivo se representa mediante un fichero de dispositivo por bloques. Un fichero de dispositivo por bloques es aquel que se transfiere información en lectura o escritura por bloques de un tamaño que puede ir desde los 512 bytes hasta los 32Kb dependiendo del tamaño de cluster en el dispositivo.

¿Cómo saber el archivo de dispositivo que corresponde a un dispositivo de almacenamiento? Los ordenadores pueden disponer de varios tipos de interfaces de discos duros o dispositivos externos de almacenamiento: IDE, SCSI, SATA y M2 para SSD. La última se va imponiendo poco a poco, mientras que la interfaz IDE era la habitual en los ordenadores domésticos más antiguos, aunque todavía existen muchos ordenadores que la tienen en sus discos duros. La interfaz SCSI se emplea en aquellas situaciones en las que se requiere un rendimiento mejorado como servidores web, bases de datos, etc. Lo más habitual es encontrarse con interfaces SATA en ordenadores domésticos y portátiles.

La interfaz IDE permite la conexión de hasta cuatro dispositivos. En la placa base hay dos conectores IDE, cada uno de los cuales puede tener dos dispositivos conectados mediante un cable con dos conexiones. Teniendo en cuenta que las unidades ópticas de almacenamiento como CD-ROM y DVD's también se conectan a esta interfaz, podemos tener en el ordenador un total de cuatro dispositivos entre discos duros, lectoras de CD y/o DVD y regrabadoras. Los dos conectores IDE de la placa base se denominan IDE0 e IDE1. A cada uno de estos podemos conectar dos dispositivos, denominados primario y secundario (o maestro y esclavo). Según esto, los archivos de dispositivos que corresponden serían:

- ✓ Dispositivo conectado al IDE0 (primario) maestro: `/dev/hda`
- ✓ Dispositivo conectado al IDE0 (primario) esclavo: `/dev/hdb`
- ✓ Dispositivo conectado al IDE1 (secundario) maestro: `/dev/hdc`
- ✓ Dispositivo conectado al IDE1 (secundario) esclavo: `/dev/hdd`

Para las unidades SCSI y SATA el archivo de dispositivo es `/dev/sda` para el primer disco, `/dev/sdb` para el segundo, y así sucesivamente.

Si el dispositivo representa un disco duro, el montaje se realiza sobre una partición. Al

archivo de dispositivo hay que añadir un número que representa el orden de la partición dentro del disco. Por ejemplo, para la primera partición del disco duro conectado al IDE0 (primario) maestro sería `/dev/hda1`. La cuarta partición del disco duro conectado al IDE1 (secundario) esclavo sería `/dev/hdd4`. Con los dispositivos SCSI y SATA ocurre lo mismo, es decir el archivo `/dev/sda1` es la primera partición del primer disco duro y `/dev/sdb2` es la segunda partición del segundo disco duro.

Si el disco duro contiene varias particiones y no estamos seguros del número exacto que queremos montar podemos ejecutar el comando `fdisk` para visualizar la tabla de particiones del disco. Por ejemplo, si ejecutamos el siguiente comando mostrará la tabla de particiones del primer disco duro SATA, con sus tipos y tamaños.

```
sudo fdisk -l /dev/sda
```

Las unidades ópticas suelen ser identificadas con `/dev/sr0` mientras que las memorias flash se identifican igual que los discos SATA.

## 3.2 Montaje y desmontaje con la GUI

El montaje de dispositivos externos de almacenamiento mediante la GUI es muy simple y está orientado al usuario. A continuación se muestra cómo se hace en función del tipo de dispositivo externo.

### 3.2.1 Unidades ópticas

Cuando el usuario introduce un CD o DVD en la unidad lectora, el sistema lo detecta automáticamente y mostrará un icono en el escritorio para abrirlo. Al hacer clic sobre él, si el CD o DVD dispone de un autoejecutable pedirá permiso para abrirlo. De lo contrario simplemente abrirá el administrador de archivos y mostrará su contenido. En cualquier caso, cuando abrimos el administrador de archivos podemos ver que en el panel lateral hay un nuevo elemento con el título del CD o DVD que al hacer clic sobre él nos mostrará su contenido.

Al ser una unidad óptica, solamente podremos copiar archivos desde aquí, pero no podremos borrarlos ni mover los ya que es de sólo lectura.

Una vez que hayamos terminado de trabajar con el CD o DVD podemos extraerlo automáticamente haciendo clic con el botón derecho del ratón sobre el icono del escritorio o sobre el elemento del panel lateral y seleccionar la opción *Expulsar*.

### 3.2.2 Memorias flash

Cuando conectamos un pendrive o un disco duro externo a un puerto USB, el sistema lo detecta y mostrará un icono en el escritorio por cada partición del dispositivo. Además, abrirá el administrador de archivos con su contenido visualizado directamente. Vemos como en el panel lateral también ha creado un elemento que corresponde a éste dispositivo de almacenamiento USB.

A partir de entonces podemos gestionar los archivos igual que si estuvieran en otra carpeta del sistema de archivos. Cuando terminemos de trabajar con él y queramos



desconectarlo hay que seguir los siguientes pasos:

1. Hacer clic sobre el elemento en el lanzador o en el administrador de archivos con el botón derecho del ratón.
2. Seleccionar la opción *Extraer unidad de forma segura*.
3. Si se hubieran borrado archivos en el pendrive, aparecerá un cuadro de diálogo preguntando si vacía la papelera de reciclaje del dispositivo.

### 3.3 Montaje y desmontaje en CLI. Comandos mount y umount

Existen multitud de ocasiones en las que un administrador de sistemas se ve obligado a realizar un montaje manual de alguna partición o dispositivo de almacenamiento. La mayoría de ellas será cuando esté realizando operaciones de administración en una consola de texto si la distribución Linux no tiene una GUI. Para realizar esta operación se emplea el comando `mount`.

#### Sintaxis

```
mount [opciones] dispositivo directorio.
```

#### Argumentos

##### dispositivo

Partición o dispositivo a montar. Hay que indicar un nombre de archivo que representa a la partición o al dispositivo. Estos archivos están en el directorio `/dev`.

##### directorio

Directorio donde se realizará el montaje.

#### Opciones

##### -t sistema de archivos

Se especifica el sistema de archivos del dispositivo. Los más habituales son:

- ✓ ntfs Sistema de archivos ntfs, la empleada por Windows XP, 2000, Server. Empleando este tipo solamente se puede realizar el montaje de la partición en solo lectura.
- ✓ vfat Sistema de archivos fat en cualquiera de sus versiones. Empleada en MS-DOS, Windows 95, 98, disquetes y unidades USB.
- ✓ ext2, ext3, ext4 Sistema de archivos ext2, ext3 y ext4 empleadas en distribuciones Linux.

##### -o opciones

Cadena de caracteres con las opciones del montaje específicas del sistema de archivos separadas por coma y sin espacios en blanco. Ver el siguiente epígrafe.

##### --bind dir\_actual dir\_nuevo

Vuelve a montar el dispositivo en otro directorio. A partir de entonces los archivos del

dispositivo están disponibles en dos directorios. Al desmontar uno de los directorios, la información continúa disponible en el otro. Si se desmonta utilizando el archivo de dispositivo solamente lo desmontará del último directorio montado con esta opción, en el otro estará disponible y habrá que desmontarlo explícitamente.

### --move dispositivo directorio

Mueve el montaje de un dispositivo de un directorio a otro. El desmontaje solamente se podrá hacer con el archivo de dispositivo o con el directorio actual en el que esté montado.

## 3.3.1 Opciones de montaje

La opción `-o` del comando `mount` se emplea para indicar las opciones de montaje del dispositivo específico. Existen un conjunto de opciones, algunas de las cuales son excluyentes entre sí por que una opción invalida a la otra. Las opciones que se indiquen deberán ir separadas por comas y sin espacios, formando una única cadena. Las más habituales son:

- ✓ `atime`.- Actualiza la fecha y hora de acceso de los ficheros. Esta opción está activada por defecto.
- ✓ `noatime`.- No actualiza la fecha y hora de acceso de los ficheros del dispositivo. Esto acelera el acceso a los ficheros al no tener que almacenar información relativa a la fecha y hora del último acceso en el sistema de ficheros.
- ✓ `exec`.- Permite la ejecución de archivos binarios del dispositivo. Activada por defecto
- ✓ `noexec`.- No permite la ejecución de archivos binarios del dispositivo.
- ✓ `group`.- Permite a un usuario no `root` realizar el montaje si uno de sus grupos es el grupo propietario del dispositivo.
- ✓ `ro`.- Monta el sistema de ficheros en solo lectura.
- ✓ `rw`.- Monta el sistema de ficheros en lectura y escritura.
- ✓ `remount`.- Realiza un nuevo montaje sobre un dispositivo ya montado. Se emplea cuando queremos cambiar las opciones de montaje del dispositivo.
- ✓ `uid=nº`.- Asigna la carpeta de montaje al usuario con uid indicado.
- ✓ `gid=nº`.- Asigna la carpeta de montaje al grupo con gid indicado.

Existen otras, pero estas son las más habituales. Algunas de ellas solo tiene sentido utilizarlas dentro del fichero `/etc/fstab` que se verá más adelante.

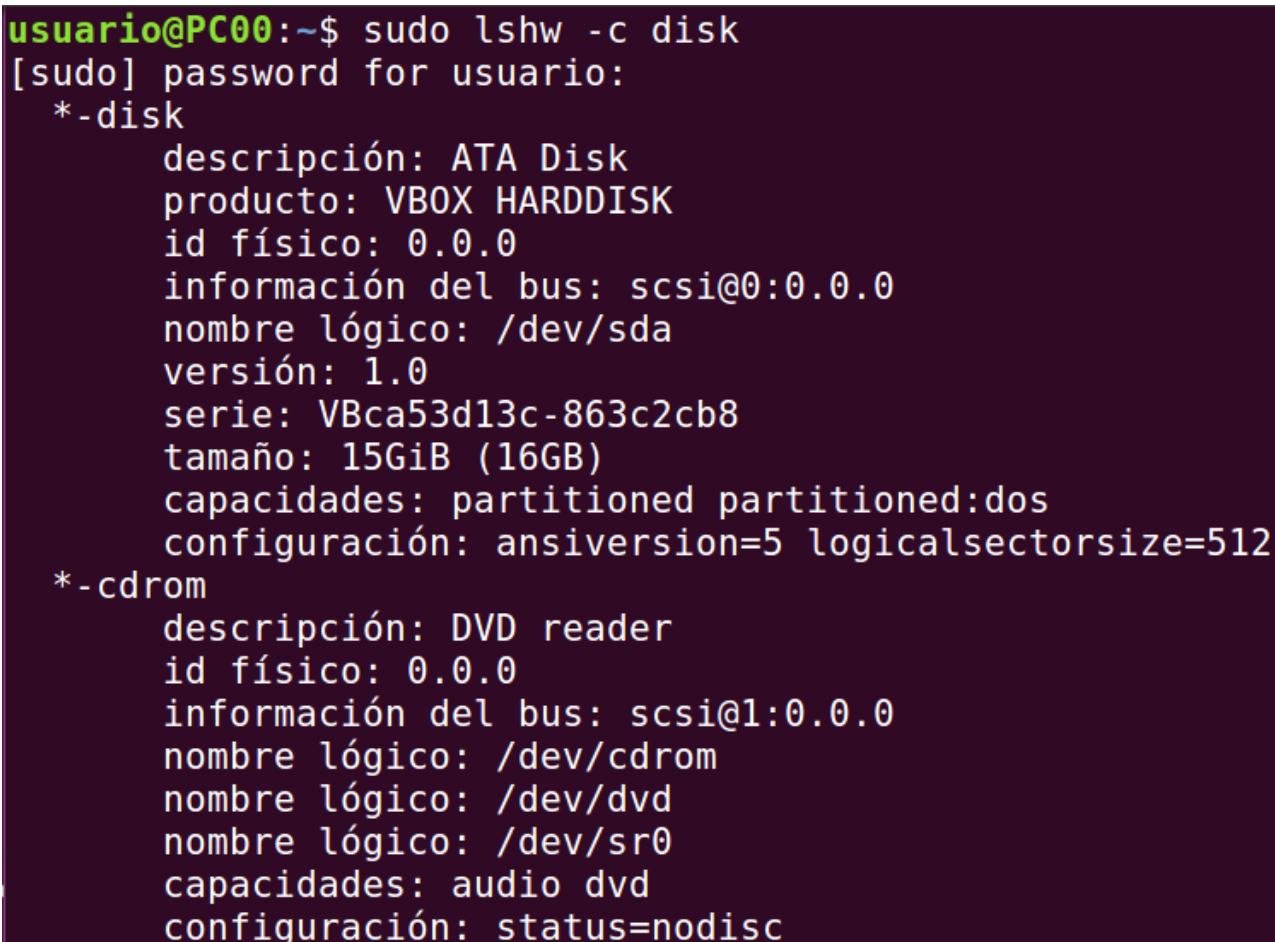
## 3.3.2 Montaje de dispositivos de almacenamiento externos

El montaje de los dispositivos de almacenamiento externos (como CD's, DVD's, pen drive, etc) es muy parecido al montaje de las particiones de los discos duros. La diferencia estriba en los sistemas de ficheros que hay que indicar en cada caso y/o del archivo de dispositivo que corresponde a cada uno.

Para los CD's y DVD's se emplean los mismos archivos de dispositivos que para los discos duros, ya que al tener interfaz IDE o SATA las unidades ópticas se conectan a la placa igual que un disco duro. El problema es cómo saber donde están conectadas estas unidades. Para no tener que abrir el ordenador ni entrar en la BIOS para averiguarlo podemos utilizar el comando `lshw`, el cual muestra información sobre el hardware del ordenador. En este caso habría que ejecutar el siguiente comando:

```
sudo lshw -c disk
```

Si ejecuto este comando la salida es parecida a lo siguiente:

A terminal window with a dark purple background. The prompt is 'usuario@PC00:~\$'. The command 'sudo lshw -c disk' is entered. The output shows the password prompt '[sudo] password for usuario:' followed by a list of hardware details for the disk and cdrom. The disk section shows it's a VBOX HARDDISK with a size of 15GiB (16GB). The cdrom section shows it's a DVD reader.

```
usuario@PC00:~$ sudo lshw -c disk
[sudo] password for usuario:
*-disk
   descripción: ATA Disk
   producto: VBOX HARDDISK
   id físico: 0.0.0
   información del bus: scsi@0:0.0.0
   nombre lógico: /dev/sda
   versión: 1.0
   serie: VBca53d13c-863c2cb8
   tamaño: 15GiB (16GB)
   capacidades: partitioned partitioned:dos
   configuración: ansiversion=5 logicalsectorsize=512
*-cdrom
   descripción: DVD reader
   id físico: 0.0.0
   información del bus: scsi@1:0.0.0
   nombre lógico: /dev/cdrom
   nombre lógico: /dev/dvd
   nombre lógico: /dev/sr0
   capacidades: audio dvd
   configuración: status=nodisc
```

Figura 3.- Salida del comando lshw

Vemos que en el PC donde se ha ejecutado el comando hay un disco duro y una unidad óptica DVD. En ambos disponemos del campo nombre lógico que indica el fichero de dispositivo para cada uno. El disco duro está en `/dev/sda` y el DVD en `/dev/sr0`. El sistema de ficheros que se emplea habitualmente en las unidades ópticas es el ISO9660. Para montar un CD o DVD en el directorio `/media/cdrom` se teclearía el siguiente comando:

```
mount -t iso9660 /dev/sr0 /media/cdrom
```

Habría que indicar que el dispositivo es de solo lectura, ya que las escrituras en CD-R/RW o DVD-R/RW se realizan con programas específicos para ellos. Sin embargo,

cuando se intenta montar un dispositivo en lectura y escritura y no es posible, visualiza un mensaje de error y lo monta en solo lectura.

Para las memorias flash que se conectan en los puertos UBS no es muy distinto. Para acceder a estos dispositivos Linux hace una emulación SCSI, es decir, los trata como si fueran discos duros SCSI, por tanto sus archivos de dispositivo serían `/dev/sdb1`, `/dev/sdb2`, etc, en función del puerto USB donde se conectara. Hay que tener en cuenta que habitualmente `/dev/sda` es el disco duro SATA. El sistema de archivos sería también `vfat`. El montaje de una unidad de este tipo en el directorio `/media/lapiz` sería:

```
mount -t vfat /dev/sdb1 /media/lapiz
```

### 3.3.3 Desmontaje

Cuando un dispositivo va a dejar de utilizarse se desmonta. A partir de entonces su información deja de estar accesible. Si hicimos el montaje de forma manual con el comando `mount`, para realizar el montaje del dispositivo se emplea el comando `umount`.

#### Sintaxis

```
umount directorio | dispositivo
```

#### Argumentos

##### directorio

Directorio sobre el que está montado el dispositivo.

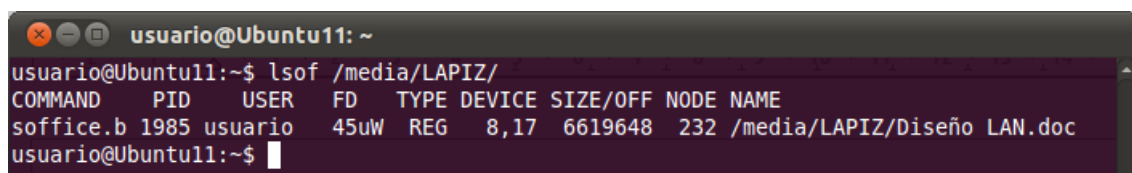
##### dispositivo

Dispositivo que se va a desmontar.

Puede usarse uno u otro, es indiferente. El desmontaje puede fallar si el dispositivo está ocupado, por ejemplo si una aplicación tiene un archivo el dispositivo abierto. Para saber qué archivo es y que aplicación lo tiene abierto se puede emplear el comando `lsof`. Por ejemplo, si queremos ver que archivos del pendrive que está montado en la carpeta `/media/lapiz` están abiertos:

```
lsof /media/lapiz
```

Mostrará una lista con los archivos abiertos y las aplicaciones que los tienen abiertos. En la siguiente imagen muestra que el archivo `Diseño LAN.doc` está abierto por el usuario `usuario` y la aplicación `soffice.b`.



```
usuario@Ubuntu11: ~
usuario@Ubuntu11:~$ lsof /media/LAPIZ/
COMMAND  PID   USER   FD   TYPE DEVICE SIZE/OFF  NODE NAME
soffice.b 1985 usuario 45uW  REG   8,17 6619648 232 /media/LAPIZ/Diseño LAN.doc
usuario@Ubuntu11:~$
```

Figura 4.- Archivo abierto

### 3.3.4 Configuración estática del montaje. Fichero `/etc/fstab`

El comando `mount` solamente puede ser empleado por el usuario `root` o por el administrador con `sudo`, pero ¿qué ocurre cuando se necesita que usuarios monten particiones del disco duro y no se les quiere dar privilegios de superusuario para que puedan hacerlo? También se desea controlar los sistemas de archivos que se montan automáticamente durante el inicio del sistema. Para esto existe el archivo `/etc/fstab`.

Este archivo de texto contiene una línea por cada dispositivo que se desea controlar su montaje. El formato de cada línea incluye:

- ✓ Dispositivo a montar.- Se indica mediante el correspondiente fichero de dispositivo. Si el sistema de archivos es ext2 o superior se puede indicar el identificador o la etiqueta con `UUID=<uuid>` y `LABEL=<label>`.
- ✓ Punto de montaje o directorio sobre el que se montará el dispositivo. Para particiones swap hay que especificar `none`. Si el nombre del directorio contiene espacios en blanco se tienen que escapar con `\040`.
- ✓ Tipo del sistema de archivos. Si se especifica `auto`, el sistema lo intentará detectar. Linux soporta muchos tipos de sistemas de archivos. Consultar la página de manual para saber cuales son. Si el tipo es swap entonces se emplea como área de intercambio.
- ✓ Opciones de montaje. Lista de opciones de montaje separadas por comas y sin espacios.
- ✓ `dump`. Si se emplea el comando `dump` para hacer copias de seguridad, este campo indica el intervalo, en días, con que se hará la copia. El valor 1 significa cada día, 2 significa un día sí y otro no, y así sucesivamente.
- ✓ `pass`.- En caso de que sea preciso comprobar los sistemas de archivos, este campo indica al comando `fsck` cuál debe verificar primero. El sistema de archivos raíz debe tener 1, los sistemas de archivos Linux deben tener 2 y 0 a todos los demás sistemas de archivos que no son Linux.

Las opciones de montaje que se pueden emplear son las mismas que se vieron anteriormente, pero además existen otras que son adecuadas utilizarlas aquí:

- ✓ `user`.- Los usuarios no administradores pueden montar el dispositivo, y solo puede desmontarlo el superusuario y el usuario que lo montó.
- ✓ `nouser`.- Solamente el superusuario puede realizar el montaje.
- ✓ `users`.- El dispositivo puede ser montado y desmontado por cualquier usuario.
- ✓ `auto`.- El dispositivo debe montarse durante el inicio del sistema.
- ✓ `noauto`.- El dispositivo debe ser montado explícitamente por parte de un usuario y no durante el inicio del sistema.

- ✓ **defaults**.- Utiliza un conjunto predefinido de opciones por defecto que incluyen: **rw, suid, dev, exec, auto, nouser, async**.

Existe la posibilidad de emplear la opción **defaults** y modificar alguna de sus opciones, como por ejemplo **defaults,user,noauto**.

No es recomendable establecer montaje al inicio del sistema para dispositivos externos ya que es probable que al inicio del sistema no estén conectados o aquellos que no se vayan a utilizar, ya que el sistema estaría dedicando recursos.

Hay una ventaja adicional cuando se añade un dispositivo al fichero **/etc/fstab** y es que a partir de entonces puede realizarse el montaje del dispositivo simplemente indicando el directorio de montaje o el archivo de dispositivo. El comando **mount** tomará el resto de información que necesite de este fichero.

Hace tiempo que las distribuciones Linux prefieren utilizar el UUID en lugar del archivo de dispositivo para indicar el primer campo. Según la documentación de Linux añade más robustez al sistema. Si vemos el contenido del siguiente archivo **/etc/fstab**.

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=ea78895e-5dec-465e-bff5-b764aed0dba8 / ext4 errors=remount-ro 0 1
# /home was on /dev/sda5 during installation
UUID=5c2dad15-cb60-45be-9f76-a531f69cc23d /home ext4 defaults 0 2
# swap was on /dev/sda6 during installation
UUID=d47c1770-63e6-41c4-a963-c31d61f2c129 none swap sw 0 0
```

Figura 5.- Archivo **/etc/fstab**

Las líneas que comienzan por **#** son comentarios.

Antes de cada línea con el montaje de un dispositivo tiene una línea de comentario en la que indica que archivo de dispositivo está montando. La primera línea que aparece para el montaje de un dispositivo es la siguiente

```
UUID=ea78895e-5dec-465e-bff5-b764aed0dba8 / ext4 errors=remount-ro
0 1
```

Corresponde a la partición del disco duro que se monta sobre la raíz del sistema de archivos. La partición está indicada con el UUID y el directorio de montaje es **/**. El sistema de archivos es **ext4** y la opción de montaje indica que se monte en solo lectura si ocurre un error. El campo **dump** tiene el valor 0, es decir, no se hará copias de seguridad de esta partición. Por último el campo **fsck** tiene el valor 1, ya que es la partición raíz del sistema.

La segunda línea es la siguiente

```
UUID=5c2dad15-cb60-45be-9f76-a531f69cc23d /home ext4 defaults 0
2
```

Vemos que la partición se indica también con UUID y se monta en la carpeta **/home**. El

sistema de archivos es ext4 y utiliza las opciones por defecto para el montaje. Por último no se harán copias de seguridad y en la comprobación de los sistemas de archivos se hará en segundo lugar, después de la partición raíz.

La última línea es la siguiente

```
UUID=d47c1770-63e6-41c4-a963-c31d61f2c129 none swap sw 0
0
```

Resulta evidente que corresponde a la partición que se utiliza como intercambio. El punto de montaje es `none` y el sistema de archivos es swap.

En el caso de que creáramos una nueva partición en algún disco duro del PC, ¿como podemos saber el identificador de la partición si quisiéramos utilizarlo en lugar del archivo de dispositivo? El comando `blkid` muestra todas las particiones con sus identificadores.

## 4 Archivos

El gestor de archivos de GNOME que viene con Ubuntu se denomina Archivos. Nos permite organizar y gestionar los archivos y directorios, incluyendo los que se encuentran en medios extraíbles como unidades ópticas o memorias USB. A través de él podemos ver, eliminar, copiar o mover archivos y carpetas. Puedes abrir Archivos de dos formas:

- ✓ Haciendo clic en el primer botón del lanzador
- ✓ Haciendo una búsqueda de archivos en el tablero. Al hacer clic sobre la carpeta encontrada la abrirá con el Archivos.

### 4.1 La ventana de Archivos

Al abrir Archivos te encontrarás con una ventana similar a la de la imagen. Puede que el aspecto cambie un poco ya que la podemos personalizar.

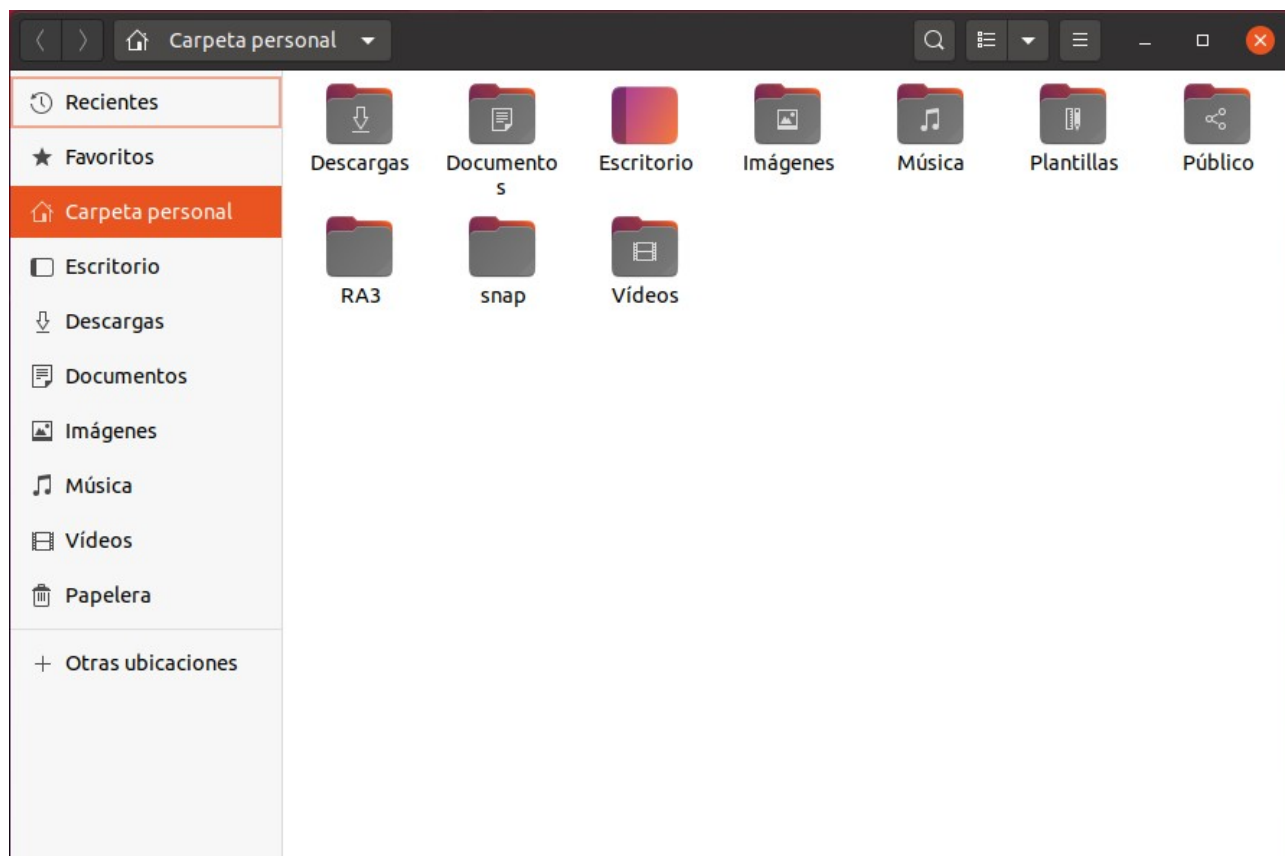


Figura 6.- Ventana de Archivos

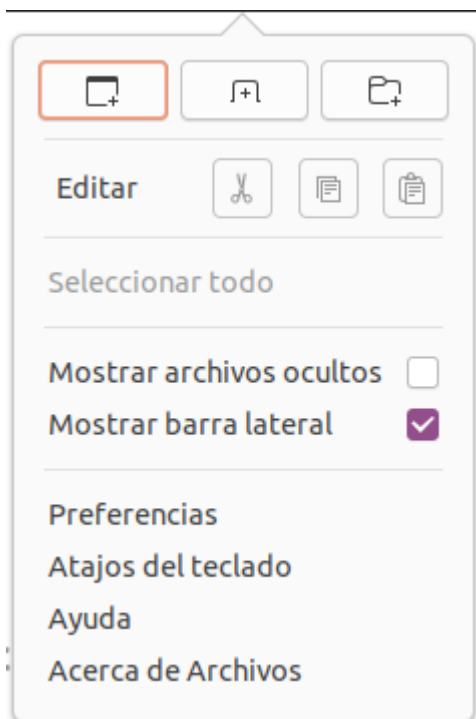


Figura 7.- Botón menú

A continuación se relacionan los elementos que podemos utilizar en una ventana de Archivos.

- ✓ **Barra de herramientas.-** Contiene botones con las operaciones más habituales para gestionar archivos.
- ✓ **Botón de menú.-** Contiene el menú que se pueden usar para realizar tareas de gestión de archivos. Aparece en la parte derecha de la barra de herramientas.
- ✓ **Barra de lugar.-** La barra de lugar es una herramienta muy potente para navegar por el equipo. Puede aparecer de tres maneras distintas dependiendo de su selección.
- ✓ **Panel lateral.-** Contiene las carpetas principales del perfil de usuario.
- ✓ **Panel de contenido.-** Muestra el contenido de la carpeta activa.






### 4.1.1 Botón menú


Dispone de botones y opciones de menú para realizar las operaciones más habituales. Si alguna opción o botón aparece difuminada significa que no puede usarse ya que se necesita una condición previa, como seleccionar un conjunto de archivos.

### 4.1.2 La barra de herramientas

Está situada justo debajo de la barra de título y contiene botones para realizar operaciones frecuentes.

Los botones  y  permiten navegar hacia atrás (a la carpeta de la que venimos) y hacia adelante (si hemos usado el botón atrás). El sistema guarda un historial de las carpetas que vamos visitando dentro de esa misma ventana para permitirnos ir adelante y atrás. Al hacer clic con el botón derecho del ratón sobre los botones Atrás y Adelante, permite desplegar una lista de las carpetas que se han recorrido en un sentido u en otro. Al hacer clic sobre una de las carpetas se mostrará su contenido. De esta forma, no es necesario pulsar sobre los botones Atrás y Adelante repetidas veces para saltar de una carpeta a otra que se recorrió anteriormente y se encuentra a diferente nivel de la primera.

El botón de búsqueda  abre la barra de búsqueda de archivos, que se verá más adelante.

Con el botón  podemos alternar la vista de los archivos a iconos o lista y en la flecha abajo desplegamos un menú para aumentar o disminuir el tamaño de los iconos y ordenar la lista de archivos.

Vista de lista visualiza las carpetas y archivos con información adicional como el tamaño de archivo, tipo y fecha de modificación. Si es una carpeta en lugar del tamaño se visualiza el número de elementos (archivos y carpetas) que contiene.











Nombre	Tamaño	Modificación	Destacar
 Descargas	0 elementos	lun	☆
 Documentos	0 elementos	lun	☆
 Escritorio	0 elementos	lun	☆
 Imágenes	0 elementos	lun	☆
 Música	0 elementos	lun	☆
 Plantillas	0 elementos	lun	☆
 Público	0 elementos	lun	☆
 RA3	3 elementos	mié	☆
 snap	1 elemento	mié	☆
 Vídeos	0 elementos	lun	☆

Figura 8.- Vista de lista

Cuando tenemos la vista de lista podemos ordenar el listado por los campos de información que visualiza los mismos. Los encabezados de los campos son botones que al hacer clic sobre ellos ordena el resultado. Si se vuelve a hacer clic sobre un campo que ya está ordenado entonces invierte la ordenación, si era descendente la hace ascendente y viceversa. Sabemos el campo actual de ordenación porque aparece una pequeña flecha en la parte derecha del botón. Si está hacia abajo significa la ordenación es ascendente y si está hacia arriba es descendente.

#### 4.1.3 El panel lateral

Este panel puede estar visualizado o no. Para mostrar el panel lateral, elegir la opción de menú *Mostrar la barra lateral*. Aquí podemos observar los siguientes elementos:

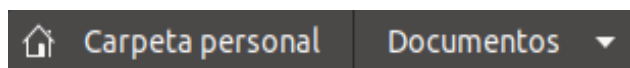
- ✓ Recientes.- Muestra los archivos que se han utilizado recientemente.
- ✓ Favoritos.- Muestra los archivos marcados como favoritos.
- ✓ Carpeta personal.- Seleccionando este elemento nos desplazamos a la carpeta personal del usuario.
- ✓ Escritorio.- Muestra el contenido del escritorio del usuario.
- ✓ Descargas.- Muestra el contenido de la carpeta [Descargas](#).
- ✓ Documentos.- Muestra el contenido de la carpeta [Documentos](#).

- ✓ Imágenes.- Muestra el contenido de la carpeta [Imágenes](#).
- ✓ Música.- Muestra el contenido de la carpeta [Música](#).
- ✓ Vídeos.- Muestra el contenido de la carpeta [Vídeos](#).
- ✓ Papelera.- Muestra los archivos borrados que se encuentran en la papelera de reciclaje.
- ✓ Otras ubicaciones.- Al seleccionarlo, realiza una búsqueda de los equipos que hay conectados en red y que tengan carpetas compartidas.

En el caso de que conectemos un dispositivo de almacenamiento extraíble como un CD o DVD, o conectamos una memoria flash, nos aparecerá su correspondiente etiqueta en el panel lateral para acceder a su contenido.

#### 4.1.4 Barra de lugar

Esta barra nos indica dónde estamos situados y qué otras carpetas hay por encima y debajo de la actual en la jerarquía del almacenamiento. Es dinámica y su contenido cambia con la actividad del usuario al navegar por las diferentes carpetas.



Está compuesta por un conjunto de botones, uno por cada carpeta que hemos ido abriendo durante la navegación con Archivos. Si navegamos por la misma rama del árbol irán apareciendo más botones, en la parte delante si estamos navegando por niveles superiores de la rama del árbol o al final si estamos profundizando en la misma rama del árbol.

Cuando hacemos clic en una carpeta que pertenece a otra rama, entonces los botones que aparecerán serán los necesarios para recorrer el camino hasta la carpeta actual.

Estos botones nos permiten ir directamente a cada carpeta sin necesidad de deshacer el camino recorrido. Aquél botón que se encuentra hundido corresponde a la carpeta en la que nos encontramos en ese momento.

También podemos navegar a una carpeta cualquiera directamente sin necesidad de pasar por ninguna otra si conocemos el path absoluto de dicha carpeta. Si pulsamos la combinación de teclas Ctrl+L la barra de lugar cambia y aparece un cuadro de texto donde podemos teclear el path que queremos visualizar.

#### 4.1.5 Panel de contenido

En esta zona aparece la lista de los archivos contenidos en la carpeta en que estamos situados, es decir, la carpeta de la barra de lugar. Si hemos efectuado una búsqueda sólo aparecerán los archivos que cumplan las condiciones de la búsqueda. La forma y cantidad de información que vemos de cada archivo depende del tipo de vista.

El panel de contenido puede tener diferentes pestañas, cada una con el contenido de una carpeta. Si pulsamos la combinación de teclas Ctrl + T o vamos a la opción de menú *Archivo* → *Nueva pestaña* nos abrirá otra pestaña. Podemos abrir las pestañas que

necesitemos.

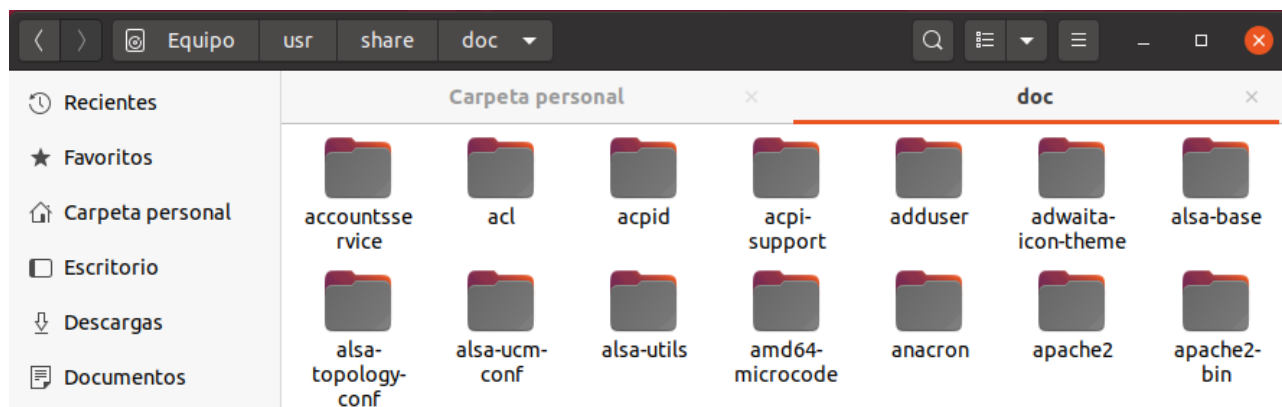


Figura 9.- Pestañas en el panel de contenido

Cada pestaña tiene un botón de cerrado en su extremo superior derecho. Al hacer clic sobre él la pestaña se cierra.

## 4.2 Gestión de archivos y carpetas

Esta sección explica como trabajar con archivos y carpetas.

### 4.2.1 Crear una carpeta

El usuario tiene pleno control sobre su carpeta personal, mientras que en otras carpetas del sistema de archivos generalmente solo podrá visualizar su contenido. Por tanto, lo más probable es que el usuario solamente pueda crear carpetas en su carpeta personal o en alguna que haya a partir de esta. Así, cada usuario puede crearse su propia estructura de carpetas y subcarpetas para organizar mejor sus archivos.

Para crear una carpeta hay que seguir los siguientes pasos:

1. Situar en la carpeta donde queremos crear una nueva.
2. Hacer una de las siguientes opciones:
  - a) Hacer clic con el botón derecho del botón y seleccionar la opción *Carpeta nueva*.
  - b) Desplegar el menú y seleccionar el botón *Carpeta nueva*.
  - c) Pulsar la combinación de teclas **Mayús+Ctrl+N**.
3. Escribir el nombre de la nueva carpeta y pulsar Intro.

Naturalmente, la nueva carpeta está vacía y lista para guardar archivos o crear nuevas carpetas.

### 4.2.2 Abrir un archivo

La mayoría de los archivos tienen una acción predeterminada cuando hacemos doble clic sobre él. Generalmente esta consiste en abrirse con la aplicación que creó dicho archivo.

Por ejemplo, abrir un archivo de música lo reproducirá con la aplicación de reproducción de música predeterminada, abrir un archivo de texto permitirá leerlo y editarlo en un editor de texto, y abrir un archivo de imagen mostrará la imagen.

El gestor de archivos comprueba el contenido de un archivo para determinar el tipo de un archivo. Si las primeras líneas no determinan el tipo del archivo, entonces el gestor de archivos comprueba la extensión de archivo.

Si se abre un archivo de texto ejecutable, esto es, uno que Archivos considera que puede ejecutarse como un programa, entonces preguntará qué queremos hacer: ejecutarlo, o mostrarlo en un editor de texto. Se puede modificar este comportamiento en las preferencias del Archivos.

Para ejecutar acciones distintas de la acción predeterminada para un archivo, seleccionar el archivo en el que quiere ejecutar una acción y ejecutar la opción de menú *Archivo* → *Abrir con*. Seleccionar la opción deseada desde esta lista. También podemos hacer clic con el botón derecho del ratón sobre el archivo y seleccionar la opción *Abrir con*.

### 4.2.3 Selección de archivos

Muchas operaciones con archivos y/o carpetas implica indicar sobre qué archivos o carpetas se realizará la operación. Por ejemplo, cuando borramos un archivo hay que indicar previamente el archivo que queremos borrar. Esto se hace con una selección de archivos.

Podemos seleccionar archivos individuales o conjuntos de archivos. Dependiendo de la selección que queramos hacer tenemos las siguientes opciones:

- ✓ Para seleccionar un archivo solamente hay que hacer clic sobre él.
- ✓ Para seleccionar un conjunto de archivos que están contiguos en el panel de visualización hacemos clic con el ratón sobre una zona en blanco y arrastramos el ratón para formar un recuadro. Todos los archivos que estén dentro del recuadro quedarán seleccionados.
- ✓ Podemos hacer varias selecciones pulsando la tecla Ctrl cuando hacemos una nueva selección. Así, mantendremos la anterior que uniremos a la nueva.
- ✓ Si elegimos la opción de menú *Seleccionar todo* o pulsamos la combinación de teclas Ctrl+A todos el contenido de la carpeta se selecciona.
- ✓ Si elegimos el atajo de teclado Shift + Ctrl + I invertimos la selección, es decir, estamos seleccionando los archivos que no estaban seleccionados mientras que perdemos la selección de los que si lo estaban.

Existe otra forma de seleccionar archivos basándonos en el nombre del archivo. Consiste en construir un patrón y todo archivo cuyo nombre se ajuste a este patrón será incluido dentro de la selección.

Para construir el patrón usaremos los caracteres que pueden formar parte del nombre de un archivo y dos caracteres especiales:

- ✓ El cierre de interrogación ? sustituye a un carácter.
- ✓ El asterisco \* sustituye a 0 o varios caracteres.

Por ejemplo, al patrón `ap*.txt` se ajustarán todos los archivos cuyo nombre comienza por `ap` y acaban en `.txt`. En este caso el asterisco sustituye a cualquier conjunto de caracteres que hay entre `ap` y `.txt`.

Otro ejemplo, al patrón `chang*.t??` se ajustarán todos los archivos cuyo nombre comienza por `chang` y acaba en `.t` y dos caracteres cualesquiera más. En este caso el asterisco sustituye a cualquier conjunto de caracteres que hay después de `chang` y antes de `.t`. Los dos cierres de interrogación sustituyen a un carácter cada uno.

Para seleccionar todos los archivos cuyo nombre se ajusta a un patrón hay que pulsar el atajo de teclado Ctrl+S. Aparecerá un cuadro de diálogo para introducir el patrón y haremos clic en el botón *Aceptar*.

Ten en cuenta que si tienes seleccionados varios elementos, y haces clic sobre otro, sin pulsar ni la tecla Ctrl ni Mayúsculas, perderás la selección previa y sólo quedará seleccionado el elemento sobre el que has pulsado.

#### 4.2.4 Copiar archivos

Al copiar un elemento, archivo o carpeta, lo que estamos haciendo es duplicarlo, crear una copia de él, en otra ubicación o en la misma. Cuando queramos crear copias exactas de un conjunto de archivos seguiremos los siguientes pasos:

1. Situar en la carpeta donde están los archivos a copiar.
2. Seleccionar los archivos que queremos copiar.
3. Hacer una de las siguientes acciones:
  - a) Elegir la opción de menú *Copiar*.
  - b) Pulsar la combinación de teclas Ctrl+C
  - c) Hacer clic sobre el botón derecho del ratón sobre cualquier archivo de la selección y seleccionar la opción *Copiar*.
4. Navegar a la carpeta de destino.
5. Realizar una de las siguientes acciones:
  - a) Elegir la opción de menú *Pegar*.
  - b) Pulsar la combinación de teclas Ctrl+V
  - c) Hacer clic sobre el botón derecho del ratón sobre cualquier archivo de la selección y seleccionar la opción *Pegar*

Si copiamos una carpeta, también copiamos su contenido, incluyendo subcarpetas y archivos. Si la carpeta de destino es la misma que la de origen, el elemento se renombrará

como `Nombre original (copia)`.

No se permite que haya dos archivos o carpetas con el mismo nombre en la misma carpeta. Si en la carpeta de destino es distinta a la original y existe algún archivo con el mismo nombre que uno que estamos copiando aparecerá un cuadro de diálogo como el siguiente.



Figura 10: Conflicto en la copia de archivos

Nos avisa de que ya existe un archivo con el mismo nombre y nos ofrece las siguientes opciones:

- ✓ Cancelar.- Cancela la copia de archivos y no copia ninguno. Si antes de encontrar la coincidencia se copiaron algunos, estos permanecen.
- ✓ Omitir.- No copia el nuevo archivo, por lo tanto, permanece el antiguo.
- ✓ Reemplazar.- Sustituye el nuevo archivo por el antiguo. Si encuentra alguna coincidencia más volverá a presentar este cuadro de diálogo.

Activando la casilla *Aplicar esta acción a todos los archivos* no volverá a preguntar en el caso de que haya más conflictos y aplicará la acción elegida en ese momento.

También podemos escribir un nuevo nombre para el archivo de destino haciendo clic en *Seleccionar un nombre nuevo para el destino*.

En el caso de que el conflicto ocurra con una carpeta el mensaje que aparece es el siguiente:

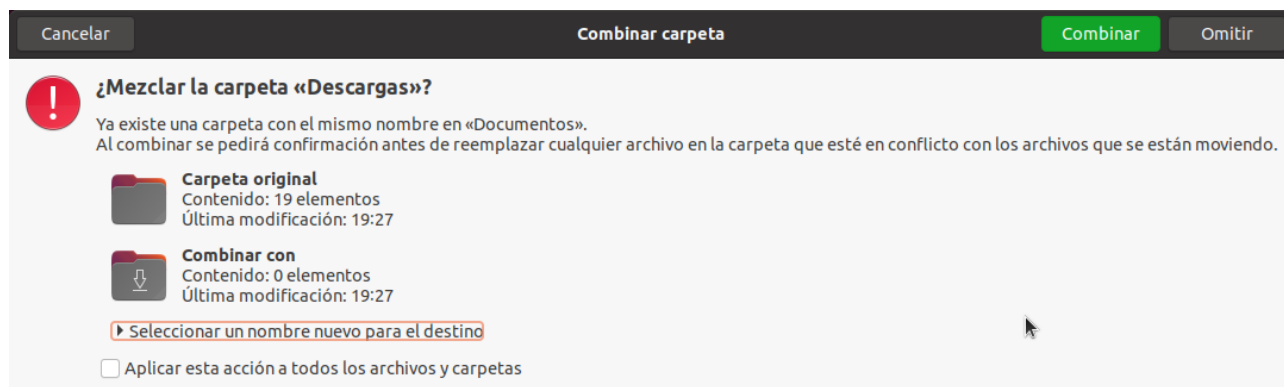


Figura 11: Conflicto en la copia de carpetas

Aquí disponemos de las siguientes opciones:

- ✓ Omitir.- No copia la carpeta, por lo tanto, permanece la antigua.
- ✓ Combinar.- Copia los archivos de la nueva carpeta en la antigua. Si hay conflicto en la copia de archivos actúa como se ha visto anteriormente. Si encuentra alguna coincidencia más volverá a presentar este cuadro de diálogo.

De nuevo disponemos de la casilla *Aplicar esta acción a todos los archivos* para aplicar la misma acción elegida a los conflictos que puedan volver a ocurrir en la operación actual de copia o movimiento de la carpeta.

En este conviene recordar que dos archivos con el mismo nombre no implican que sean iguales, de la misma forma dos carpetas con el mismo nombre no tienen los mismos archivos. Los archivos son iguales cuando tienen el mismo nombre y el mismo contenido. Si en una copia de archivos sustituimos un archivo por otro con el mismo nombre, podemos estar perdiendo información.

También puedes copiar archivos y carpetas desde la carpeta origen hasta la carpeta de destino arrastrándolos de una ventana a otra de Archivos. Si mantenemos pulsada la tecla Ctrl durante el arrastre de los archivos y antes de soltar el botón del ratón estamos copiando los archivos seleccionados.

Para este caso es muy útil tener varias pestañas de contenido abiertas, lo cual nos permite arrastrar un archivo de uno a otro para copiarlo. Cuando arrastremos una selección de archivos para copiarla tendremos que llevarla a la etiqueta de la pestaña de destino para que aparezca esta pestaña y entonces soltar la selección ahí.

Si los archivos se copian de entre dispositivos de almacenamiento, entonces no es necesario mantener pulsada la tecla Ctrl.

#### 4.2.5 Mover archivos

El movimiento de archivos implica llevar un conjunto de archivos de una carpeta a otra. Desaparecen de la carpeta origen y se crearán en la carpeta de destino con el mismo nombre. Para mover archivos seguir los siguientes pasos:

1. Situar en la carpeta donde están los archivos a mover.



2. Seleccionar los archivos que queremos mover.
3. Hacer una de las siguientes acciones:
  - a) Elegir la opción de menú *Cortar*.
  - b) Pulsar la combinación de teclas Ctrl+X
  - c) Hacer clic sobre el botón derecho del ratón sobre cualquier archivo de la selección y seleccionar la opción *Cortar*.
4. Navegar a la carpeta de destino.
5. Realizar una de las siguientes acciones:
  - a) Elegir la opción de menú *Pegar*.
  - b) Pulsar la combinación de teclas Ctrl+V
  - c) Hacer clic sobre el botón derecho del ratón sobre cualquier archivo de la selección y seleccionar la opción *Pegar*.

Si movemos una carpeta, también movemos su contenido, incluyendo subcarpetas y archivos. Si en la carpeta de destino existen archivos con el mismo nombre que los archivos movidos, se comporta de la misma manera que hace con la copia de archivos.

Al igual que antes también podemos mover archivos de una ubicación a otra arrastrando una selección desde una ventana de Archivos a otra o desde una pestaña a otra. Sin embargo, en esta ocasión no hay que mantener pulsada la tecla Ctrl mientras arrastramos ya que de lo contrario estaríamos copiando los archivos.

#### 4.2.6 Eliminar archivos

Cuando borramos archivos, estos se mueven a la Papelera de reciclaje, de donde podremos recuperarlos posteriormente. La papelera de reciclaje no es más que un espacio reservado en el disco duro para que en caso de haber eliminado algún elemento que nos era necesario podamos recuperarlo. Una vez vaciemos la papelera ya no podremos recuperar los archivos. Para borrar un archivo o carpeta seguir los siguientes pasos:

1. Situar en la carpeta donde están los archivos a borrar.
2. Seleccionar los archivos que queremos borrar.
3. Hacer una de las siguientes acciones:
  - a) Pulsar la tecla Supr
  - b) Hacer clic sobre el botón derecho del ratón sobre cualquier archivo de la selección y seleccionar la opción *Mover a la papelera*.

Si borramos una carpeta, se está borrando todo su contenido, incluyendo subcarpetas y archivos.

Sin embargo, también podemos borrar un archivo definitivamente sin que pase por la papelera, sino que se borra del sistema de archivos inmediatamente. La opción de menú *Eliminar* sólo está disponible si tenemos seleccionada la opción *Incluir un comando Eliminar que no use la Papelera* en las preferencias de Archivos.

Para borrar un archivo o carpeta realizar los siguientes pasos:

1. Seleccionar los archivos a borrar.
2. Realizar alguna de las siguientes opciones:
  - a) Elegir la opción de menú *Eliminar*.
  - b) Hacer clic con el botón derecho del ratón sobre el archivo o carpeta y después elija *Eliminar*.
  - c) Pulsar la combinación de teclas Shift+Supr.

#### 4.2.7 Renombrar archivos

Podemos cambiar el nombre de un archivo o carpeta. Para ello seguir los siguientes pasos:

1. Situar en la carpeta donde está el archivo a renombrar.
2. Seleccionar el archivo que queremos cambiar el nombre.
3. Hacer una de las siguientes acciones:
  - a) Pulsar la tecla F2
  - b) Hacer clic sobre el botón derecho del ratón sobre cualquier archivo de la selección y seleccionar la opción *Cambiar nombre...*

No importa cuál de las tres formas utilizemos, debemos de apreciar que el nombre se ha editado. Está en modo edición, eso quiere decir que podemos escribir en ese cuadro. Escribiremos el nuevo nombre y pulsaremos la tecla Intro o haremos clic en cualquier zona de la ventana que no sea el archivo renombrado, para que los cambios se hagan efectivos.

En GNU/Linux no existen las extensiones de archivo pero si el archivo tiene una lo utiliza para identificar el tipo y ponerle el icono adecuado. Al editar el nombre solamente lo hace hasta el punto, pero podemos cambiar la extensión si queremos. Si renombramos un archivo debemos mantener la extensión tal cual estaba. En caso contrario es posible que al intentar ejecutarlo no funcione. Esto se solucionaría escribiendo de nuevo su extensión.

#### 4.2.8 Propiedades de archivos y carpetas

Tanto las carpetas como los archivos tienen sus propias características o propiedades: el tamaño, la ubicación, la fecha de creación, etc. Podemos ver en un momento dado las propiedades de un archivo o carpeta concretos, de forma más ampliada. Para conocer las características de una carpeta o archivo podemos hacer clic sobre él con el botón derecho del ratón y elegir la opción *Propiedades* del menú contextual.

Aparecerá una ventana con varias pestañas. Las pestañas que aparezcan dependerán del tipo de archivo.



Figura 12.- Propiedades de archivo

La pestaña *Básico* es la más utilizada. Contiene información sobre:

- ✓ Tipo.- Describe con qué tipo de elemento estamos tratando.
- ✓ Tamaño.- Aparece el tamaño en la unidad de medida más apropiada para su tamaño, además de en bytes entre paréntesis. Si vemos las propiedades de una carpeta, el tamaño corresponderá a la suma de los archivos que contiene.
- ✓ Lugar.- La ruta donde está guardado.
- ✓ Accedido.- Propiedad no disponible para carpetas. Es la fecha de la última vez que se abrió el archivo, independientemente de que se hicieran cambios o no.
- ✓ Modificado.- Propiedad no disponible para carpetas. Es la fecha de la última modificación.

La pestaña *Permisos* permite especificar los permisos de lectura y escritura sobre el archivo, de modo que podamos, por ejemplo, protegerlo de modificaciones. En un capítulo posterior se verán los permisos y su gestión.

La pestaña *Abrir con* se emplea para gestionar una lista de aplicaciones que pueden abrir el archivo.

La cuarta pestaña puede variar dependiendo del tipo de archivo. En documentos aparecerá la pestaña *Documento*, en archivos de sonido aparecerá *Sonido*, en archivos de vídeo aparecerá la pestaña *Sonido/Vídeo*. En cada una de ellas incluirá información sobre el archivo específica de su formato.

## 4.3 Archivos ocultos

Por omisión, Archivos no muestra ciertos archivos de sistema y los archivos de respaldo en las carpetas. Esto impide la modificación accidental o el borrado de ellos, lo que puede perjudicar la operación de su equipo, y también reduce la confusión en lugares como su Carpeta Personal. Archivos no muestra:

- ✓ Archivos ocultos, cuyo nombre empieza por un punto (.),
- ✓ Archivos de respaldo, cuyo nombre termina con una vergulilla (~)
- ✓ Archivos listados en el archivo `.hidden` de una carpeta particular.

Podemos ocultar o mostrar archivos ocultos en una carpeta particular seleccionando *Ver ► Mostrar archivos ocultos*. Para hacer que Archivos siempre muestre los archivos ocultos, ver la sección *Preferencias*.

### 4.3.1 Ocultar un archivo o carpeta

Para ocultar un archivo o carpeta en Archivos, o renombra el archivo para que su nombre comience por el carácter punto (.), o cree un archivo de texto llamado `.hidden` en la misma carpeta, y añada su nombre a él. Quizá sea necesario refrescar la ventana de Archivos para ver el cambio.

## 5 Búsquedas de archivos

La búsqueda de archivos es una herramienta imprescindible en los sistemas operativos de la actualidad. La enorme capacidad de los actuales discos duros unido a la gran cantidad de información que almacenamos obligan a tener una herramienta de este tipo para que el usuario pueda encontrar rápida y fácilmente el archivo que necesita.

Ubuntu incorpora la función de búsqueda la cual permitirá encontrar los archivos que necesitemos rápidamente. Disponemos de dos opciones para realizar la búsqueda: el panel lugares, donde podremos buscar tanto carpetas y archivos como aplicaciones; y Archivos, donde solamente buscamos carpetas y archivos.

### 5.1 Búsqueda con Archivos

Para buscar archivos o carpetas podemos utilizar el botón de búsqueda que se encuentra en el extremo derecho de la barra de herramientas del Archivos. La forma de realizar una búsqueda es la siguiente:

1. Abrir Archivos y situarnos en la carpeta desde donde queremos comenzar la búsqueda.
2. Hacer clic en el botón de búsqueda en la barra de herramientas.
3. Escribir una cadena de texto en el cuadro *Buscar*. Se buscarán todos los archivos en cuyo nombre o contenido aparezca la cadena de texto escrita.
4. Hacer clic en el icono de lupa que hay en el extremo derecho del cuadro de texto *Buscar*.

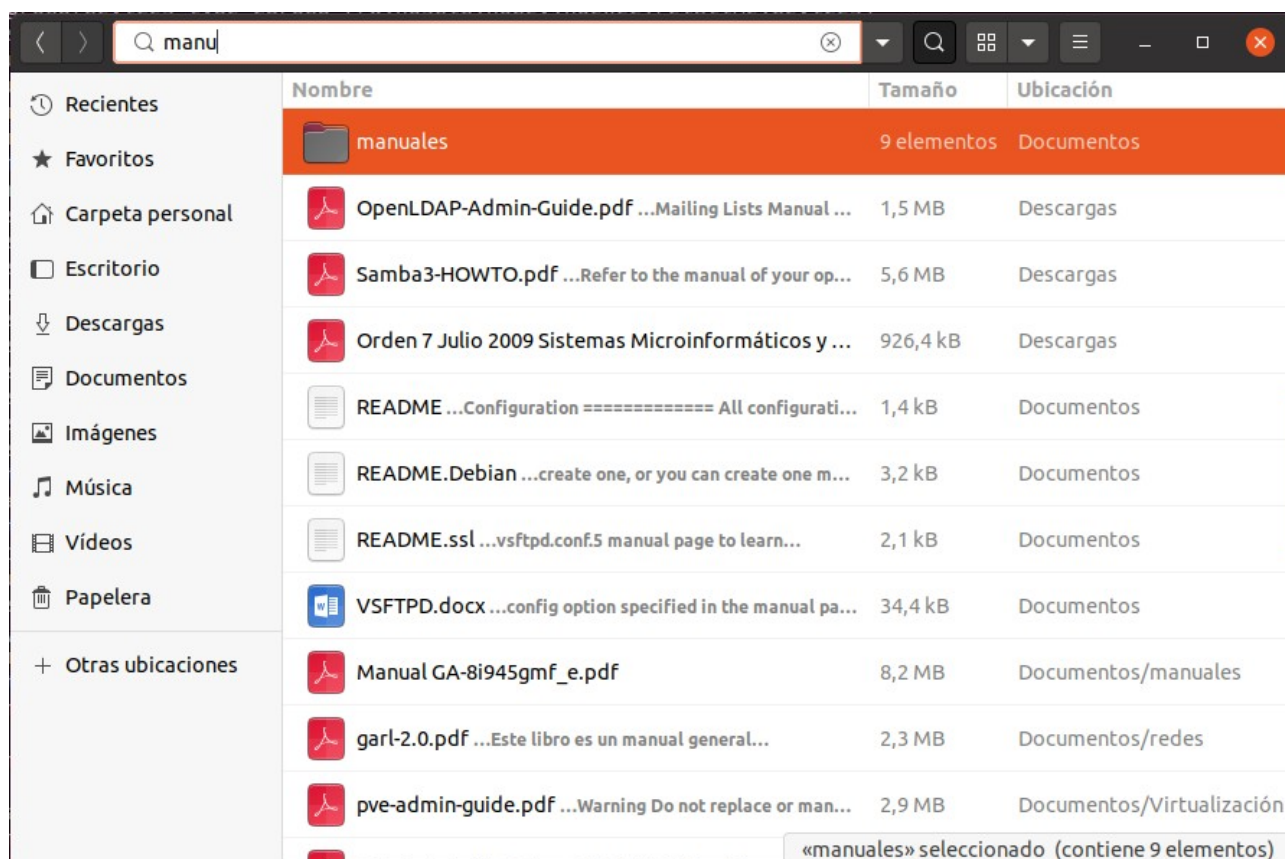


Figura 13.- Búsqueda de archivos

El primer paso es situarnos en la carpeta adecuada, esto es importante porque cuando realizamos una búsqueda mediante esta caja sólo se busca en la carpeta en que estamos situados y en sus subcarpetas. Es decir, si estamos en la carpeta *Música* y buscamos un archivo de la carpeta *Imágenes*, no lo encontrará. En cambio, si la carpeta *Música* tiene 30 carpetas con canciones en su interior, sí que podríamos encontrar en ellas el título que hayamos indicado.

No es necesario, obviamente, que vayamos a la carpeta exacta donde el archivo está situado, porque si lo supiéramos no necesitaríamos buscar nada, pero sí debemos abrir una carpeta que tenga el archivo en alguno de sus subniveles.

Como vemos en la imagen anterior, el resultado de la búsqueda lo muestra en el panel de contenido. Aparecen todos los archivos y carpetas cuyo nombre o contenido de archivo

contiene la cadena introducida en el cuadro de texto *Buscar*, independientemente de si se tecleo en mayúsculas o minúsculas.

Los resultados de la búsqueda aparecerán en el panel de contenido de Archivos, y por lo tanto las operaciones que podemos realizar sobre estos archivos o carpetas resultado de una búsqueda son las mismas que sobre cualquier otra carpeta o archivo de Archivos. Podremos copiar, eliminar, cambiar de nombre, abrir el archivo, elegir el tipo de vista, etc.

## 5.2 Búsqueda con el botón aplicaciones

La barra de búsqueda en el botón Aplicaciones también permite buscar aplicaciones y archivos mostrándote los que has usado recientemente. Si has trabajado con un documento o editado una imagen y olvidaste donde las guardaste encontrarás útil esta herramienta.

El botón Aplicaciones incluye una barra de búsqueda para aplicaciones, lanzadores de las aplicaciones más utilizadas. La barra de búsqueda es global lo que significa que puedes usarla para buscar tanto archivos como aplicaciones. Para usarla haz clic en el botón Aplicaciones y comienza a escribir en la barra de búsqueda. Los nombres e iconos de aplicaciones relevantes y ficheros aparecerán tan pronto como introduzcas texto. Al hacer clic sobre el archivo abrirá el archivo.

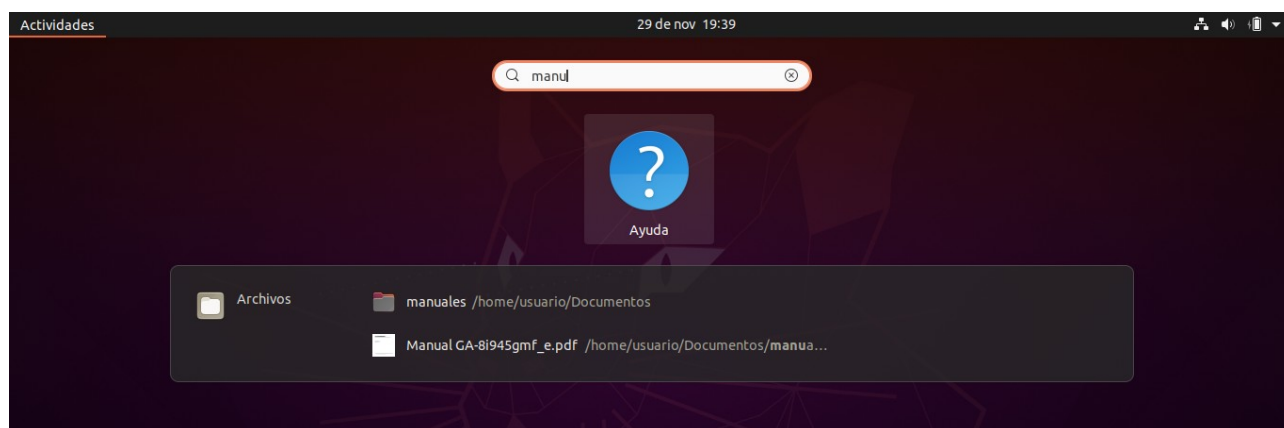


Figura 14.- Búsqueda de archivos con el tablero

## 6 La papelera de reciclaje

La papelera no es más que una carpeta donde se almacena los archivos o carpetas que eliminamos, de forma que podamos recuperarlos. Por tanto, es un paso intermedio antes del borrado definitivo.

Si tratamos de recuperar un elemento eliminado que se encuentre en la papelera de reciclaje por defecto volverá a su ubicación original. Es decir si eliminamos un archivo situado en la carpeta *Documentos* cuando lo restauremos lo volverá a guardar en esa carpeta.

La papelera mantendrá los documentos eliminados de forma permanente, a menos que la vaciemos o borremos alguno de los elementos que contiene de forma manual. Si borramos el contenido de la papelera ya no podremos recuperarlo, así que debemos tener cuidado y observar bien qué contiene antes de vaciarla.

El icono de la papelera es descriptivo. Si no contiene elementos eliminados, el dibujo es una papelera vacía, si en cambio contiene algún elemento, se mostrará con papeles en su interior.



Figura 15.- Iconos de la papelera

Para abrir la papelera podemos utilizar dos métodos:

1. Haz clic en el último icono del Lanzador.
2. Abre Archivos y en el panel lateral haz clic sobre el elemento *Papelera*.

Verás una ventana similar a la siguiente:

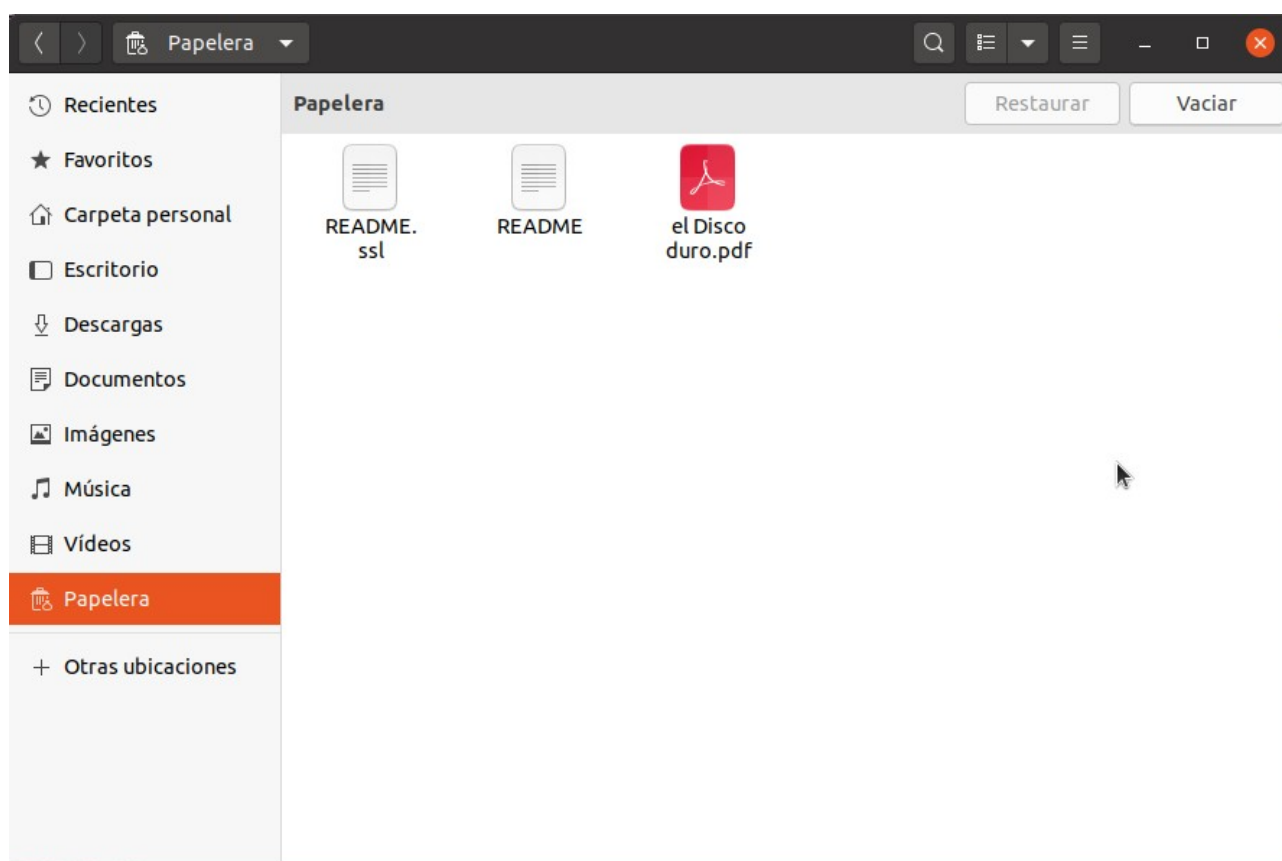


Figura 16.- Papelera de reciclaje

Se trata de una ventana más de Archivos pero en la parte superior del panel de contenido contiene los botones *Vaciar*, que permite eliminar definitivamente su contenido y *Restaurar* los elementos seleccionados, que devuelve los archivos seleccionados a su ubicación original.

Los archivos son accesibles. Podemos visualizar el contenido de un documento o los elementos de una carpeta haciendo clic sobre ellos.

## 6.1 Restaurar elementos eliminados

Tenemos dos opciones:

- ✓ Restaurar archivos o carpetas a su lugar de origen.- Lo haremos mediante el botón correspondiente del panel de contenido. Seleccionamos los archivos y hacemos clic en el botón *Restaurar* los elementos seleccionados.
- ✓ Restaurar archivos o carpetas en un lugar distinto del que se eliminó.- Lo haremos simplemente moviendo el contenido desde la papelera hasta la carpeta que queramos, como ya hemos aprendido.

## 6.2 Eliminar permanentemente

Antes de eliminar algún elemento de la papelera de reciclaje has de tener muy claro que luego no podrás recuperarlo. Para eliminarlos debes:

1. Abrir la papelera de reciclaje.
2. Seleccionar qué elementos quieres borrar definitivamente.
3. Eliminarlos. Se eliminan exactamente igual que el resto de archivos. Podemos pulsar la tecla *Supr* o hacer clic con el botón derecho del ratón sobre el archivo y seleccionar la opción *Eliminar permanentemente*.

Es posible que queramos borrar un archivo definitivamente desde el principio. En ese caso sería una pérdida de tiempo enviarlo a la papelera para luego ir a buscarlo allí y borrarlo definitivamente. Para borrar un archivo permanentemente sin pasar por la papelera de reciclaje simplemente debemos pulsar la tecla *Mayúsculas* (*Shift*) mientras estamos pulsando la opción *Eliminar* o la tecla *Supr*.

## 6.3 Vaciar la papelera

La papelera va acumulando elementos eliminados, como hemos dicho, de forma indefinida. Esto significa que están disponibles para su recuperación, pero también implica que ocupan espacio en nuestro disco duro. Por lo tanto, es recomendable eliminar permanentemente aquella información que ya no nos vaya a ser de utilidad, especialmente cuando se trate de archivos que ocupan más espacio, como vídeos.

Para vaciar la papelera de todo su contenido debemos:

1. Abrir la papelera de reciclaje.
2. Pulsar el botón del panel de contenido que se encarga de eliminar todo el contenido.
3. Confirmar que queremos borrarlos pulsando *Vaciar* en el cuadro de diálogo.

También podemos vaciarla sin abrirla, lo único que hay que tener en cuenta en este caso es que no vemos qué estamos borrando, por lo que es recomendable hacerlo sólo cuando conocemos el contenido y sabemos que no hay problema en eliminarlo. Los pasos serían:



1. Hacer clic con el botón derecho del ratón sobre el icono de la papelera en el lanzador.
2. Elegir la opción *Vaciar la papelera* del menú contextual.
3. Confirmar que queremos borrarlos pulsando *Vaciar la papelera* en el cuadro de diálogo.

## 7 Almacenamiento en la nube

---

Actualmente, como sistema de almacenamiento de información se están imponiendo los servicios en la nube que algunas multinacionales tecnológicas están ofreciendo a sus usuarios. Existen diversas opciones, y conforme pasa el tiempo, estas se están ampliando. Cada servicio de almacenamiento difiere principalmente en la cantidad de espacio que ofrece a cada usuario y/o el coste, aunque algunos de ellos son gratuitos.

En este epígrafe vamos a ver como podemos disponer en nuestro PC de acceso al almacenamiento con dos de los servicios en la nube más populares: Google Drive y DropBox.

### 7.1 Google Drive

La mayoría de las distribuciones Linux incluye una gestión de las cuentas en línea de los usuarios con diferentes proveedores de servicios en la nube. Una de ellas es Google Drive, con la que podremos acceder a nuestro espacio de almacenamiento empleando nuestra cuenta de Google.

Para poder hacer esto tenemos que configurar nuestro usuario y clave de acceso en la configuración del equipo. A partir de entonces, podemos acceder a él desde el administrador de archivos y también desde alguna aplicación, como LibreOffice.

Para configurar nuestra cuenta Google seguimos los siguientes pasos:

1. Hacer clic en *Configuración* del tablero y entrar en *Cuentas en línea* o buscar este complemento de administración directamente en *Aplicaciones*.
2. De la lista de proveedores de servicios en la nube seleccionar *Google*.
3. Aparece el cuadro de diálogo para abrir sesión. Introducir la dirección de correo del usuario Google y pulsar Intro.

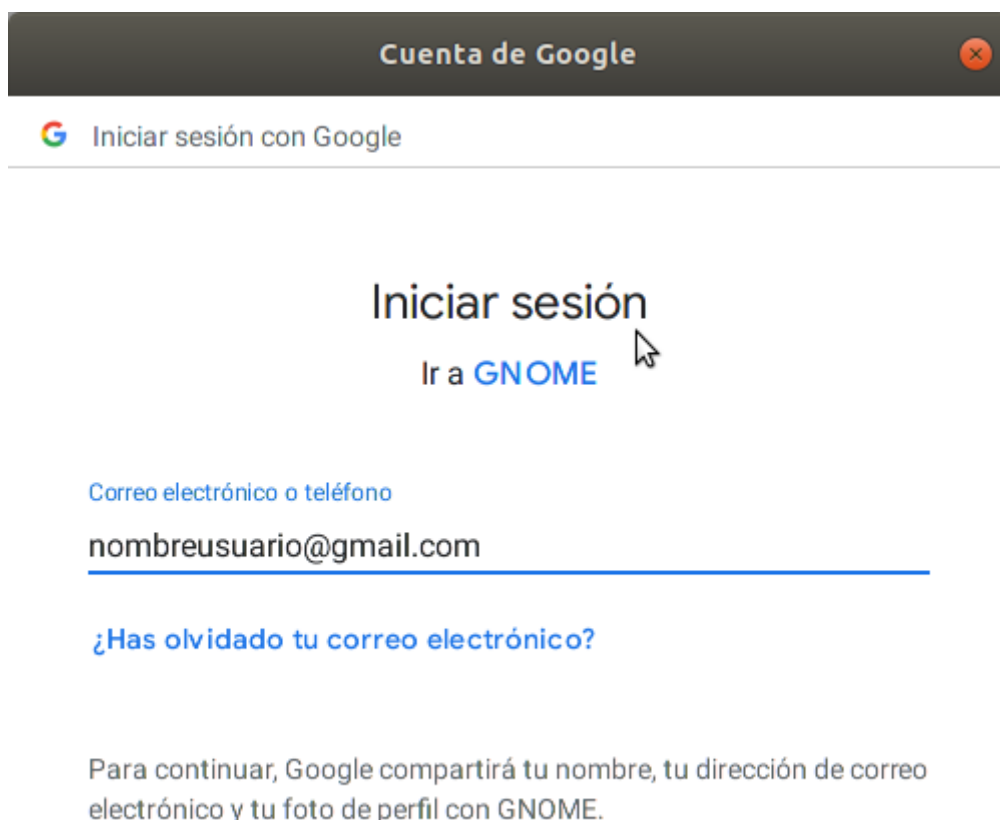


Figura 17.- Nombre de usuario Google

4. Introducir la contraseña y pulsar Intro.
5. Si las credenciales de usuario son correctas, Google solicitará permiso para acceder a tus datos en la nube mediante la aplicación de GNOME. Hacemos clic en el botón *Permitir*.
6. Aparecerá un cuadro de diálogo en el que podremos configurar el acceso a los diferentes servicios de Google desde GNOME (correo, calendario, contactos, ...). Por defecto está todo activado, pero podemos desactivar algunos de los usos de la cuenta. Cuando hayamos terminado cerramos la ventana.

Listo, si abrimos el administrador de archivos veremos que disponemos en el panel izquierdo de un elemento con nuestra cuenta de Google. Si hacemos clic en él veremos el contenido de nuestro espacio de almacenamiento en Google Drive. Desde aquí podremos gestionar nuestros archivos fácilmente, descargarlos copiándolos a una carpeta local o incluso editándolos directamente.

Si en algún momento queremos desvincular nuestro PC con la cuenta de Google, solamente tenemos que ir al complemento *Cuentas en línea* de *Configuración*, seleccionar la cuenta y quitarla.

## 7.2 DropBox

Ubuntu no incluye una cuenta en línea del proveedor DropBox, pero afortunadamente podemos instalar un complemento que nos permitirá acceder a nuestro espacio de almacenamiento de este proveedor de servicios en la nube. Para configurar DropBox en nuestro equipo seguir el siguiente proceso:

1. Descargar el complemento de DropBox para Ubuntu en [https://www.dropbox.com/es\\_ES/install-linux](https://www.dropbox.com/es_ES/install-linux).
2. Desde el administrador de archivos ir a la carpeta de descarga y hacer doble clic sobre el archivo.
3. Hacer clic en el botón *Instalar* e introducir la contraseña de usuario.
4. Buscar en aplicaciones *DropBox* y hacer doble clic sobre el icono correspondiente.
5. Aparece un cuadro de diálogo para instalar DropBox. Hacemos clic en el botón *Aceptar*.
6. Cuando termine la instalación se abrirá el navegador para que iniciemos sesión con nuestra cuenta de DropBox.
7. Si hemos tenido éxito con la autenticación tendremos nuestro equipo vinculado con nuestra cuenta de DropBox. Ahora se creará una carpeta dedicada llamada *Dropbox* en tu carpeta personal del usuario. Podemos mantener nuestros datos en esta carpeta para sincronizarlos en todos los dispositivos.

## 8 Gestión de archivos y directorios en CLI

A continuación se relacionan un conjunto de comandos para gestión de archivos y directorios. Sin embargo, es necesario aclarar antes algunos términos de uso común en todos ellos y que hacen referencia al nombre de archivo o directorio que puede aparecer como argumento de un comando.

El directorio activo es aquél en el que nos encontramos en el momento de ejecutar el comando. Todo usuario tiene un directorio o carpeta personal denominado *HOME*. Si un usuario abre una sesión gráfica dispone de un icono en el lanzador para acceder a su carpeta personal. Lo habitual es que esta carpeta tenga su localización en */home/login*, siendo *login* el nombre de la cuenta de usuario. Si este usuario abre una ventana de terminal el directorio activo será el directorio *HOME* y si abrió una sesión en una consola texto también.

La mayoría de los siguientes comandos tienen nombres de archivos o directorios como argumentos, los cuales podrán ser especificados de forma absoluta o relativa al directorio activo. Es decir, si el nombre se especifica utilizando la ruta completa se hará desde el directorio raíz */* y continuando por la ruta de directorios hasta finalizar en el nombre del archivo, separándolos por */*. Si el nombre es relativo el nombre se compone de la misma forma, pero comienza desde el directorio activo, que ya vimos se referencia con un punto, el

cual se puede omitir, y a partir de ahí se continúa la ruta de directorios separándolos también por `/`.

Por ejemplo, supongamos que el usuario `mluque` tiene en su directorio `HOME` un directorio que se llama `trabajos` y dentro de esto otro que se llama `oficina`. Si en este directorio contiene un archivo cuyo nombre es `memoria.odt` el nombre completo de este archivo será:

```
/home/mluque/trabajos/oficina/memoria.odt
```

El nombre relativo dependerá del directorio activo en el momento de referirnos a él. Si este directorio es el directorio personal, el nombre relativo del archivo será:

```
./trabajos/oficina/memoria.odt
```

Observe que el punto inicial sustituye al directorio activo. Ya vimos en los fundamentos del sistema de archivos que todo directorio tiene dos directorios implícitos, el punto que hace referencia al directorio y los dos puntos que hace referencia al directorio padre. En el ejemplo anterior el punto inicial se refiere al directorio `/home/mluque`.

Podemos deducir que un archivo tiene muchos nombres relativos, ya que el directorio activo en un momento determinado puede ser cualquier directorio de toda la jerarquía de almacenamiento. Siguiendo con el ejemplo anterior, si el directorio activo fuera `trabajos`, el nombre relativo del archivo sería:

```
./oficina/memoria.odt
```

Finalmente, si el directorio activo es aquél donde se encuentra el archivo, podemos hacer referencia al mismo usando únicamente su nombre, pero de forma relativa podríamos anteponerle el punto y la barra de dividir. Si el directorio activo fuera `oficina` el nombre relativo del archivo anterior sería:

```
./memoria.odt
```

Todo lo anterior es válido cuando el archivo al que se desea hacer referencia está por debajo del directorio activo en la jerarquía de almacenamiento. Cuando se encuentra por encima o en otra rama distinta del árbol, entonces tendremos que emplear los dos puntos que hacen referencia al directorio padre para formar el nombre relativo del archivo.

El caso más simple sería aquél en el que el archivo se encuentra en el directorio padre del directorio activo. Si el directorio activo fuera `/home/mluque/trabajos/oficina/apuntes` y quisiéramos acceder al archivo `trabajo.odt` para componer el nombre relativo del archivo anterior pondríamos lo siguiente:

```
../trabajo.odt
```

ya que `trabajo.odt` se encuentra en el directorio padre de `apuntes`. Si tuviéramos que nombrar de forma relativa al archivo `informe.odt` que se encuentra en `/home/mluque` lo haríamos así:

```
../.././informe.odt
```

Vemos que hemos empleado los dos puntos cada vez que necesitamos subir un nivel en el árbol de directorios. Los dos primeros hacen referencia al directorio `oficina`, los dos siguientes al directorio `trabajos` y finalmente los dos últimos a `mluque`.

Si tenemos necesidad de acceder a un archivo que se encuentra en una rama diferente del árbol de la que se encuentra el directorio activo, se haría de la misma forma. Por ejemplo, se necesita acceder al archivo `/home/ajimenez/documentos/manual.txt` y el directorio activo es `/home/mluque/trabajos`. El nombre relativo del archivo sería:

```
../../ajimenez/documentos/manual.txt
```

A la hora de especificar el nombre de un archivo como argumento a un comando es indiferente que se haga de forma absoluta o relativa.

## 8.1 Metacaracteres en los nombres de archivo

A menudo se desea que un comando funcione sobre un grupo de archivos sin tener que escribirlos todos. La expansión de los nombres de archivo que utiliza los metacaracteres o caracteres comodín del intérprete de comandos facilita esta tarea.

Cuando un nombre incluye uno o varios caracteres comodín, se está construyendo un patrón al que se puede ajustar los nombres de un conjunto de archivos, es decir, estos archivos estarían representados por este nombre único. De esta forma, se podría hacer que un comando actuase sobre todos ellos sin necesidad de tener que indicar una lista con el nombre de cada uno explícitamente.

Los caracteres comodín más habituales son:

- ✓ `*` Coincide con cualquier grupo de 0 o más caracteres.
- ✓ `?` Coincide exactamente con un único carácter.
- ✓ `[...]` Coincide exactamente con un único carácter de los encerrados entre corchetes.
- ✓ `[a-z]` Coincide exactamente con un único carácter de la lista comprendida entre a y z.
- ✓ `[!...]` Coincide exactamente con un único carácter diferente a los encerrados entre corchetes.
- ✓ `{...}` Contiene una lista de uno o más caracteres separados por comas. Cada elemento de la lista se utiliza por turnos para expandir un nombre de archivo que iguala la expresión completa en la que está inmersa las llaves.

## 8.2 Crear directorios. Comando `mkdir`

El comando `mkdir` se emplea para crear la estructura jerárquica de directorios y

subdirectorios.

### Sintaxis

```
mkdir [opciones] directorio ...
```

### Parámetros

directorio ...

Lista de directorios separada por espacios que se van a crear.

### Opciones

-p, --parents

Crea también los directorios padre que falten, ignorando los que ya pudieran existir.

## 8.3 Cambiar el directorio activo. Comando cd

El comando `cd` se emplea para cambiar el directorio activo y recorrer el árbol de directorios.

### Sintaxis

```
cd [directorio]
```

### Parámetros

directorio

Directorio al que se va a cambiar. Si se omite, se cambiará al directorio HOME del usuario.

Existen tres expansiones de tilde (^) que permiten cambiar a directorios específicos y de mucho uso. Estos son:

Expansión	Descripción
~[Usuario]	Directorio personal del usuario indicado. Si no se expresa nada \$HOME .
~+	Directorio actual ( \$PWD ).
~-	Directorio anterior ( \$OLDPWD ).

En el primer caso el uso de una tilde hace referencia al directorio personal del usuario. Si indicamos también el usuario hacemos referencia a su directorio personal.

El último resulta muy útil ya que permite cambiar al anterior directorio activo en el que estábamos antes del actual.

## 8.4 Eliminar directorios. Comando rmdir

El comando `rmdir` eliminar un directorio vacío. Si no está vacío emitirá un mensaje de error.

```
rmdir [opciones] directorio ...
```

### Parámetros

`directorio ...`

Lista de directorios separada por espacios en blanco que se van a borrar.

### Opciones

`-p, --parents`

Elimina un subárbol completo de directorios vacíos, es decir, el directorio y sus antecesores.

## 8.5 Listar el contenido de un directorio. Comando ls

El comando `ls` se emplea para realizar un listado de los archivos y directorios que contiene un directorio.

### Sintaxis

```
ls [opciones] [archivos ...]
```

### Sintaxis

`archivos ...`

Lista de archivos, separados por espacios, que se quieren listar.

### Opciones

`-l`

Listado largo. La lista de archivos incluye el tamaño total de todos los archivos listados en bloques de 512 bytes y por cada archivo la siguiente información:

- ✓ Tipo de archivo y Permisos
- ✓ N° de Enlaces
- ✓ Propietario
- ✓ Grupo propietario
- ✓ Tamaño en bytes
- ✓ Marca de tiempo con la última modificación del archivo.
- ✓ Nombre

`-t`

Ordena el listado por fecha descendente

-S

Ordena el listado por tamaño descendente

-X

Ordena el listado por la extensión<sup>2</sup> del archivo.

-r

Realiza la ordenación especificada con las opciones anteriores ascendentemente.

-R

Realiza un listado recursivo con el contenido de los directorios. Por cada directorio que se encuentre en la lista hace un listado de su contenido. Esta recursividad es por defecto cuando se especifica en el comando un archivo o archivos a listar.

-a

Incluye en el listado los archivos ocultos, que en Linux comienzan por .

-d

Los directorios los lista como si fueran archivos, es decir, anula la recursividad.

## 8.6 Copiar archivos y directorios. Comando cp

El comando `cp` permite crear copias de archivos existentes.

```
cp [opciones] archivos_origen ... destino
```

### Parámetros

`archivos_origen ...`

Lista de archivos, separada por espacios, que se desean copiar. Se pueden utilizar caracteres comodín para copiar un conjunto de archivos.

`destino`

Si el destino es un nombre de directorio existente, crea un archivo copia por cada archivo de la lista origen. Cada archivo creado tendrá el mismo nombre que el original y se almacenará en el directorio destino.

Si el destino no es un nombre de directorio existente, crea un único archivo que es copia del archivo origen. En este caso no se permite que los archivos origen sean un conjunto, solamente puede ser un único archivo. El nombre del archivo nuevo será el especificado con este argumento y se almacenará en el directorio donde se encuentra el archivo origen.

---

<sup>2</sup> En el sistema operativo MS-DOS la extensión del archivo son los caracteres que aparecen detrás del punto, al final de su nombre. En Linux no existe la extensión de archivo propiamente dicha, el punto se considera un carácter más del nombre. Sin embargo, se sigue utilizando esta convención para identificar el tipo de archivo.



### Opciones

`-i`

Si el archivo destino ya existe, pregunta si se quiere sobrescribir antes de perder la información actual del archivo destino. Por defecto, la copia es destructiva, es decir, elimina el archivo destino si existe antes de crear el nuevo.

`-r, -R`

Realiza una copia recursiva de los directorios. Si la lista de los archivos origen contiene un directorio, copiará su contenido recursivamente. Si se encuentra algún enlace simbólico copia los originales.

`--parents`

Preserva la ruta de directorios, creando los que no existieran. Si el nombre del archivo origen con `./` tiene una ruta y esta no existe en el destino, se creará.

`-p`

Preserva los atributos de archivo como propietario y permisos.

`-u`

Si el archivo destino existe, lo sustituye por el origen si la fecha de modificación de éste es más reciente que la del archivo destino.

## 8.7 Mover archivos y directorios. Comando `mv`

El comando `mv` mueve archivos de su ubicación actual a otro directorio. También se emplea para cambiar el nombre de los archivos.

### Sintaxis

```
mv [opciones] archivos_origen ... destino
```

### Parámetros

`archivos_origen ...`

Lista de archivos separados por espacios que se van a mover. Admite caracteres comodín.

`destino`

Si el destino es un nombre de directorio existente, mueve los archivos origen a este. Si no es un nombre de directorio existente, renombra el único archivo origen que se especifique.

### Opciones

`-f`

No pide confirmación al sobrescribir un archivo destino. Por defecto el comando `mv` sobrescribe el destino si existe.

`-u`

No mueve los archivos origen si existen en el destino y la fecha de modificación es más reciente.

## 8.8 Eliminar archivos y directorios. Comando `rm`

El comando `rm` borra archivos y directorios.

### *Sintaxis*

```
rm [opciones] archivo ...
```

### *Parámetros*

`archivo ...`

Lista de archivos que se van a borrar. Pueden utilizarse caracteres comodín.

### *Opciones*

`-r, -R`

Borrado recursivo. Se emplea cuando se borran directorios sin límite de profundidad.

`-f`

No pide confirmación al usuario para efectuar el borrado.

## 8.9 Mostrar el contenido de archivos de texto. Comandos `cat` y `more`

El comando `cat` se emplea para concatenar archivos texto, aunque se utiliza también para mostrar su contenido, ya que su salida la efectúa por la salida estándar o pantalla.

### *Sintaxis*

```
cat archivo ...
```

### *Parámetros*

`archivo ...`

Lista de archivos que va a concatenar.

El comando `more` permite visualizar archivos texto y además interactuar con el usuario para facilitar la navegación por el mismo.

### *Sintaxis*

```
more [opciones] archivo
```

### *Parámetros*

`archivo`

Nombre del archivo a visualizar.

### Opciones

`-n`

Especifica el número de líneas por pantalla para la visualización.

Cuando se ejecuta el comando `more` sobre un archivo aparece en pantalla las primeras `n` líneas del mismo. A partir de entonces se pueden emplear las siguientes pulsaciones de teclas para la navegación:

- ✓ Intro.- Avanza una línea
- ✓ Barra espaciadora.- Avanza una pantalla
- ✓ q.- Salir
- ✓ /patrón.- Realiza una búsqueda de todas las cadenas representadas por *patrón*.
- ✓ f.- Avanza una pantalla
- ✓ b.- Retrocede una pantalla

## 8.10 Búsqueda de archivos y directorios. Comando find

El comando `find` realiza una búsqueda por un subárbol de almacenamiento y ejecuta una serie de acciones sobre los archivos que encuentra. Debido a la potencia de funcionamiento, tiene una sintaxis algo complicada.

### Sintaxis

```
find directorio inicio [opciones] [acciones]
```

### Parámetros

directorio inicio

Directorio a partir del cual comenzará la búsqueda de los archivos

### Opciones

`-name` patrón

Indica el patrón al que se tienen que ajustar los nombres de los archivos. Para construir el patrón se pueden emplear los caracteres comodín.

`-size` `n[cwbKMG]`

Busca los archivos por su tamaño siendo `n` un número que representa el tamaño del archivo y a continuación se especifica el carácter que representa la magnitud. Esta puede ser:

- ✓ c: bytes

- ✓ w: palabras de 2 bytes.
- ✓ b: Bloques de 512 bytes.
- ✓ K: kilobytes
- ✓ M: Megabytes
- ✓ G: Gigabytes

Además, el tamaño  $n$  puede especificarse como tamaño mínimo, máximo o exacto. Si al número  $n$  se antepone el signo + se indica que el archivo tiene como tamaño mínimo  $n$ . Si se antepone el signo - se indica que el archivo tiene como tamaño máximo  $n$ . Si solamente se especifica  $n$  representa el tamaño exacto de los archivos que busque.

### `-type c`

Busca los archivos por su tipo. El carácter  $c$  especifica el tipo de archivo y puede tomar alguno de los siguientes valores:

- ✓ b: Archivo de dispositivo por bloques.
- ✓ c: Archivo de dispositivo por carácter.
- ✓ d: Directorio.
- ✓ p: Pipe
- ✓ f: Archivo normal.
- ✓ e: Enlace
- ✓ s: Socket

Una vez ha encontrado archivos, puede realizar acciones sobre los mismos. Las acciones más habituales son:

### `-print`

Visualiza por pantalla el nombre completo del archivo encontrado.

### `-delete`

Borra los archivos encontrados.

### `-exec comando \;`

Ejecuta el comando especificado. Todos los archivos encontrados pueden utilizarse como argumentos del comando, empleando para ello '{}'.

## 8.11 Compresión de archivos. Utilidad `gzip`

La utilidad `gzip` comprime archivos. Por cada archivo comprimido genera uno con nombre igual al original pero añadiendo el sufijo `.gz`. El archivo original deja de existir.

### *Sintaxis*

```
gzip [opciones] archivo ...
```

### *Parámetros*

## archivo ...

Lista de archivos a comprimir. Pueden indicarse grupos de archivos con caracteres comodín. En este caso genera un archivo comprimido por cada archivo origen cuyo nombre se ajusta al patrón.

### Opciones

#### -d

Realiza la descompresión del archivo para obtener el original. Similar a utilizar el comando *gunzip*.

#### -1 .. -9

Tasa de compresión. -1 indica la tasa mínima, aumentando la rapidez de la compresión. -9 es la tasa máxima, disminuyendo la rapidez de la compresión. Por defecto se comprimen con -6

## 8.12 Empaquetado de archivos. Comando tar

En Linux, se emplea el comando *tar* para empaquetar un conjunto de archivos en uno sólo. El empaquetado se separa de la compresión para que un usuario pueda elegir entre una amplia gama de compresores de archivo.

Este comando es muy utilizado en las instalaciones de aplicaciones desde código fuente ya que los programas liberados bajo licencia GPL suelen estar empaquetados y comprimidos con el comando *tar*.

### Sintaxis

```
tar [opciones] archivo ...
```

### Parámetros

## archivo ...

Archivos que va a empaquetar. Pueden especificarse caracteres comodín.

### Opciones

#### -c

Crea un archivo empaquetado.

#### -f archivo\_salida.tar

Especifica el nombre de archivo empaquetado. Conviene añadirle la extensión *.tar*

#### -x

Extrae los archivos que se encuentran dentro del archivo empaquetado.

#### -t

Realiza un listado de los archivos que hay dentro del empaquetado.

—z

Realiza la compresión del archivo empaquetado. En este caso conviene añadir la extensión `.tar.gz` al final del nombre del archivo de salida.

## 8.13 Enlaces simbólicos. Comando `ln`

Un enlace es un nombre que se asocia a un archivo para poder hacer referencia al mismo. Es similar a un acceso directo en Windows. Existen dos tipos de enlaces:

- ✓ Duro o físico.- Es un nombre que se añade al archivo.
- ✓ Blando o simbólico.- Es un archivo especial que apunta al archivo original.

Cuando creamos un enlace duro, estamos añadiendo otro nombre al archivo. Por tanto podemos tener un archivo con varios nombres y ninguno de ellos se considera original. Si borramos un archivo que tiene varios nombres, el archivo continúa existiendo, solamente le hemos borrado uno de sus nombres. El archivo se eliminará cuando se elimine el último de sus nombres.

Al crear un enlace simbólico en realidad estamos creando un nuevo archivo, el cual contiene el nombre completo del archivo original. El fichero original no tiene por qué existir, aunque no tendría sentido. El borrado del archivo enlace no afecta al archivo original.

Para crear enlaces se emplea el comando `ln`.

### Sintaxis

```
ln [opciones] archivo_origen enlace
```

### Parámetros

#### `archivo_origen`

Archivo al que se está creado el enlace. Muy recomendable que se especifique con ruta completa, ya que de lo contrario al mover el enlace a otro directorio quedaría inutilizado al perder la referencia.

#### `enlace`

Nombre del enlace que se está creando.

### Opciones

—s

El enlace creado es simbólico. Si se omite, el enlace es duro.

La utilidad de los enlaces radica en la posibilidad de utilizar un nombre, distinto al nombre del archivo, para hacer referencia al mismo. Sin embargo, el comportamiento del sistema en lo que se refiere al uso del archivo varía en función de la operación que se realiza y el tipo de enlace:

- ✓ En una operación de visualización de su contenido (comandos `cat` y `more`) al

emplear el enlace como argumento, se está visualizando el contenido del archivo original, nunca del enlace, independientemente de si es simbólico o duro.

- ✓ En una operación de copia (comando `cp`) si se emplea el enlace como archivo origen de la copia, se estaría copiando el archivo original, no importa si el enlace es duro o simbólico. Si ejecutamos el comando `cp` con la opción `-d` copiamos el enlace en lugar del archivo original, solamente con enlaces simbólicos.
- ✓ En una operación de movimiento (comando `mv`) si se emplea el enlace como archivo origen se mueve éste y no el archivo original, excepto si es un enlace duro, que si se movería el archivo original.
- ✓ En una operación de borrado (comando `rm`) al eliminar un enlace no se elimina el archivo original, sólo el enlace. Si es un enlace duro se estaría borrando uno de los nombres del archivo y hasta que no se elimina su último nombre no se elimina el archivo.

## 8.14 Alias

Un alias es una abreviación de un comando para evitar teclear una larga secuencia de comando y opciones. Cuando empleamos muy frecuentemente un comando que incluye varios argumentos y opciones, podemos definir un alias que evita teclear largas sintaxis de comandos. Por ejemplo, supongamos que realizamos con mucha frecuencia un listado del directorio activo en formato largo, ordenados ascendentemente por tamaño y haciendo pausas por pantalla. Este comando sería

```
ls -l -S -R | more
```

Podemos asignar un alias a este comando y llamarlo simplemente `lm`.

```
alias lm="ls -l -S -r | more"
```

Nótese que entre el alias y el signo igual no hay espacio. Tampoco entre el signo igual y el comando. En este otro ejemplo podemos crear un alias para cambiar al directorio activo anterior. Sería así

```
alias cda="cd ~"
```

La definición del alias se mantiene durante la sesión actual, pero al terminarla el alias desaparece. Si queremos que los alias se mantengan entre sesiones tenemos que incluirlos en el archivo `.bashrc` del directorio personal del usuario para que al abrir una nueva sesión se creen de nuevo y los tengamos disponibles.

También podemos definir alias con comandos que requieren de algún argumento. Por ejemplo

```
alias lm="ls -l -S -r $1 | more"
```

El argumento `$1` es un parámetro similar a los que se emplean en la programación de scripts que se expande con el valor introducido en la línea de comando junto con el alias. El

alias que hemos definido aquí sirve para listar los archivos en formato largo y ordenados ascendentemente de un directorio que se incluya al ejecutar el alias. Si quisiéramos utilizar el alias para listar el contenido del directorio `/usr/share/doc` ejecutaríamos el siguiente comando.

```
lm /usr/share/doc
```

Vemos que el argumento del alias `lm` se expandirá en `$1` reemplazándolo.

En cuanto queramos dejar de usar un alias podemos borrarlos con `unalias`. Por ejemplo.

```
unalias lm
```

## 9 Control de acceso

La gestión del control de acceso es sustancialmente diferente en Sistemas Unix/Linux y en Sistemas Microsoft Windows.

En los Sistemas Unix/Linux, la gestión de los permisos que los usuarios y los grupos de usuarios tienen sobre los archivos se realiza mediante un sencillo esquema de tres tipos de permisos (lectura, escritura y ejecución) aplicables a tres tipos de usuarios (propietario, grupo propietario y resto). Este sencillo esquema se desarrolló en los años 70 y aún hoy resulta adecuado para la gran mayoría de los sistemas que podamos encontrar en cualquier tipo de organización. Es cierto que tiene algunas limitaciones pero la ventaja de ser sencillo hace que su administración sea fácil y su rendimiento muy elevado.

En los Sistemas Microsoft Windows, la gestión de los permisos que los usuarios y los grupos de usuarios tienen sobre los archivos, se realiza mediante un complejo esquema de listas de control de acceso (ACLs = *Access Control Lists*) para cada carpeta y cada archivo. El sistema de ACLs tiene la ventaja de ser mucho más flexible que el sistema Unix ya que se pueden establecer más tipos de permisos, establecer permisos solo a algunos usuarios y algunos grupos, denegar permisos, etc., pero como hemos comentado anteriormente, en la mayoría de los casos, con las prestaciones del Sistema Unix es suficiente. En el lado contrario, el sistema de ACLs es más complejo de administrar y más lento ya que antes de acceder a las carpetas o archivos, el sistema debe comprobar listas de permisos mientras que en Unix hace una operación lógica de los bits que especifican los permisos lo cual es muchísimo más rápido.

### 9.1 Usuario propietario y grupo propietario de un archivo

Anteriormente se ha comentado que en Linux todos los archivos pertenecen obligatoriamente a un usuario y a un grupo. Cuando un usuario crea un nuevo archivo, el propietario del archivo será el usuario que lo ha creado y el grupo del archivo será el grupo principal de dicho usuario.

Ejemplo, si un usuario llamado `maria` cuyo grupo principal es el grupo `maria` crea un nuevo archivo, el propietario del archivo será `maria` y el grupo propietario del archivo será `maria`, o lo que es lo mismo, el archivo pertenecerá al usuario `maria` y al grupo `maria`.



Obligatoriamente, todos los archivos pertenecen a algún usuario y a algún grupo.

Con el comando `ls` y la opción `-l` (formato largo) podemos visualizar el usuario propietario y el grupo propietario del archivo, por ejemplo:

```

usuario@Ubuntu11:~$ ls -l
total 36
drwxr-xr-x 2 usuario usuario 4096 2011-08-29 18:24 Descargas
drwxr-xr-x 2 usuario usuario 4096 2011-08-21 18:44 Documentos
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Escritorio
-rw-r--r-- 1 usuario usuario 179 2011-08-19 11:51 examples.desktop
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Imágenes
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Música
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Plantillas
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Público
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Vídeos
usuario@Ubuntu11:~$
  
```

Figura 18.- Listado de archivos en formato largo

Comprobamos que el usuario propietario es usuario y el grupo propietario es usuario. La misma información podemos verla desde el Nautilus. Para ello hay que poner el panel de visualización en formato de lista y activar la visualización de la columna *Permisos*. Esto último lo conseguiremos ejecutando la opción de menú *Ver → Columnas visibles...* y activando *Permisos*.

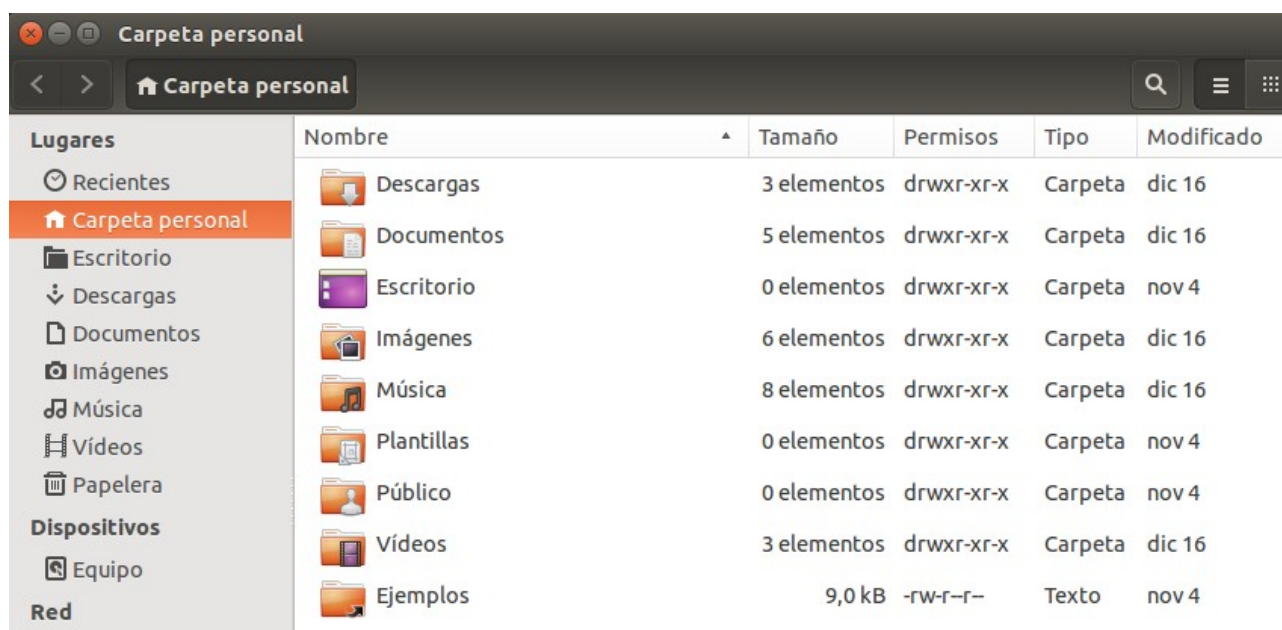


Figura 19.- Listado de archivos en Nautilus con la columna permisos

## 9.2 Tipos de permisos

En los Sistemas Unix/Linux, la gestión de los permisos que los usuarios y los grupos de usuarios tienen sobre los archivos y las carpetas, se realiza mediante un sencillo esquema de tres tipos de permisos que son:

- ✓ Permiso de lectura

- ✓ Permiso de escritura
- ✓ Permiso de ejecución

En los listados anteriores, vemos que los permisos se simbolizan con un conjunto de 9 caracteres con el siguiente formato

**Tipo de archivo:**

(-) para archivos normales

(d) para carpetas (directory)

(l) para enlaces (link)

(s)=socket, (p)=tubería (pipe), (b)=dispositivo de bloque.

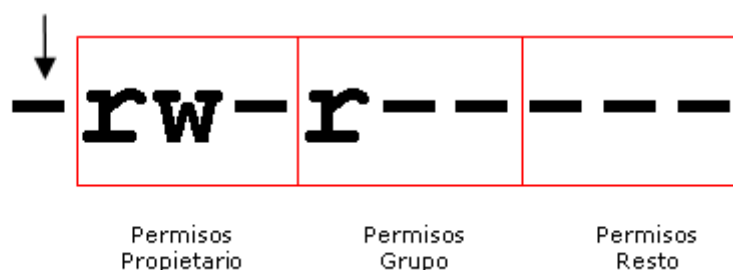


Figura 20.- Esquema de permisos Unix/Linux

El significado de estos permisos difiere si se tienen sobre archivos o sobre carpetas. A continuación veremos el significado para cada uno de los casos:

### 9.2.1 Permisos de lectura

Cuando un usuario tiene permiso de lectura de un archivo significa que puede leerlo o visualizarlo, bien sea con una aplicación o mediante comandos. Ejemplo, si tenemos permiso de lectura sobre el archivo `documento.txt`, significa que podemos ver el contenido del archivo. Si el usuario no tiene permiso de lectura, no podrá ver el contenido del archivo.

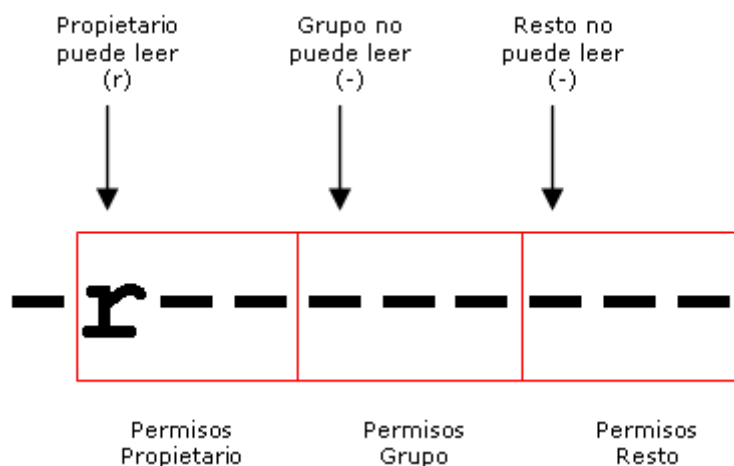


Figura 21.- Permiso de lectura

Cuando un usuario tiene permiso de lectura de una carpeta o directorio, significa que puede visualizar el contenido de la carpeta, es decir, puede ver los archivos y carpetas que contiene, bien sea con el comando `ls` o con un explorador de archivos como Nautilus. Si el usuario no tiene permiso de lectura sobre la carpeta, no podrá ver lo que contiene. El

permiso de lectura se simboliza con la letra **r** del inglés 'read'.

### 9.2.2 Permiso de escritura

Cuando un usuario tiene permiso de escritura sobre un archivo significa que puede modificar su contenido o borrar el archivo. También le da derecho a cambiar los permisos del archivo así como cambiar su propietario y el grupo propietario. Si el usuario no tiene permiso de escritura, no podrá modificar el contenido del archivo.

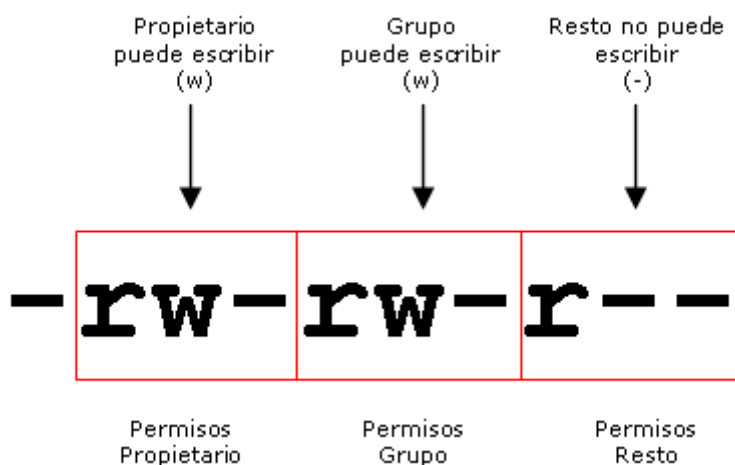


Figura 22.- Permiso de escritura

Cuando un usuario tiene permiso de escritura sobre un directorio, significa que puede modificar el contenido del directorio, es decir, puede crear y eliminar archivos y otros directorios dentro de él. Si el usuario no tiene permiso de escritura sobre el directorio, no podrá crear ni eliminar archivos ni directorios dentro de él.

El permiso de escritura se simboliza con la letra **w** del inglés 'write'.

### 9.2.3 Permiso de ejecución

Cuando un usuario tiene permiso de ejecución de un archivo significa que puede ejecutarlo. Si el usuario no dispone de permiso de ejecución, no podrá ejecutarlo aunque sea una aplicación.

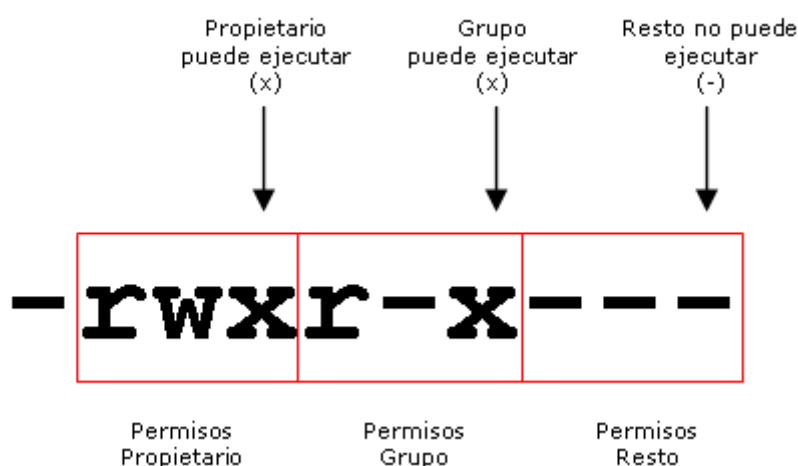


Figura 23.- Permiso de ejecución

Los únicos archivos ejecutables son las aplicaciones y los archivos de comandos (scripts). Si tratamos de ejecutar un archivo no ejecutable, dará errores.

Cuando un usuario tiene permiso de ejecución sobre una carpeta, significa que puede entrar en ella, bien sea con el comando `cd` o con un explorador de archivos como Nautilus. Si no dispone del permiso de ejecución significa que no puede ir a dicha carpeta.

El permiso de ejecución se simboliza con la letra x del inglés 'eXecute'.

#### 9.2.4 ¿A quién se puede otorgar permisos?

Los permisos solamente pueden ser otorgados a tres tipos o grupos de usuarios:

- ✓ Al usuario propietario del archivo
- ✓ Al grupo propietario del archivo
- ✓ Al resto de usuarios del sistema (todos menos el usuario propietario y los usuarios que estén en el grupo propietario)

Se pueden dar permisos de lectura, escritura, ejecución o combinación de ambos al usuario propietario del archivo, al grupo propietario del archivo o al resto de usuarios del sistema. En GNU/Linux no existe la posibilidad de asignar permisos a usuarios concretos ni a grupos concretos, tan solo se puede asignar permisos al usuario propietario, al grupo propietario o al resto de usuarios.

Ejemplo, si disponemos de un archivo llamado `documento.txt` cuyo propietario es `maria` y cuyo grupo propietario es `maria`, se pueden dar permisos de lectura, escritura, ejecución o combinación de estos al usuario `maria`, a los usuarios que pertenecen al grupo `maria` y al resto de usuarios, pero no podremos dar permisos a otros usuarios distintos de `maria` (`jose`, `rafael`,...) ni a otros grupos (`directivos`, `personal`,...) ya que el esquema no lo permite.

Supongamos que la siguiente figura representa los permisos de `documento.txt`:

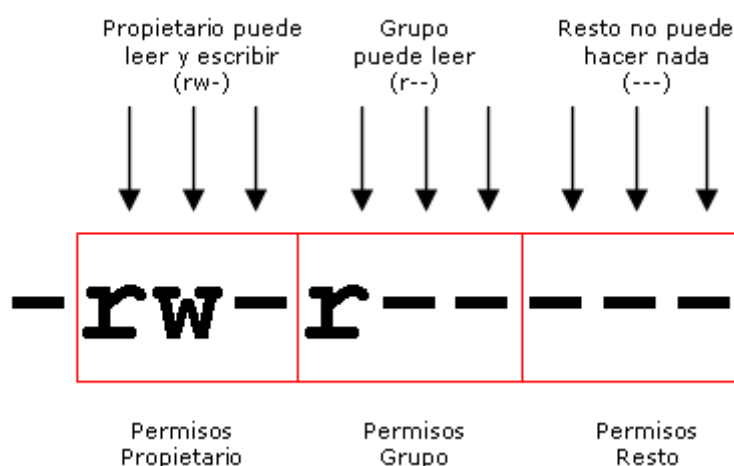


Figura 24.- Permisos del archivo *documento.txt*

El usuario propietario usuario podrá leer y escribir en el documento. Los pertenecientes al grupo usuario podrán leerlo y el resto no podrá hacer nada.

Si deseo que otros usuarios tengan algún permiso sobre el archivo *documento.txt*, no me quedará más remedio que incluirlos en el grupo *maria* u otorgar el permiso al resto de usuarios pero si hago esto último, absolutamente todos los usuarios del sistema gozarán del permiso, por eso no se recomienda salvo que eso sea nuestra intención.

Para poder cambiar permisos sobre un archivo, es necesario poseer el permiso de escritura sobre el mismo. El usuario *root* puede modificar los permisos de cualquier archivo ya que tiene acceso total sin restricciones a la administración del sistema.

### 9.2.5 Visualizar los permisos de archivos y carpetas

Con el comando `ls -l` podemos visualizar los permisos de los archivos o carpetas. Al ejecutar el comando aparecen todos los archivos, uno por línea. El bloque de 10 caracteres del principio simboliza el tipo de archivo y los permisos.

```

usuario@Ubuntu11:~$ ls -l
total 36
drwxr-xr-x 2 usuario usuario 4096 2011-08-29 18:24 Descargas
drwxr-xr-x 2 usuario usuario 4096 2011-08-21 18:44 Documentos
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Escritorio
-rw-r--r-- 1 usuario usuario 179 2011-08-19 11:51 examples.desktop
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Imágenes
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Música
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Plantillas
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Público
drwxr-xr-x 2 usuario usuario 4096 2011-08-19 11:53 Vídeos
usuario@Ubuntu11:~$
    
```

Figura 25.- Permisos de archivos y carpetas

El primer carácter indica de qué tipo de archivo se trata. Si es un guión `-` significa que se trata de un archivo normal, la letra `d` significa que se trata de una carpeta o directorio, la letra `l` significa que se trata de un enlace (link). Otros valores son `s`, `p`, `b` que se refieren a

sockets, tuberías (pipe) y dispositivos de bloque respectivamente.

Los 9 caracteres siguientes simbolizan los permisos del usuario propietario (3 caracteres), los permisos del grupo propietario (3 caracteres) y los permisos del resto de usuarios (3 caracteres). Vienen codificados con las letras **r**, **w** y **x** que se refieren a los permisos de lectura, escritura y ejecución. Si en lugar de aparecer dichas letras aparecen guiones significa que se carece de dicho permiso. Ejemplo, si los diez primeros caracteres son **-rw-r-----** significa que es un archivo normal, que el usuario propietario dispone de permisos de lectura y escritura, pero no de ejecución, que el grupo propietario dispone tan solo de permiso de lectura y el resto de usuarios no dispone de ningún permiso. Veámoslo en la siguiente imagen:

#### Tipo de archivo:

(-) para archivos normales

(d) para carpetas (directory)

(l) para enlaces (link)

(s)=socket, (p)=tubería (pipe), (b)=dispositivo de bloque.

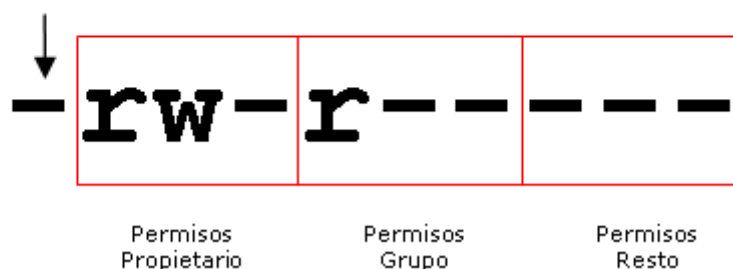


Figura 26.- Permisos de archivo

En el siguiente ejemplo vemos que **usuario** tiene permiso de lectura y escritura y que el resto solo tiene permiso de lectura tanto sobre todos los archivos.

```
-rw-r--r-- 1 usuario usuario 8656 2011-08-30 18:23 ejemplo_calc.ods
-rw-r--r-- 1 usuario usuario 8872 2011-08-30 18:22 ejemplo_pdf.pdf
-rw-r--r-- 1 usuario usuario 2 2011-08-30 18:07 ejemplo.txt
-rw-r--r-- 1 usuario usuario 9217 2011-08-30 18:22 ejemplo_writer.odt
```

Figura 27.- Ejemplo de permisos

## 9.3 Cambio de permisos. Comando chmod

El comando **chmod** cambia los permisos de un archivo o carpeta. Para cambiar los permisos de un archivo o una carpeta es necesario disponer del permiso de escritura sobre dicho archivo o carpeta.

#### Sintaxis

```
chmod [opciones] modo nombre_archivo_o_carpeta
```

#### Parámetros

##### modo

Indica los permisos que se conceden o se deniegan. Existen dos formas de establecer esta

máscara:

**Modo simbólico.**- Se emplean una operación de dos operandos y un operador. El primer operando indica a quién se definen los permisos. Después viene el operador y por último los permisos.

La sintaxis del modo simbólico es **[ugoa][+=[rwx]** donde:

- ✓ [ugoa] indica a quién se está definiendo los permisos. Pueden ponerse uno o varios de los siguientes: u usuario, g grupo, o otros y a todos.
- ✓ [+=[ indica la operación sobre los permisos. Solamente puede ponerse un operando y son + para añadir permisos, - para quitar permisos y = para establecer exactamente los permisos.
- ✓ [rwx] indica los permisos que se están definiendo. Pueden ponerse uno o varios de los siguientes: r lectura, w escritura y x ejecución.

**Modo octal.**- Los permisos se representan mediante un código numérico cuya transformación al binario representaría la activación o desactivación de los permisos. El código numérico está compuesto por tres cifras entre 0 y 7. La primera de ellas representaría los permisos del usuario propietario, la segunda los del grupo propietario y la tercera los del resto de usuarios.

En binario, las combinaciones representan el tipo de permisos. El bit más a la derecha (menos significativo) se refiere al permiso de ejecución (1=activar y 0=desactivar). El bit central se refiere al permiso de escritura y el bit más a la izquierda se refiere al permiso de lectura. La siguiente tabla muestra las 8 combinaciones posibles:

Código	Binario	Permisos efectivos
0	0 0 0	- - -
1	0 0 1	- - x
2	0 1 0	- w -
3	0 1 1	- w x
4	1 0 0	r - -
5	1 0 1	r - x
6	1 1 0	r w -
7	1 1 1	r w x

Si deseamos otorgar sólo permiso de lectura, el código a utilizar es el 4. Si deseamos otorgar sólo permiso de lectura y ejecución, el código es el 5. Si deseamos otorgar sólo permiso de lectura y escritura, el código es el 6. Si deseamos otorgar todos los permisos, el código es el 7. Si deseamos quitar todos los permisos, el código es el 0.

[nombre\\_archivo\\_o\\_carpeta](#)

Archivos o carpetas a los que se les cambian los permisos. Se pueden usar caracteres comodín para cambiar los permisos a varios archivos o carpetas.

*Opciones*



-R

--recursive

Cambia los permisos de carpetas recursivamente, es decir, cambiaría los permisos de la carpeta y de lo que contiene. Si contiene más carpetas también se cambiarían los permisos de sus archivos. Así sucesivamente hasta el final del árbol.

Ejemplos utilizando el modo simbólico:

```
// Dar permiso de escritura al usuario propietario sobre el archivo
'examen.txt'
# chmod u+w examen.txt

// Quitar permiso de escritura al resto de usuarios sobre el archivo
'examen.txt'
# chmod o-w examen.txt

// Dar permiso de ejecución al grupo propietario sobre el archivo
'/usr/bin/games/tetris'
# chmod g+x /usr/bin/games/tetris

// Dar permiso de lectura al grupo propietario sobre el archivo
'examen.txt'
# chmod g+r examen.txt

// Se pueden poner varios permisos juntos separados por comas
# chmod u+w,g-r,o-r examen.txt

// Se pueden poner varios usuarios juntos
# chmod ug+w examen.txt
```

Ejemplos en modo octal:

```
// Dar todos los permisos al usuario y ninguno ni al grupo ni al
resto
# chmod 700 examen.txt

// Dar al usuario y al grupo permisos de lectura y ejecución y
ninguno al resto
# chmod 550 examen.txt

// Dar todos los permisos al usuario y lectura y ejecución al grupo y
al resto
# chmod 755 /usr/bin/games/tetris

// Dar todos los permisos al usuario y de lectura al resto, sobre
todos los archivos
# chmod 744 *

// Cambiar permisos a todos los archivos incluyendo subcarpetas
# chmod -R 744 *
```



## 9.4 Permisos especiales: el bit Sticky, SUID, SGID

Hay una serie de permisos especiales sobre el sistema de archivos de Linux que pueden resultar útiles para determinadas tareas o para organizar directorios colaborativos entre diferentes usuarios.

### 9.4.1 El bit Sticky

El bit Sticky se utiliza para permitir que cualquiera pueda escribir y modificar sobre un archivo o directorio, pero que solo su propietario o `root` pueda eliminarlo. Un ejemplo de uso es el directorio `/tmp`, que debe tener permisos para ser utilizado por cualquier proceso, pero solo el dueño o `root` puede eliminar los archivos que crea. Este bit en modo octal sería el 1, equivalente al permiso de ejecución al resto de usuarios. En modo simbólico se emplea la letra `t` la cual aparece sustituyendo al permiso de ejecución para todos los usuarios, por ejemplo.

```
drwxr-xr-t 2 usuario usuario 4096 mar  6 09:32 directorio
```

Vemos en el ejemplo anterior que el permiso de ejecución para el resto de usuarios es `t` lo cual indica que el bit sticky está activado.

### 9.4.2 El bit SUID

El bit SUID activo en un archivo significa que el que lo ejecute va a tener los mismos permisos que el que creó el archivo. Esto puede llegar a ser muy útil en algunas situaciones pero hay que utilizarlo con cuidado, dado que puede generar grandes problemas de seguridad. Este bit en modo octal sería el 4, equivalente al permiso de lectura. En modo simbólico es la letra `s` o `S`, dependiendo de si está activado el permiso de ejecución para el propietario. Por ejemplo

```
-rwSrwxr-x 1 usuario usuario 133 ago 27 08:25 script1.sh
-rwsrw-r-- 1 usuario usuario 213 ago 27 08:25 script2.sh
```

Vemos que el archivo `script1.sh` tiene la letra `S` en lugar de `x` en el permiso de ejecución del usuario propietario. Eso significa que tiene asignado el bit SUID pero no se le ha asignado el permiso de ejecución, por lo que el bit SUID no tiene efecto. Sin embargo, el archivo `script2.sh` tiene la letra `s` en lugar de la `x` en el permiso de ejecución del usuario propietario, lo que significa que tiene asignado el bit SUID y también el permiso de ejecución.

### 9.4.3 El bit SGID

El SGID es lo mismo que en el SUID, pero a nivel de grupo. Es decir, todo archivo que tenga activo el SGID, al ser ejecutado, tendrá los privilegios del grupo al que pertenece.

Esto es muy usado cuando queremos configurar un directorio colaborativo: si aplicamos este bit al directorio, cualquier archivo creado en dicho directorio, tendrá asignado el grupo al que pertenece el directorio. Este bit en modo octal sería el 2, equivalente al permiso de escritura. En modo simbólico también se representa mediante la letra `s` o `S`, dependiendo de si está activado el permiso de ejecución para el grupo.

### 9.4.4 Asignar bit Sticky, SUID, SGID

Teniendo en cuenta que los bits Sticky, SUID y SGID son permisos especiales, para asignarlos disponemos del comando `chmod`. Si recordamos, el comando recibe como argumentos una máscara de permisos en formato octal formada por tres números: el primero para el propietario, el segundo para el grupo y el tercero para el resto de usuarios.

Hay un cuarto número que se emplearía para estos permisos especiales y que se pone en primer lugar. Generalmente, estos permisos no se modifican por lo que este primer número se omite y el comando `chmod` asume que es 0, tomando los tres números siguientes para establecer los permisos del archivo o directorio.

Por tanto, si queremos establecer estos permisos tendremos que añadir este primer número empleando la suma de los bits correspondientes: 1 para sticky, 2 para SGID y 4 para SUID. Por ejemplo para asignar el bit sticky a un directorio habría que escribir el siguiente comando:

```
chmod 1755 ~/directorio
```

En modo simbólico tendríamos que añadir o asignar el permiso con la letra `t` al resto de usuarios. Por ejemplo

```
chmod o+t ~/directorio
```

Para el bit SUID tenemos el bit 4 o la letra `s` para el propietario del archivo. Por ejemplo

```
chmod 4755 ~/directorio
```

En modo simbólico tendríamos que añadir o asignar el permiso con la letra `s` al propietario del archivo o directorio. Por ejemplo

```
chmod u+s ~/directorio
```

Si el usuario tiene el permiso de ejecución verá una `s` al listar el archivo. Por el contrario si no lo tiene verá la letra `S`.

Para el bit SGID debemos de utilizar el bit 2 o la letra `s` para el grupo. Por ejemplo

```
chmod 2755 ~/directorio
```

## 9.5 Gestión de permisos en formato gráfico

Existe la posibilidad de cambiar los permisos utilizando el explorador de archivos. Para ello tan solo hay que seleccionar los archivos o carpetas y haciendo clic sobre la selección con el botón derecho del ratón y eligiendo la opción *Propiedades*, nos aparecerá la ventana de propiedades. Haciendo clic en la pestaña *Permisos* podremos establecer los permisos de una forma sencilla.

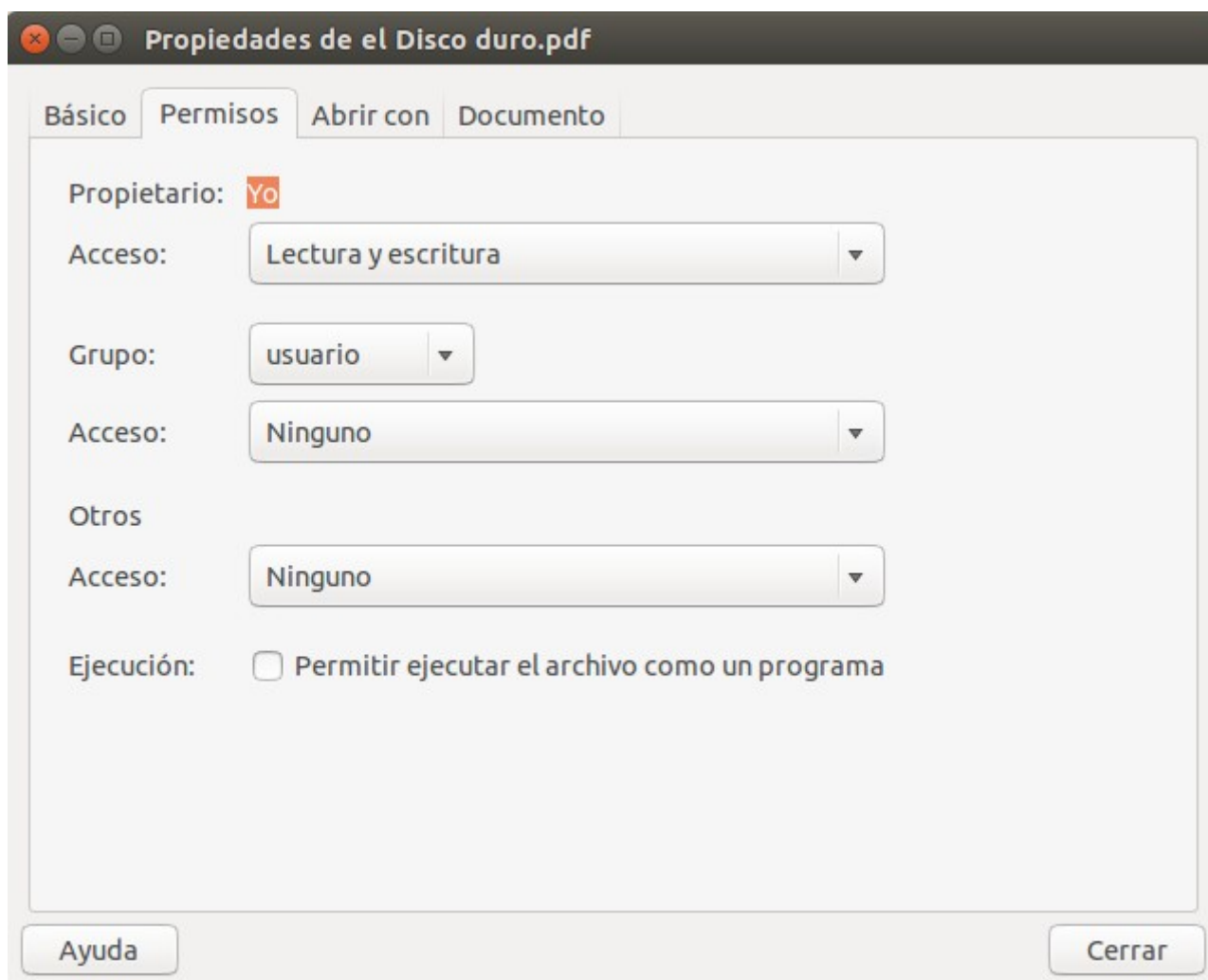


Figura 28.- Permisos en formato gráfico

Como se ve en la imagen anterior disponemos de tres listas para especificar los permisos del archivo al propietario, grupo y el resto de usuarios. Solamente podremos asignar permisos de lectura y escritura. En el caso de los archivos podemos conceder permiso de ejecución activando la casilla *Permitir ejecutar el archivo como un programa*.

Para una carpeta los permisos son diferentes. En la siguiente imagen vemos que también disponemos de tres listas para asignar permisos al propietario de la carpeta, el grupo propietario y el resto de usuarios. Sin embargo, al ser una carpeta el permiso de lectura corresponde a *Solo listar archivos*, el de lectura y ejecución a *Acceder y listar archivos* y finalmente el de lectura, escritura y ejecución a *Crear y eliminar archivos*.

Si además de cambiar los permisos hacemos clic en el botón *Cambiar permisos a los archivos contenidos* hace un cambio de permisos recursivo, asignando la máscara de permisos definida en la carpeta a los archivos y carpetas que contiene hasta el final de la jerarquía del almacenamiento.



Figura 29.- Permisos de carpeta

## 9.6 Cambio de propietario. Comando chown

Cada archivo y carpeta en Linux tiene un usuario propietario que es el usuario que crea el archivo o carpeta. También tiene un grupo propietario que es el grupo por defecto del usuario que crea el archivo o carpeta.

El usuario propietario y el grupo propietario pueden cambiarse, pero solamente lo puede hacer el usuario `root`. Para ello se emplea el comando `chown`.

### Sintaxis

```
chown [opciones] [login][:grupo] archivo
```

### Parámetros

`login`

Login del nuevo propietario

`grupo`

Nuevo grupo propietario.

`archivo`

Archivos a los que se les cambia el propietario

### Opciones

-R

Cambia el propietario de forma recursiva. Si es un directorio cambiará el propietario a su contenido.

Si se indica únicamente el `login` cambiará el usuario propietario, pero no el grupo propietario. Si se indica el `login` seguido de los dos puntos, se cambiará el usuario propietario y también el grupo propietario al grupo por defecto del nuevo usuario propietario. Si se omite el login y se indican los dos puntos con el nombre del grupo solamente se cambiará el grupo propietario.

## 9.7 Máscara de permisos. Comando `umask`

Con el comando `chmod` podemos cambiar los permisos del archivo, pero ... ¿cuáles son los permisos del archivo cuando se crea? Durante una sesión de trabajo el usuario puede crear una gran cantidad de archivos al emplear un editor de textos, navegar por Internet, etc. Las aplicaciones que utilice crean archivos que pertenecen al usuario y estos archivos tienen que tener un conjunto de permisos iniciales. Para establecer cuales serán los permisos de un archivo nuevo creado se emplea el comando `umask`.

En realidad, la máscara no establece los permisos por defecto, sino su complemento. Por ejemplo si tenemos una máscara por defecto de 0002, entonces los archivos se crean con permisos 0664 (resultado de restar 0666 a 0002) y los directorios con permisos 775 (resultado de restar 0777 a 0002). El primer número, al igual que con el comando `chmod`, corresponde a los bits especiales.

### Sintaxis

```
umask [mascara]
```

### Argumentos

#### `máscara`

Máscara en modo octal. En realidad es el complemento del modo deseado para los archivos. Es decir, la máscara efectiva de permisos para un archivo nuevo sería el resultado de restar 0777 a la máscara indicada con `umask`. Por ejemplo, si la máscara es 022,  $777-022 = 755$ , que sería la máscara de permisos para los archivos nuevos. Estos tendrán todos los permisos para el propietario, y los de lectura y ejecución para el grupo y resto de usuarios.

Resulta conveniente que el usuario `root` tenga una máscara que sea 077, para que nadie pueda acceder a sus archivos.

## 9.8 ACLs

En los epígrafes anteriores hemos visto los permisos de archivo y directorio, los cuales

crean tres grandes grupos de usuarios a los que se les puede conceder o denegar permisos: propietario, grupo propietario y el resto de usuarios.

Sin embargo, hay ocasiones en los que se necesita una mejor granularidad a la hora de asignar permisos a los archivos o directorios, como por ejemplo asignar un conjunto de permisos a un usuario o grupo concreto sin necesidad de asignar esos mismos permisos a todos los usuarios. Para estas situaciones podemos emplear las ACLs.

Una ACL (*Access Control List*) es un conjunto de usuarios, grupos y permisos asignados a esos usuarios y grupos sobre un archivo o directorio. Cada archivo o directorio tiene su propia ACL independiente del resto. En los siguientes apartados vamos a ver como utilizar esta característica que no sustituye al tradicional sistema de permisos de archivo y directorio, sino que lo complementa.

### 9.8.1 Habilitar ACL en un sistema de archivos

Para poder emplear ACLs en los sistemas de archivos hay que habilitarlas. Esto se consigue con la opción de montaje `acl`. Es muy posible que si el sistema de archivos es ext ya esté habilitado por defecto. Para comprobarlo podemos ejecutar el siguiente comando

```
usuario@U1804:~$ sudo tune2fs -l /dev/sda2 | grep "Default mount options:"
Default mount options:      user_xattr acl
```

Como podemos ver en el resultado de la ejecución del comando, la segunda partición del primer disco duro tiene como opción por defecto de montaje `acl`. Esta partición contiene las carpetas personales de los usuarios. Por tanto ya está habilitada por defecto, salvo que en las opciones de montaje de esta partición en `/etc/fstab` hayamos puesto `noacl`.

En el caso de que no estuviera habilitada por defecto tenemos dos formas para hacerlo. La primera es indicar la opción de montaje `acl` en el archivo `/etc/fstab`. La segunda es mediante la ejecución del siguiente comando.

```
usuario@U1804:~$ sudo tune2fs -o acl /dev/sda2
```

También, si queremos activar opciones de montaje por defecto disponemos de la opción `-O acl` del comando `mke2fs` que crea el sistema de archivos ext2,3 o 4.

En los siguientes apartados veremos como asignar, quitar y ver los permisos mediante el uso de ACLs. Para ello supondremos que tenemos el archivo `/home/usuario/archivo.txt` y el directorio `/home/usuario/directorio` para los que vamos a gestionar una ACL.

### 9.8.2 Listar permisos. Comando `getfacl`

El comando `getfacl` permite mostrar los permisos concedidos a un archivo o directorio.

#### *Sintaxis*

```
getfacl [-aceEsRLPtpndvh] archivo ...
```

### Argumentos

`archivo`

Archivo o directorio a visualizar los permisos de la ACL.

### Opciones

`-a, --access`

Visualiza la ACL del archivo.

`-d, --default`

Visualiza la ACL por defecto del directorio.

`-c, --omit-header`

No visualiza la cabecera (las tres primeras filas de la salida de cada archivo).

`-e, --all-effective`

Visualiza todos los comentarios de permisos efectivos incluso si coinciden con los de la correspondiente entrada de la ACL.

`-E, --no-effective`

No visualiza los comentarios con los permisos efectivos.

`-s, --skip-base`

Se salta los archivos que solamente tienen las entradas ACL base.

`-R, --recursive`

Lista las ACLs de todos los ficheros y directorios recursivamente.

`-L, --logical`

Sigue los enlaces simbólicos a los directorios y salta los enlaces simbólicos encontrados en subdirectorios. Solo es efectivo combinado con -R.

`-P, --physical`

No sigue los enlaces simbólicos de los directorios. Solo es efectivo combinado con -R.

`-t, --tabular`

Usa una salida en formato de tabla. Los permisos no efectivos debido a una entrada de máscara se visualizan con letra mayúscula.

`-p, --absolute-names`

No eliminar la barra / inicial. El comportamiento por defecto es quitar la barra / inicial del nombre de archivo.

`-n, --numeric`

Lista los IDs de usuario y grupo en lugar de sus nombres.

`-v, --version`

Visualiza la versión de getfacl.

`-h, --help`

Visualiza la ayuda del comando.

`--`

Finaliza las opciones en línea del comando. El resto de parámetros se interpretan como nombres de archivo, incluso si alguno comienza con guión medio -.

`-`

Si el nombre de argumento es un guión medio, getfacl lee la lista de archivos de la entrada estándar.

Para cada archivo visualiza el nombre, propietario, grupo y ACL. Si el es un directorio mostrará también los permisos por defecto. La salida del comando puede mostrar lo siguiente:

```
1: # file: algún_directorio/
2: # owner: usuario_propietario
3: # group: grupo_propietario
4: # flags: -s-
5: user::rwx
6: user:pepe:rwx           #effective:r-x
7: group::rwx             #effective:r-x
8: group:amigos:r-x
9: mask::r-x
10: other::r-x
11: default:user::rwx
12: default:user:pepe:rwx  #effective:r-x
13: default:group::r-x
14: default:mask::r-x
15: default:other:---
```

En la salida del comando vemos lo siguiente:

- ✓ Las líneas 1 a 3 indican el directorio, propietario, grupo y grupo propietario.
- ✓ La línea 4 hace referencia a los bits especiales setuid (s), setgid (s) y sticky (t). En el caso de que estén activados aparecerá la correspondiente letra. En caso contrario aparece un guión medio -.
- ✓ Las líneas 5, 7 y 10 hacen referencia a los permisos del directorio para el usuario propietario, el grupo propietario y el resto de usuarios, es decir, los permisos tradicionales de los sistemas Unix/Linux. Estas entradas se conocen como entradas de ACL base.



- ✓ Las líneas 6 y 8 son entradas para un usuario y un grupo con nombre. Son permisos concedidos a un usuario o grupo específico.
- ✓ La línea 9 se refiere a los permisos de la máscara que indican los derechos máximos que se pueden conceder a usuarios con nombre, grupos con nombre y grupo propietario. El propietario y el resto de usuarios no se ven afectados por la máscara, solamente los usuarios y grupos con nombre y el grupo propietario.
- ✓ Las líneas 11 a 15 visualizan la ACL por defecto asociada con este directorio. Los directorios tienen permisos por defecto, mientras que los archivos no los tienen. Estos permisos por defecto se aplican a los archivos creados dentro del directorio.

El comportamiento por defecto de `getfacl` es visualizar la ACL y la ACL por defecto, e incluir los derechos efectivos comentados en aquellas líneas que difieren de los derechos de la entrada.

Por ejemplo, si ejecutamos este comando veremos lo siguiente

```
usuario@U1804:~$ getfacl archivo.txt
# file: archivo.txt
# owner: usuario
# group: usuario
user::rw-
group::r--
other::r--
```

Vemos que aparece el nombre y los propietarios del archivo. Inicialmente, `archivo.txt` solamente tiene los permisos tradicionales de Unix/Linux. Serían los que aparecen al listar el archivo en formato largo.

```
usuario@U1804:~$ ls -l archivo.txt
-rw-r--r-- 1 usuario usuario 2 dic 31 17:53 archivo.txt
```

Si ejecutamos el mismo comando sobre el directorio `directorio`.

```
usuario@U1804:~$ getfacl directorio
# file: directorio
# owner: usuario
# group: usuario
user::rwx
group::r-x
other::r-x
```

Seguiremos usando este comando conforme vayamos asignando entradas a la ACL del archivo y/o directorio.

### 9.8.3 Asignar permisos en una ACL. Comando `setfacl`

Para asignar permisos a usuarios y grupos en una ACL empleamos el comando `setfacl`. Este comando se basa en el uso de entradas de ACL. Una entrada de ACL es un conjunto de permisos asignados a un usuario o grupo concreto, además de a las tres

entidades básicas susceptibles de tener permisos: usuario propietario, grupo propietario y resto de usuarios. La ACL de un archivo o directorio se compone de un conjunto de entradas de ACL las cuales establecen los permisos para un conjunto de usuarios y/o grupos. Veamos a continuación como se pueden gestionar las entradas de ACL en un archivo o directorio.

### Sintaxis

```
setfacl [-bkndRLPvh] [{-m|-x|--set} entrada ACL] [{-M|-X|--set-file}  
archivo ACL] archivo ...  
setfacl --restore=file
```

### Argumentos

`-m entrada ACL`

Modifica la entrada ACL

`-x entrada ACL`

Borra la entrada ACL

`--set entrada ACL`

Reemplaza las entradas ACLs actuales con la entrada ACL especificada.

### Opciones

`-b, --remove-all`

Quita todas las entradas ACL, excepto las entradas de grupo y resto de usuarios.

`-k, --remove-default`

Quita las entradas ACL por defecto.

`-n, --no-mask`

No recalcula la máscara de permisos efectivos. Por defecto, la entrada de máscara se recalcula a no ser que se especifique una máscara explícitamente. La entrada de máscara es establecida con la unión de todos los permisos del grupo propietario y todas las entradas de usuario y grupo con nombre. Estas son exactamente las entradas afectadas por la entrada máscara.

`--mask`

Recalcula la máscara de permisos efectivos incluso si una entrada máscara fue establecida explícitamente.

`-d, --default`

Todas las operaciones se aplican a una ACL por defecto. Las entradas ACL regulares son promovidas a entradas ACL por defecto.

`--restore=archivo`

Restaura una copia de permisos creada con getfacl -R o similar. Todos los permisos de un

subárbol completo de directorio se restaura usando este mecanismo. Si el archivo contiene comentarios del propietario o grupo entonces se intenta restaurar el propietario y grupo. Si el archivo contiene comentarios de los flags de bits especiales los establece, en otro caso los borra. Esta opción solo puede combinarse con `-test`.

`--test`

Modo de prueba. En lugar de cambiar las ACLs del archivo, muestra el resultado.

`-R, --recursive`

Lista las ACLs de todos los ficheros y directorios recursivamente

`-L, --logical`

Sigue los enlaces simbólicos a los directorios y salta los enlaces simbólicos encontrados en subdirectorios. Solo es efectivo combinado con `-R`

`-P, --physical`

No sigue los enlaces simbólicos de los directorios. Solo es efectivo combinado con `-R`.

`-v, --version`

Visualiza la versión de getfacl

`-h, --help`

Visualiza la ayuda del comando.

`--`

Finaliza las opciones en línea del comando. El resto de parámetros se interpretan como nombres de archivo, incluso si alguno comienza con guión medio `-`.

`-`

Si el nombre de argumento es un guión medio, getfacl lee la lista de archivos de la entrada estándar.

Los formatos aceptados de una entrada ACL son los siguientes:

`[d[efault]:][u[ser]:]uid[:permisos]`

Permisos de un usuario nombrado o del propietario si se omite el identificador de usuario está vacío.

`[d[efault]:]g[roup]:gid[:permisos]`

Permisos de un grupo nombrado o del grupo propietario si se omite el identificador de grupo.

`[d[efault]:]m[ask][:][:permisos]`

Máscara efectiva de permisos

`[d[efault]:]o[ther][:][:permisos]`

Permisos del resto de usuarios

Las entradas ACL que incluyen permisos se emplean en operaciones para modificar o establecer permisos (opciones `-m`, `-M`, `--set` y `--set-file`). Entradas sin el campo permisos se emplean para borrado de entradas (opciones `-x` y `-X`). Pueden especificarse varias entradas ACL separadas por coma.

En lugar de los identificadores de usuario y grupo se pueden emplear el nombre.

El campo `permisos` es una combinación de caracteres que indican los permisos: lectura (`r`), escritura (`w`), ejecución (`x`). El permiso de ejecución solo si es un directorio o ya tiene un permiso de ejecución para algún usuario. Alternativamente, el campo permisos puede estar en notación octal (0-7).

Inicialmente, los archivos y directorios solo contienen las tres entradas ACL base para el propietario, grupo y resto de usuarios. Hay reglas que se tienen que cumplir para que una ACL sea válida. Son estas:

- ✓ Las tres entradas base no pueden borrarse.
- ✓ Si se añaden entradas con nombre (usuario o grupo) debe haber una máscara de permisos efectivos. En el caso de que no la haya `setfacl` la crea calculándola con los mismos permisos de la entrada con nombre añadida.
- ✓ Si una ACL contiene entradas por defecto, las tres entradas ACL base (propietario, grupo y resto de usuarios) por defecto deben existir. Si no existen, se crean unas copiando las ACL base.
- ✓ Si una entrada ACL por defecto contiene nombre de usuario o grupo, también tiene que contener una máscara de permisos efectivos por defecto.

Veamos algunos ejemplos. Inmediatamente, después de cada ejemplo se mostrarán las ACLs con el comando `getfacl` visto en el epígrafe anterior. El siguiente comando asigna permiso de lectura y escritura al usuario `usuario2` sobre el archivo `archivo.txt`.

```
usuario@U1804:~$ setfacl -m u:usuario2:rw archivo.txt
usuario@U1804:~$ getfacl archivo.txt
# file: archivo.txt
# owner: usuario
# group: usuario
user::rw-
user:usuario2:rw-
group::r--
mask::rw-
other::r--
```

Vemos que hay una entrada para el `usuario2` con permisos de lectura y escritura. También hay una entrada máscara (`mask`) con los mismos permisos anteriores. Esta entrada no la hemos introducido nosotros, sino que `setfacl` la ha calculado empleando para ello las entradas que a su vez son afectadas por la máscara: usuarios con nombre, grupos con nombre y grupo propietario.

En el siguiente ejemplo creamos ACL por defecto. En este caso solamente puede ser para un directorio.

```
usuario@U1804:~$ setfacl -m d:u::rwx directorio
usuario@U1804:~$ getfacl directorio
# file: directorio
# owner: usuario
# group: usuario
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:other::r-x
```

Vemos que se ha asignado una ACL por defecto al directorio en el que el usuario propietario tiene todos los permisos, mientras que el grupo propietario y el resto de usuarios han copiado los permisos Unix/Linux.

En el siguiente ejemplo damos permisos por defecto al `usuario2` sobre el directorio `directorio`.

```
usuario@U1804:~$ setfacl -m d:u:usuario2:rwx directorio
usuario@U1804:~$ getfacl directorio
# file: directorio
# owner: usuario
# group: usuario
user::rwx
group::r-x
other::r-x
default:user::rwx
default:user:usuario2:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

Vemos que al añadir una entrada ACL por defecto para el `usuario2` solamente existía previamente una entrada ACL por defecto para el propietario, para el grupo propietario y el resto de usuarios, las cuales se mantienen.

Al añadir la primera entrada ACL por defecto con nombre para el `usuario2` también se ha creado una entrada ACL de máscara por defecto copiando los permisos de la entrada ACL por defecto con nombre creada para ese usuario.

El directorio contiene ahora entradas de ACL por defecto. Por tanto, cuando el propietario del directorio crea un archivo dentro tendrá una ACL. Supongamos que dentro del directorio creamos el archivo `archivo1.txt`. Este tiene las siguientes ACLs.

```
usuario@U1804:~$ getfacl ./directorio/archivo1.txt
# file: directorio/archivo1.txt
# owner: usuario
```

```
# group: usuario
user::rw-
user:usuario2:rw-      #effective:rw-
group::r-x             #effective:r--
mask::rw-
other::r--
```

Vemos que se han creado entradas para el propietario, grupo propietario, resto de usuarios y el usuario `usuario2`. Si nos fijamos en las ACLs por defecto del directorio padre dispone del permiso de ejecución que en los archivos no se hereda.

Si creamos un directorio dentro de este veremos los siguientes permisos.

```
usuario@U1804:~$ getfacl ./directorio/d1
# file: directorio/d1
# owner: usuario
# group: usuario
user::rwx
user:usuario2:rwx
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:user:usuario2:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

Al ser un directorio el permiso de ejecución se hereda de las ACL por defecto.

Como se mencionó anteriormente, la máscara establece los máximos permisos que se pueden aplicar a usuarios con nombre, grupos con nombre y grupo propietario. Aunque establezcamos explícitamente unos permisos para algún usuario o grupo con nombre, la máscara limitará estos permisos. A la derecha de cada entrada ACL puede aparecer el comentario `#effective:permisos` para indicar cuáles son los permisos efectivos que se están aplicando en el caso de que los concedidos no coincidan con la máscara.

La máscara se recalcula cada vez que se modifican las entradas ACL que afectan a usuarios con nombre, grupos con nombre o grupo propietario. Entonces se unen todos los permisos de estas entradas y se forma la máscara. Sin embargo, podemos modificarla explícitamente. En el siguiente ejemplo modificamos la máscara del archivo `archivo.txt`. Se muestran la ACL antes y después de modificar la máscara.

```
usuario@U1804:~$ getfacl archivo.txt
# file: archivo.txt
# owner: usuario
# group: usuario
user::rw-
user:usuario2:rw-
group::r--
```

```
mask::rw-
other::r--

usuario@U1804:~$ setfacl -m m:r archivo.txt
usuario@U1804:~$ getfacl archivo.txt
# file: archivo.txt
# owner: usuario
# group: usuario
user::rw-
user:usuario2:rw-      #effective:r--
group::r--
mask::r--
other::r--
```

Vemos como hemos cambiado la máscara y hemos dejado solamente el permiso de lectura. Al variar la máscara con los permisos del `usuario2`, aparece el comentario `#effective:r--` a su derecha.

Sin embargo, si creamos una nueva entrada ACL a alguna de las entidades afectadas por la máscara, esta se recalcula de nuevo. Por ejemplo, vamos a asignar permisos de lectura, escritura y ejecución al `usuario3` sobre el archivo `archivo.txt`. Al final del ejemplo anterior tenemos la ACL del `archivo.txt` y en el siguiente vemos lo que ocurre con la máscara.

```
usuario@U1804:~$ setfacl -m u:usuario3:rwX archivo.txt
usuario@U1804:~$ getfacl archivo.txt
# file: archivo.txt
# owner: usuario
# group: usuario
user::rw-
user:usuario2:rw-
user:usuario3:rwX
group::r--
mask::rwX
other::r--
```

Al asignar una entrada ACL a un usuario con nombre, la máscara se ha recalculado uniendo todos los permisos asignados a las tres entidades afectadas por la máscara. Ahora la máscara es `rwX`.

## 10 Bibliografía

---

SCHRODER, C. *Curso de Linux – 1ª Edición* 2005. Anaya Multimedia, ISBN 978-844-151-857-5

MAN, *Páginas de manual de Linux*

UBUNTU 20.04, *Guía del escritorio de Ubuntu*

DOCILE,E. *How to manage ACL on Linux* [accedido diciembre 2018]. Disponible en <<https://linuxconfig.org/how-to-manage-acls-on-linux>>