

**MP\_0483. Sistemas informáticos**

**UF3. Gestión de la información**

**3.2. Sistemas de archivos**

# Índice

---

☰	Objetivos	3
☰	Sistemas de archivos	4
☰	Tipos de sistemas de archivos	8
☰	¿Cuál es mi sistema?	14
☰	Creación de particiones	18
☰	Borrado de particiones	19
☰	Redimensionado de particiones	20
☰	Fusión de particiones	21
☰	Copia de particiones	22
☰	Recuperación de particiones	23
☰	Conversión de particiones	24
☰	Desfragmentación	25
☰	Ejercicio/problema práctico	30
☰	Del sistema de ficheros a las estructuras de directorios	33
☰	Montar volúmenes en carpetas	35
☰	Volúmenes en entorno Windows	36
☰	Volúmenes en entorno Unix/Linux	37
☰	Directorios y subdirectorios	39
☰	Ejercicio: directorios en Linux	41
☰	Resumen	43

# Objetivos

---

En esta unidad perseguimos los siguientes objetivos:

1

Comprender qué es un sistema de archivos.

2

Conocer los diferentes tipos de sistemas de archivos que existen, tanto en entornos libres como propietarios.

3

Conocer cómo se almacenan los archivos en directorios y carpetas dentro de un sistema de archivos.



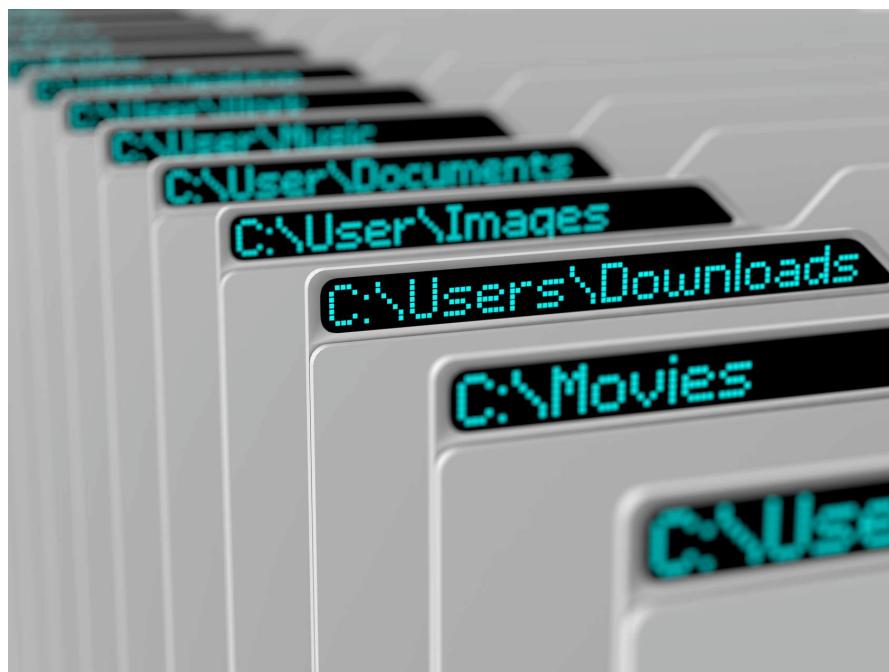
¡Ánimo y adelante!

# Sistemas de archivos

Cuando hablamos de sistema de archivos nos referimos al modo de organizar la información dentro de un dispositivo de almacenamiento.

Una de las funciones del sistema operativo es la de independizar al usuario de los detalles profundos del funcionamiento del HW del ordenador. Para lograrlo, una de las tareas fundamentales es el **establecimiento de un sistema de acceso a los archivos de programa y datos del disco**, la carga en memoria para su ejecución o su reemplazo cuando proceda, etc.

De este modo, sobre los dispositivos de almacenamiento **montaremos una organización de la información**, que es a lo que normalmente nos referimos como “**sistema de archivos**” (ficheros), y que se montará **sobre cada una de las particiones del disco** que hemos visto antes.



El usuario (o la aplicación de usuario) puede hacer una petición de lectura de un fichero, por ejemplo, trabajar con él y, cuando termine, ordenar que sea almacenado. Las rutinas del sistema operativo serán las encargadas de acceder al disco (o al software que lo controla) y provocar la lectura o escritura física del fichero almacenado en un sector/pista/etc.

## Ficheros y archivos especiales

Algunos sistemas operativos, como Unix por ejemplo, utilizan ciertos “archivos especiales” con el objeto de que los dispositivos de E/S se vean y sean tratados como ficheros (archivos). De esta forma pueden leer/escribir sobre ellos usando las mismas llamadas a sistema que para cualquier archivo normal.

Hay dos tipos de archivos especiales: de carácter y de bloque.

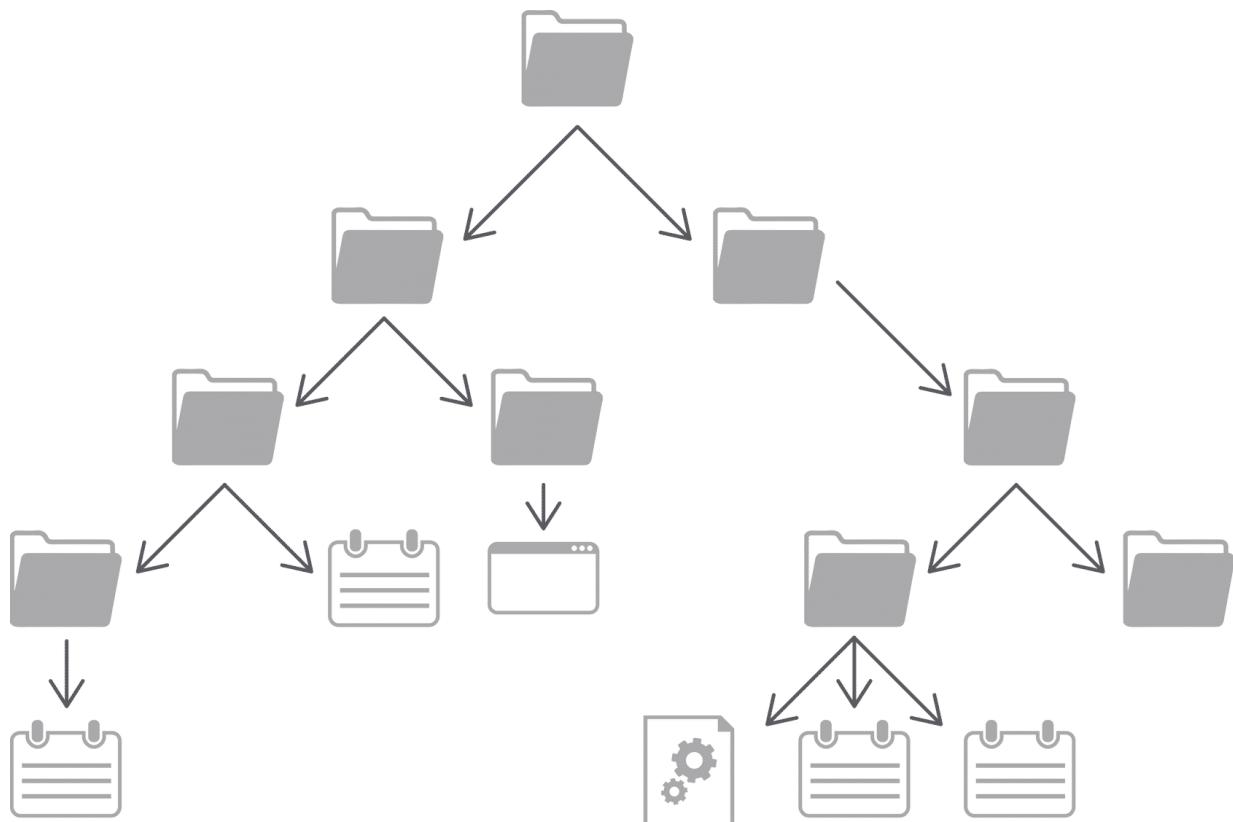
- **De bloque:** para dispositivos en los que se accede a un bloque de datos en su contenido (por ejemplo, un disco duro).
- **De carácter:** dispositivos a los que enviamos, o de los que recibimos, un flujo de caracteres (por ejemplo, una impresora).

También podemos considerar archivos especiales a las **tuberías (pipes)** que se utilizan para la comunicación entre procesos activos en el sistema. Cuando un proceso quiere enviarle información a otro escribe en la tubería (=canal=pipe) como si fuese un archivo de salida, y el otro proceso leerá en ella como si fuese un archivo de entrada.

El hecho de disponer de estos archivos especiales no influye en la organización de los sistemas de archivos en nuestro disco, pero es interesante conocer su existencia.

## Directarios

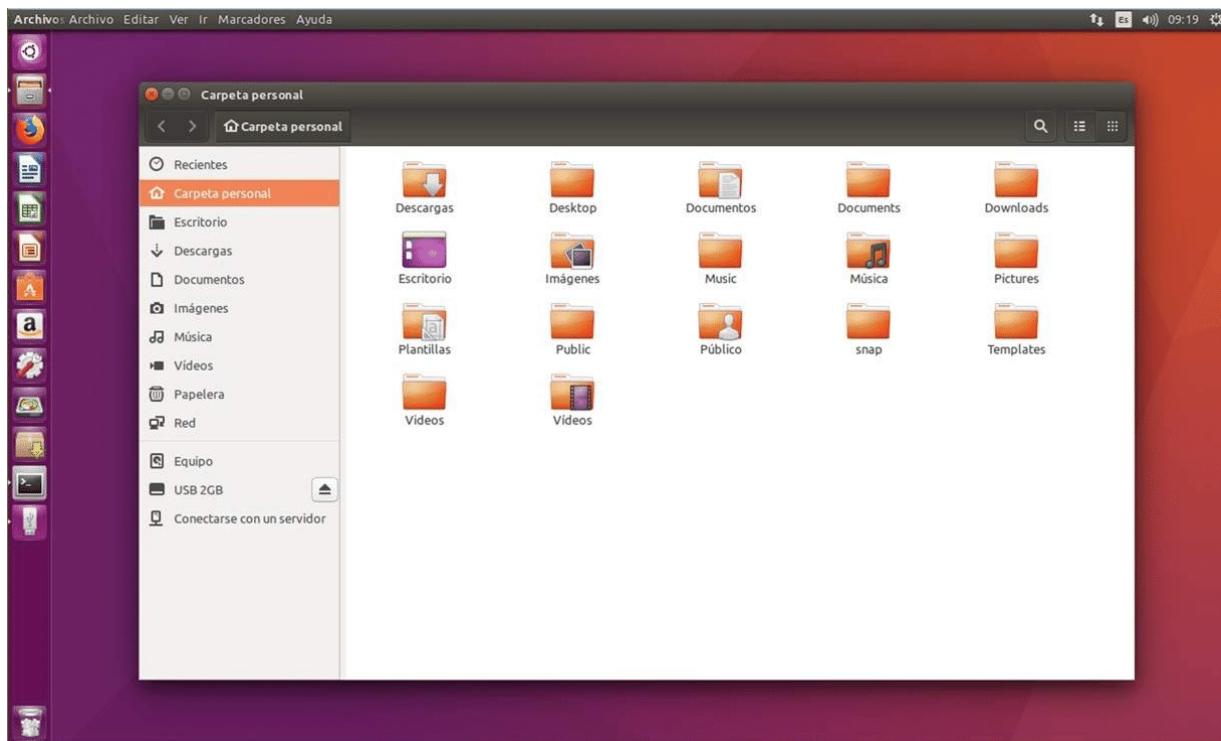
Dentro de cada sistema de archivos el sistema operativo organiza el almacenamiento de los ficheros y los agrupa en directorios y subdirectorios dentro de una estructura jerárquica, es decir, dentro de una estructura en árbol.



Estructura en árbol de un sistema de archivo.

---

Algunos de estos directorios son creados por el propio S.O. durante su instalación y otros se crean durante la instalación de las aplicaciones. Pero, además, el propio usuario creará los directorios que necesite para el almacenamiento de sus archivos y programas.



A nivel profundo, todo lo relativo al acceso, lectura, creación y eliminación de archivos y directorios se hace a través de las "**llamadas al sistema**" del propio S.O. Pero el usuario no lo ve, ya que accede directamente a las aplicaciones, o bien mediante una interfaz de comandos o algún entorno gráfico (GUI) con aplicaciones de gestión de archivos.

# Tipos de sistemas de archivos

---

Para poder utilizar los dispositivos de almacenamiento es necesario que demos un formato (estructura) a las particiones del disco.

Una vez que hemos elegido el registro de arranque (MBR o GPT) y han sido establecidas las particiones del disco, daremos estructura a estas particiones. Esta estructura será el **sistema de archivos** elegido para la partición.

Los sistemas de archivos determinan cómo se accede a los ficheros dentro de las particiones. Además del acceso (lectura/escritura) pueden incorporar más funcionalidades, como la posibilidad de recuperar datos, indexación para búsquedas rápidas, menor fragmentación de los bloques de archivo, etc.

El objetivo de cualquier sistema de archivos será:

- Optimizar el espacio del disco (gestionar el espacio disponible y el asignado).
- Mantener la integridad de los directorios y archivos.
- Facilitar el acceso a la información de forma rápida.

Es importante señalar que el sistema de archivos es algo particular de cada partición, de forma que en un mismo disco podemos tener particiones formateadas con un sistema (p. ej. NTFS) y otras con otro (p. ej. FAT32).

#### Partición 3 - extendida



Ejemplo de estructura en disco

Los sistemas de archivos más comunes son: sistemas FAT, sistemas NTFS, y sistemas UFS y Linux. Modernamente, aunque algunos sistemas de archivos nacieron como propios de un determinado sistema operativo, los principales pueden ser gestionados por la mayoría de los sistemas operativos modernos y así, por ejemplo, tanto Windows como Linux pueden manejar particiones con sistemas FAT o NTFS.

### Algunos ejemplos de sistemas de ficheros:

- FAT/FAT32 (MS-DOS, Windows).
- NTFS (Windows).
- UFS/FFS, ZFS (UNIX).
- ext2, ext3, ext4, ReiserFS (Linux).
- HFS, HFS+ (Mac OS).
- ISO 9660, CDFS (CD-ROM).
- NFS, SMB, Coda, AFS (red).



Puedes ver una lista más extensa en:

[https://es.wikipedia.org/wiki/Anexo:Comparaci%C3%B3n\\_de\\_sistemas\\_de\\_archivos](https://es.wikipedia.org/wiki/Anexo:Comparaci%C3%B3n_de_sistemas_de_archivos)

### Sistemas FAT

FAT es un acrónimo de “File Allocation Table” (tabla de asignación de ficheros) y fue el sistema de archivos originario del sistema operativo MS-DOS, que se ha mantenido en muchos de los sistemas Windows posteriores .

El formato FAT se utilizó sobre todo para los discos flexibles (disquetes) y se sigue usando para dispositivos extraíbles (como por ejemplo las tarjetas de memoria), ya que en la práctica es admitido por la mayoría de los S.O. de ordenadores personales.

---

En los sistemas FAT cada entrada de la tabla corresponde a un **clúster** (conjunto de sectores) e indica si este se encuentra en blanco o es defectuoso, cuál es el siguiente clúster donde continúa el archivo o si es el último.

Los sistemas de archivos FAT tienen algunas desventajas: **suelen dejar fragmentos residuales** fruto de los borrados de archivos. Esto con el tiempo va haciendo más lento el dispositivo, por lo que se hace necesaria la **desfragmentación del disco**.

Por otro lado, **no es un sistema con redundancia a fallos y soporta nombres de archivos cortos** para el uso actual. En principio eran solamente ocho caracteres para el nombre y tres para la extensión, pero luego se amplió a 255 caracteres.

Existen muchos sistemas FAT: FAT12, FAT16, FAT32, VFAT, FASTFAT. Normalmente cuando se habla solamente de FAT nos referimos a FAT16, pero el más extendido ha sido FAT32. Las diferencias entre ellos son varias, pero empiezan por el tamaño del direccionamiento en la tabla de asignación de archivos (por ejemplo 16 bits en FAT16, 32 bits en FAT32, etc.), lo cual proporciona una capacidad mayor o menor de clústeres a acceder, y por tanto una capacidad mayor o menor en el tamaño de las particiones a gestionar.

## FAT 32

El sistema FAT32 puede direccionar hasta  $2^{28}$  clústeres. De los 32 bits de direccionamiento de la tabla se usan 28 realmente para direccionar clústeres. Un clúster puede tener  $2^n$  sectores, donde  $n$  es un valor entre 0 y 6. El tamaño máximo de una partición vendrá limitado por el tamaño de los clústeres, pero en la práctica no se admiten clústeres mayores de 32 KB (salvo excepciones).

De este modo, el tamaño máximo teórico de la partición podría ser de 8 TiB ( $2^{28} \times 32\text{KiB}$ ), pero en la práctica el límite de una partición FAT32 es de 2 TB y el tamaño máximo de archivo de 4 GB.




---

NTFS son las siglas de *New Technology File System*, y algunas de sus características son:

- 64 bits para direccionar bloques. Tamaño máximo de la unidad: 16 HexaBytes.
- Los nombres de archivos pueden tener 256 caracteres y son almacenados en código UTF-16.
- Permite definir el tamaño del clúster a partir de 512 bytes, independientemente del tamaño de la partición.
- Aunque en teoría puede manejar volúmenes mayores, en la práctica el **máximo del volumen NTFS está en torno a los 16 TB** (con clústeres de 4 KB).
- Es un sistema adecuado para particiones de gran tamaño. De hecho, no se recomienda utilizar NTFS en volúmenes menores de 400 MB por el espacio de los archivos del sistema, que pueden ocupar unos 4 MB sobre una partición de 100 MB.
- El tamaño mínimo recomendado para una partición es de 10 GB, y para cada volumen, de 2 TB.
- El tamaño máximo del fichero estará fijado por el tamaño del volumen.

- Toda la información relativa a los ficheros se almacena en forma de “**metadatos**”. Todos los ficheros tienen al menos dos atributos, un nombre de fichero y un “flujo de datos” (con los datos del fichero), pero pueden tener más.
  - Soporta compresión y cifrado de ficheros.
  - Existen varias versiones de NTFS (v4.0, ..., v6.0) y las más recientes incluyen características como las cuotas de disco y los puntos de montaje de volúmenes.
- 

El sistema UFS, **Unix File System**, es utilizado por varios sistemas UNIX y POSIX, y por muchas distribuciones de Unix, adaptándolo a todas ellas y dando lugar a muchos otros.

Esto se debe en parte al gran número de distribuciones Unix y Linux existentes.

Linux puede trabajar con muchos sistemas de ficheros, como por ejemplo ext2, ext3, ReiserFS, XFS, JFS, UFS, ISO9660, FAT, FAT32 o NTFS, y otros orientados para el trabajo en red, como NFS (entre equipos Linux) o SMB (entre equipos Linux y Windows). Algunos de ellos son:

### **Ext2**

Muy estable y con una fragmentación muy baja. Es un poco lento con archivos grandes, pero puede manejarlos. Trabaja con particiones de disco de hasta 4 TB y con ficheros de hasta 2 GB, y admite nombres de ficheros de hasta 255 caracteres.

### **Ext3**

Es compatible con el anterior, de hecho puede hacerse la actualización de Ext2 a Ext3. Es prácticamente un estándar en Linux. Incorpora facilidades de fiabilidad y mantenimiento.

## **Ext4**

Es más eficiente que las anteriores versiones; aporta más rendimiento a la CPU y más velocidad de lectura/escritura. Puede trabajar con ficheros de hasta 16 TB y particiones de hasta 1024 PB (PetaBytes).

## **ReiserFS**

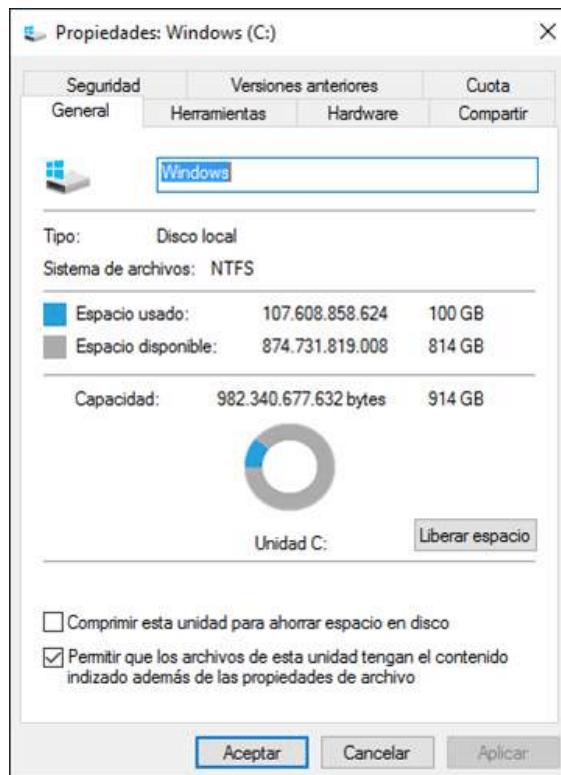
Es el más actual para Linux y de máxima eficiencia, pero al ser “nuevo” muchas herramientas no lo soportan para, por ejemplo, recuperar datos.

# ¿Cuál es mi sistema?

¿Cómo saber qué sistema de ficheros tiene mi disco duro?

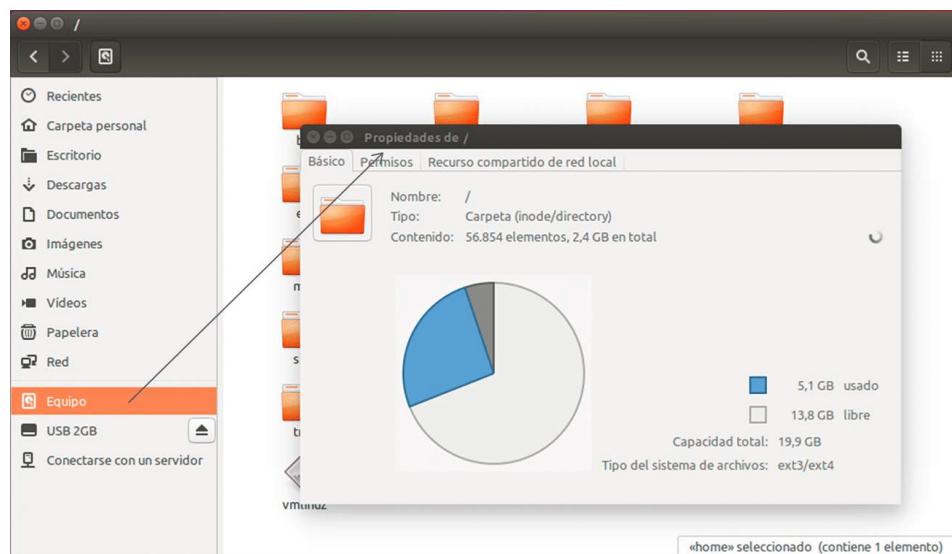
Si estás trabajando sobre un ordenador moderno bajo Windows lo más probable es que el sistema empleado sea **NTFS**.

Podemos verlo abriendo el administrador de archivos, colocando el ratón sobre la unidad del disco y seleccionando con el botón derecho “propiedades” del disco. En la imagen podemos ver un ejemplo realizado sobre Windows 10.



## ¿Y cómo veo cuál es mi sistema de archivos en Ubuntu?

Una forma muy fácil es utilizar el administrador de archivos de Ubuntu, ir a "equipo" y con el botón derecho del ratón ver las "propiedades".



Por supuesto, en Linux ya sabes que toda la información se puede ver también por línea comando de terminal, y existen muchos comandos que nos dan información sobre el sistema de ficheros que tenemos instalado. Te ponemos algunos ejemplos (recuerda que nuestra máquina virtual tiene su disco virtual montado en /dev/sda1) que nos muestran que nuestro sistema de archivos es "ext4".

```
usuario@ubuntu:~$ 
usuario@ubuntu:~$ df -Th
S.ficheros  Tipo      Tamaño Usados  Disp Uso% Montado en
udev        devtmpfs  983M    0  983M  0% /dev
tmpfs       tmpfs     202M   8,9M  193M  5% /run
/dev/sda1   ext4      19G   4,8G  13G  28% /
tmpfs       tmpfs     1006M  252K  1006M  1% /dev/shm
tmpfs       tmpfs     5,0M   4,0K  5,0M  1% /run/lock
tmpfs       tmpfs     1006M  0  1006M  0% /sys/fs/cgroup
/dev/loop0   squashfs  78M    78M  0  100% /snap/core/4108
/dev/loop1   squashfs  143M  143M  0  100% /snap/notepadqq/116
/dev/loop2   squashfs  78M    78M  0  100% /snap/core/4208
tmpfs       tmpfs     202M   84K  202M  1% /run/user/1000
/dev/sdb1   vfat      1,9G  203M  1,7G  11% /media/usuario/USB 2GB
usuario@ubuntu:~$
```

```
usuario@ubuntu:~$ lsblk -f
NAME   FSTYPE  LABEL      UUID                                     MOUNTPOINT
loop1  squashfs
loop2  squashfs
loop0  squashfs
sda
└─sda2
└─sda5_swap  swap      d33dbef1-1fa4-44e1-a9f4-cd87adc65306 [SWAP]
└─sda1 ext4    0bf3a2c7-06a4-4548-8211-a28eaf11c747 /
usuario@ubuntu:~$
```

```
usuario@ubuntu:~$ cat /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=0bf3a2c7-06a4-4548-8211-a28eaf11c747 /          ext4    errors=remount-ro  1
# swap was on /dev/sda5 during installation
UUID=d33dbef1-1fa4-44e1-a9f4-cd87adc65306 none        swap    sw            0      0
/dev/fd0      /media/floppy0 auto    rw,user,noauto,exec,utf8 0      0
usuario@ubuntu:~$
```

```
usuario@ubuntu:~$ fsck -N /dev/sda1
fsck de util-linux 2.27.1
[ /sbin/fsck.ext4 (1) -- / ] fsck.ext4 /dev/sda1
usuario@ubuntu:~$
```

 Aprovechamos para comentarte un detalle sobre Linux.

En los sistemas Linux normalmente dentro del árbol de directorios tendremos:

- Punto de montaje (acceso) a los **discos de tipo IDE** instalados:

`/dev/hd*` (donde el "\*" puede ser una letra y un número o solo una letra).

- Punto de montaje (acceso) a los **discos de tipo SCSI o NO IDE** instalados:

`/dev/sd*` (donde el "\*" igualmente puede ser una letra y un número o solo una letra).

Sin embargo, también es cierto que hay muchas distribuciones Linux que tratan tanto a los discos IDE como SCSI o SATA del mismo modo y los están ubicando en `/dev/sd*`.

En general, debajo de `/dev` tendremos los dispositivos físicos y virtuales del sistema y, por ejemplo, se suelen definir algunos subdirectorios como:

- `/dev/fd` - Para las disqueteras de los discos flexibles o "*floppy disks*", ya en desuso.
- `/dev/null` - Directorio virtual "nulo", no almacena nada que pongamos en él.
- `/dev/sr` - Para conectar dispositivos SCSI especiales, como un grabador de DVD.
- `/dev/loop` - Dispositivos que trabajan en modo carácter y no en bloque.

No te preocupes si te parece que Linux tiene muchos comandos con muchas posibilidades. Aquí solamente te mostramos unos ejemplos, pero dedicaremos una sección específica a conocer los comandos más usados en la gestión de archivos.

# Creación de particiones

---

La **creación de particiones** es una de las operaciones que podemos realizar con particiones del disco.

## Particiones primarias

Para crear una partición primaria necesitamos disponer de espacio libre (no particionado) en el disco. Si está totalmente vacío la partición puede ocupar el total del disco, a excepción de la parte que ocupe el gestor de arranque (MBR o GPT). Si la partición primaria va a contener el sistema operativo no podrá estar en cualquier sitio, sino que su inicio debe situarse dentro de los primeros 2 GB del disco. Esto es lo que se conoce como “**límite de arranque**”.

## Particiones extendidas y lógicas

Las particiones lógicas estarán dentro de una partición extendida, que habremos tenido que crear primero, aunque algunas aplicaciones las crean automáticamente al solicitar la creación de la partición lógica.

## Particiones de seguridad

Es bastante común, sobre todo en equipos portátiles, la creación de una partición primaria o lógica dedicada a contener información de recuperación del sistema. Su creación sigue las mismas reglas que cualquier otra, aunque se suele recomendar ubicarla al final de la lista de particiones.

# Borrado de particiones

---

El **borrado de particiones** es otra de las operaciones que podemos realizar con particiones del disco.

Cuando eliminamos una partición en un disco **se pierde toda la información** contenida en ella (programas y datos). Además, es posible que cambien las letras asignadas a las unidades, lo que podría hacer que las aplicaciones instaladas en otras particiones dejaran de funcionar.

En algunos casos podemos elegir entre eliminar las particiones de forma normal, o bien **eliminarlas de forma segura**. Hacerlo de forma segura significa que **esa partición no podrá ser recuperada y sus datos tampoco podrán recuperarse** (ni con herramientas dedicadas). La opción de eliminación segura o permanente se suele utilizar cuando se quieren destruir los datos por motivos de seguridad o confidencialidad.

No siempre es posible realizar la recuperación de particiones borradas, porque el proceso tiene muchas restricciones que a menudo no permiten hacerlo.

# Redimensionado de particiones

---

El **redimensionado de particiones** es una operación que nos permite aumentar o reducir el espacio de una partición.

Aunque cuando se crean se asigna un espacio a cada partición, y esto se hace en función de la configuración del disco que necesitemos en ese momento, es posible que con el uso del equipo alguna partición se nos haya quedado pequeña, o que en otra nos sobre espacio y decidamos redimensionarlas.

Para poder aumentar el espacio de una partición **debemos tener espacio disponible en el disco**. Pero disponer de espacio no particionado en el disco no es habitual, puesto que si ese espacio no está asignado a una partición no se puede utilizar. Por eso **deberemos primero reducir el tamaño de alguna de las otras particiones para poder ampliar otra**.

Si además tenemos varias particiones y la que queremos reducir no es contigua a la que queremos ampliar, es posible que tengamos que reducir primero el espacio en la última, por ejemplo, y desplazar las demás hasta que podamos asignar a la primera el espacio libre.

# Fusión de particiones

---

La fusión de particiones es una operación que  
**sirve para unir dos particiones del disco.**

Para aumentar el tamaño de una partición a costa de otra siempre podemos eliminar la que no nos interesa, dejar el espacio disponible y luego ampliar la primera.

Algunas herramientas de gestión de discos nos permiten también realizar la fusión de las dos particiones directamente, teniendo en cuenta que si los sistemas de archivos de ambas son diferentes y compatibles, **durante el proceso de fusión habrá que hacer la conversión al sistema de la partición que se va a conservar**, y también conservar los datos que hubiese en esa partición.

# Copia de particiones

---

Entre las operaciones de partición que podemos llevar a cabo en el disco, una de ellas es **copiar particiones entre varios discos.**

Copiar una partición implica generar una nueva partición con el mismo tamaño, sistema de archivos y contenido que la partición original.

Las herramientas de gestión nos permiten copiar particiones entre varios discos, esto es, que la partición de origen esté ubicada en un disco y la copia en otro.

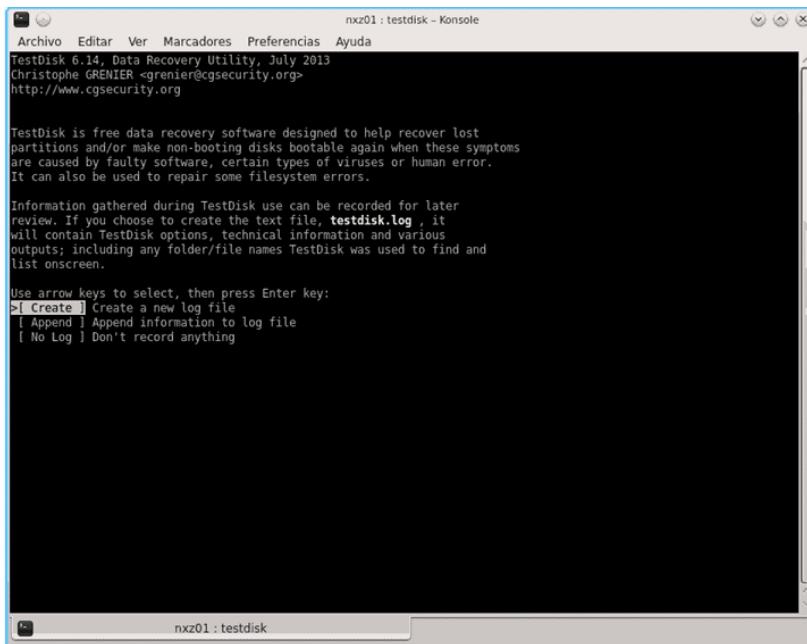
Es evidente que el disco destinado a almacenar la copia debe disponer de espacio contiguo suficiente para albergarla. En caso contrario, la operación podría no realizarse correctamente o dañar al resto de las particiones del disco destino.

# Recuperación de particiones

La recuperación de particiones sirve para **reparar daños en el disco**.

Normalmente, para poder recuperar particiones y volúmenes será necesario disponer de **alguna copia de seguridad en otro dispositivo**. Si es así, el fabricante del sistema operativo o del equipo nos suele proporcionar herramientas y asistentes para realizar la recuperación.

En ocasiones el daño solamente afecta a la tabla de particiones o al registro de arranque maestro, o es posible buscar los archivos en el disco localizando las firmas de archivos conocidos.



Existen herramientas que nos permiten recuperar particiones dañadas, siempre y cuando el daño no sea excesivo. Actúan leyendo a bajo nivel las particiones y recuperando el sistema de archivos.

# Conversión de particiones

---

La conversión de particiones se puede referir a dos acciones diferentes:

1

Al **cambio de tipo de partición**. Por ejemplo, pasar una partición primaria a partición lógica (dentro de una extendida). Aquí sería aplicable todo lo relacionado antes con los cambios en particiones.

2

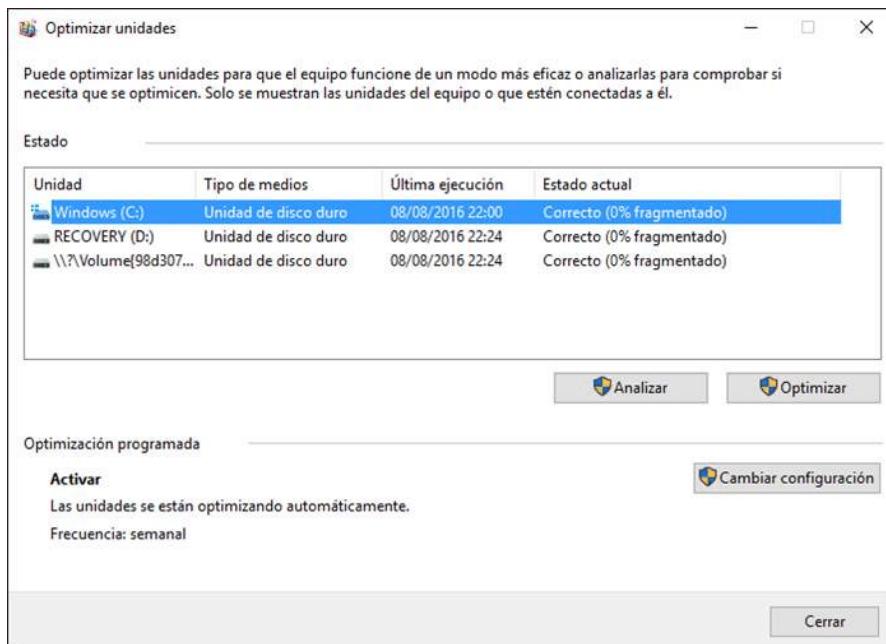
O bien se refiere al **cambio del sistema de archivos** para montar uno nuevo. Se pueden hacer conversiones desde unos sistemas a otros, FAT, FAT32, NTFS, etc. A menudo el propio sistema operativo nos ofrece comandos y utilidades para realizar esta acción por línea de comandos.

# Desfragmentación

Desfragmentar el disco significa **ordenar el contenido de sus particiones** para mejorar el rendimiento del equipo.

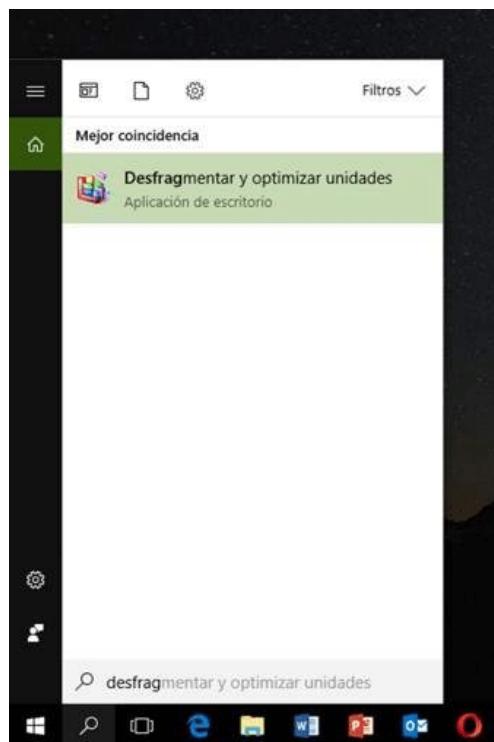
El uso continuado de cualquier sistema de ficheros produce **fragmentación** en los archivos. Esto significa que el **sistema coloca diferentes porciones de los ficheros de forma más o menos desordenada en el disco, en vez de ponerlas de forma contigua**. Esta práctica ocasiona, a la larga, mayor lentitud en el acceso a los archivos.

La desfragmentación puede ser más o menos necesaria según la eficiencia del algoritmo de asignación de bloques del sistema de ficheros. Es bastante común en sistemas Windows, y el propio sistema operativo nos proporciona herramientas para realizar la desfragmentación. Es menos común en sistemas Unix/Linux, pero también hay formas de hacerla.



A menudo hablamos de desfragmentar el disco, pero realmente lo que estamos haciendo es **desfragmentar una unidad lógica**, o lo que es lo mismo, una partición del disco (que puede no ser la única).

## Desfragmentar el disco en Windows

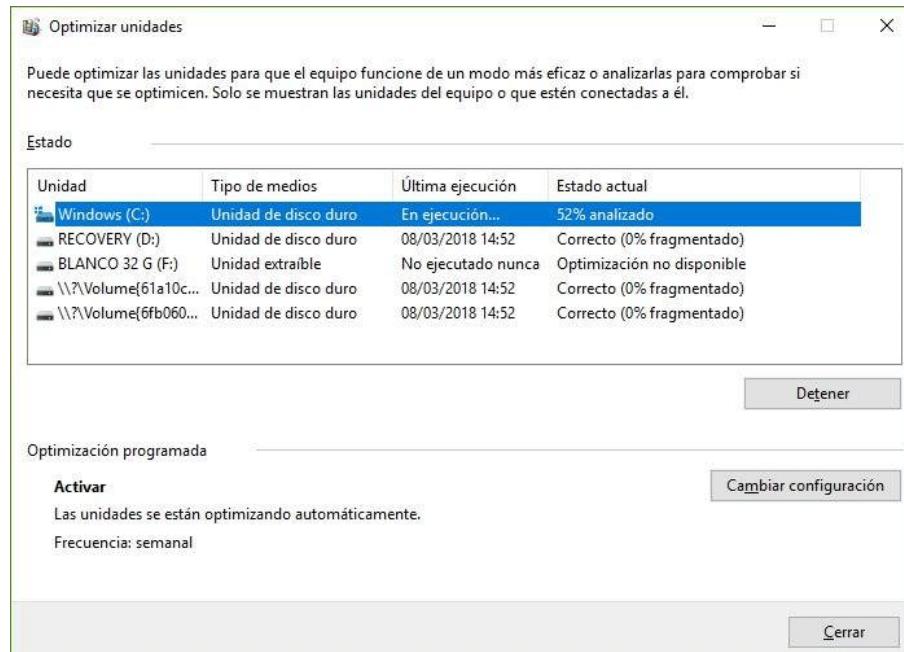


Proceder a desfragmentar el disco en Windows es sencillo, aunque el proceso puede llevar un poco de tiempo en función del tamaño y lo "desordenado" (fragmentado) que esté el disco a optimizar.

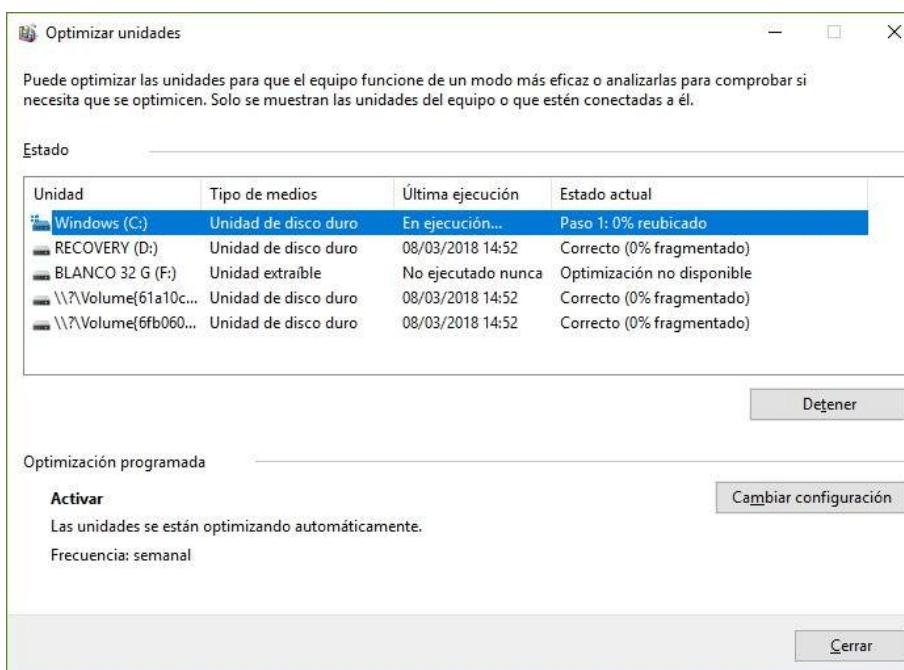
Para hacerlo podemos escribir "desfragmentar" en el cuadro de búsqueda de Windows y nos aparecerá la aplicación del sistema que se encarga de optimizar el disco.

Al activarla seleccionaremos la unidad que queremos desfragmentar y dando a aplicar comenzará el proceso.

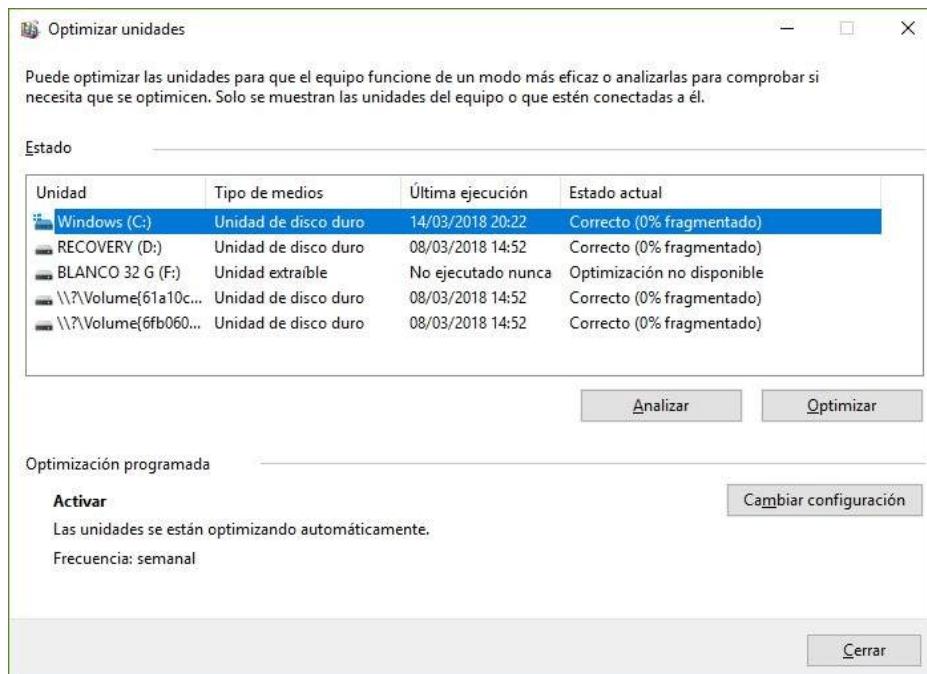
En las pantallas siguientes te mostramos la secuencia resumida y la pantalla final, donde nos dice que está 0% fragmentado y la fecha de la última optimización.



Proceso de análisis para la desfragmentación.



Proceso de reubicado.



Finalización del proceso de fragmentación.

---

Si te fijas en las pantallas, habrás notado que Windows 10 incorpora la actividad de desfragmentación de forma automática, y si no queremos que sea así debemos cambiar la configuración.

## Desfragmentar en Linux (Ubuntu)

En muchos casos se dice que en Linux la desfragmentación no es necesaria, pues los sistemas de ficheros que utiliza son muy robustos y además tienen esquemas de optimización inteligente del espacio. Sin embargo, aunque la desfragmentación puede no ser tan necesaria como en otros sistemas, siempre es conveniente con el tiempo verificar el estado de nuestras particiones y llegado el caso hacer una desfragmentación.

Para realizarlo de forma fácil podemos utilizar la herramienta "e4defrag" que se encuentra dentro del paquete SW "e2fsprogs". La buena noticia es que suele venir ya instalado en las modernas distribuciones de Ubuntu (y si no habría que instalarlo por comando o con el gestor de paquetes Synaptic, por ejemplo).

```

usuario@ubuntu:~$ sudo e4defrag -c /dev/sda1
<Fragmented files>
now/best      size/ext
1. /var/log/wtmp        20/1       4 KB
2. /var/log/auth.log    11/1       4 KB
3. /home/usuario/.cache/google-chrome/Default/Cache/index
                           13/1       4 KB
4. /var/log/apt/term.log 12/1       4 KB
5. /var/log/auth.log.1   17/1       4 KB

Total/best extents          109828/109475
Average size per extent     46 KB
Fragmentation score         0
[0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
This device (/dev/sda1) does not need defragmentation.
Done.
usuario@ubuntu:~$ █

```

Primero comprobaremos el estado de desfragmentación de nuestro volumen con:

***sudo e4defrag -c /dev/sda1***

Donde, si te has fijado, hemos puesto el punto de montaje de nuestro disco (*/dev/sda1*), pero podemos poner cualquier otro.

Al finalizar nos dará un porcentaje de fragmentación ("Fragmentation score" en la imagen) que para valores bajos indicará que no es necesario desfragmentar y si en cambio es alto (p. ej. por encima de 50) entonces sí deberemos desfragmentar.

En el caso de que queramos desfragmentar el disco podemos dar el comando:

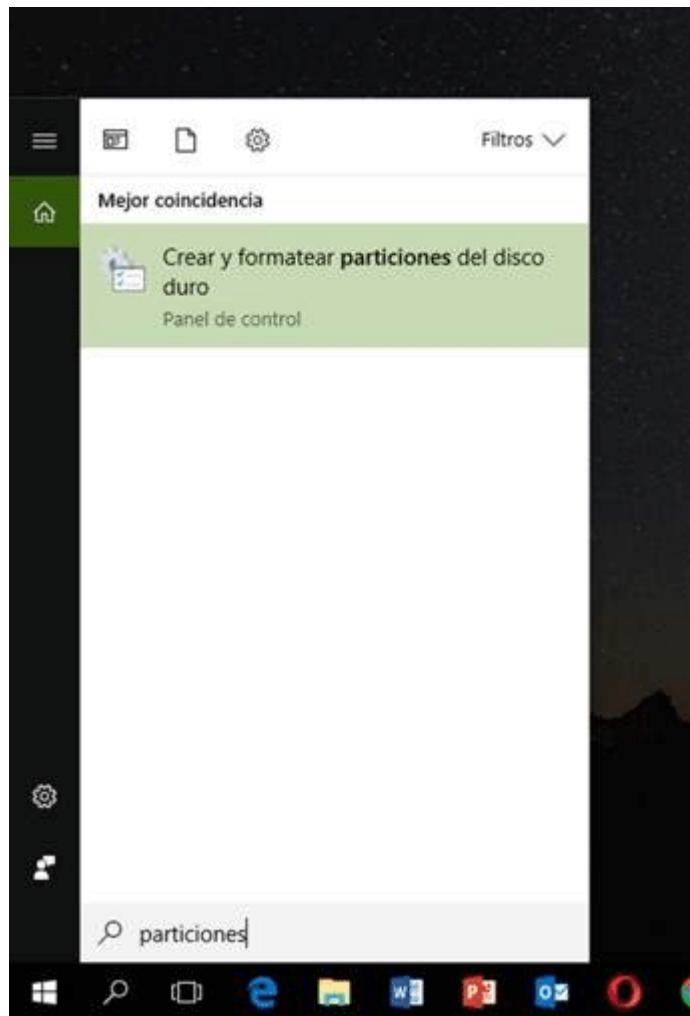
***sudo e4defrag /dev/sda1***

Al introducir el comando el sistema se pondrá a trabajar, recorriendo todos los directorios dependientes del punto de inicio (volumen o dispositivo) que le hemos dado, listando el proceso en la ventana del terminal hasta que termina.

## Ejercicio/problema práctico

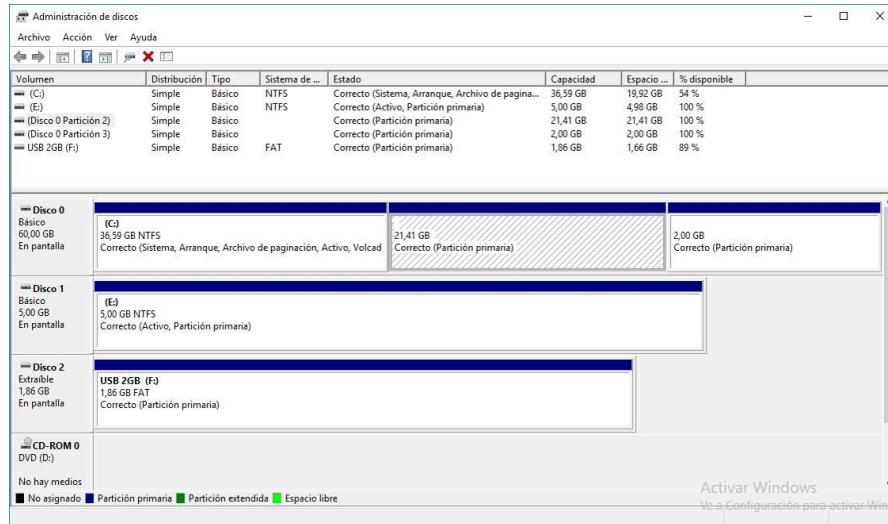
---

Te planteamos un pequeño "problema" para hacerte razonar un poco. Para resolverlo debes emplear la máquina virtual con arranque dual de Windows 10 y Ubuntu, que habrás creado antes en este curso.



1. Arranca la máquina virtual y selecciona como sistema Windows 10 dentro del gestor de arranque GRUB.
2. Cuando haya arrancado Windows 10 completamente activa la aplicación de administración de discos. Recuerda que puedes hacerlo mediante el cuadro de búsqueda de Windows y haciendo clic sobre la opción de crear particiones, o a través del panel de control.
3. Al activar la aplicación de gestión de las particiones te debe salir algo parecido a la imagen que te mostramos abajo. Quizás no te salga la unidad USB si no tienes conectada ninguna, evidentemente.

Asegúrate de que esto lo haces sobre la máquina virtual de arranque dual Windows 10/Ubuntu, no sobre otra MV de Windows 10.



## Problema

Ve seleccionando con el ratón los diferentes volúmenes/particiones dentro de los discos.

Verás que al seleccionarlos se ponen con rayas. Intenta entonces pulsar con el botón derecho del ratón para desplegar el menú y ver sus propiedades. Hazlo sobre todas las particiones.

- ¿Te permite ver las propiedades de todas ellas?
- ¿Te has fijado que algunas de ellas tienen un sistema de ficheros instalado y otras no?
- ¿Puedes instalar un sistema de ficheros en todas las que no lo tienen?
- ¿Te obliga en algún caso a eliminar el volumen?

## ¿Por qué ocurre lo anterior?

Bien, seguramente ya te lo has imaginado, pero el hecho de tener una instalación dual significa que cada uno de los sistemas operativos que hemos instalado (Windows 10 y Ubuntu) ha creado sus propias particiones y uno no reconoce las del otro.

En este caso verás que no es lo mismo tener una partición "sin sistema de ficheros" pero instalada por el sistema operativo en funcionamiento, que una partición de la que no reconocemos su sistema de ficheros y que ha sido instalada por otro sistema operativo.

Si te cuesta entender estos conceptos no te preocupes, a lo largo del curso verás que te resultan más fáciles.

# Del sistema de ficheros a las estructuras de directorios

---

Un **sistema de ficheros** utiliza normalmente el **"bloque"** como unidad de acceso al dispositivo de almacenamiento (por ejemplo un disco) y como forma de independizar las aplicaciones de la estructura física del dispositivo (que utiliza cilindros, pistas y sectores).

Además, la implementación de un sistema de archivos sobre un disco plantea **necesidades de gestión a nivel del sistema operativo**, como por ejemplo:

- Gestión de los directorios y los nombres (rutas, enlaces, etc.).
- Traducción de direcciones. Para el acceso a un “registro lógico” de un fichero es necesario encontrar los bloques físicos que lo contienen.
- Gestión de la seguridad, el acceso y los permisos de los usuarios.
- Gestión de los canales (independencia del dispositivo) y los accesos concurrentes a ficheros y colas.
- Gestión del espacio libre en el dispositivo para aumentar la eficiencia y reducir todo lo posible el tiempo de acceso.
- Gestión de la caché de bloques en memoria.
- Gestión de errores y mantenimiento de la consistencia del sistema de ficheros.

En algunos sistemas, como UNIX y Linux, la misma gestión de ficheros se aplica a otros elementos que no son archivos propiamente dichos, como por ejemplo tuberías (*pipes*) y colas FIFO, que se manejan desde el sistema de ficheros como ficheros especiales, y lo mismo se hace con los dispositivos de entrada/salida.

# Montar volúmenes en carpetas

---

Un volumen es algo diferente de una partición, aunque en algunos entornos se usa el término con significados intercambiables.

En realidad, un **volumen** es una unidad lógica a la que accedemos desde el sistema operativo y, en función del S.O. que estemos usando, se gestionará de forma algo diferente.

Dentro de estas unidades lógicas (volúmenes) ubicaremos los archivos de datos y programas a utilizar. Para acceder a ellos debemos conocer la **ruta** (“*path*”) que hace referencia al punto en el que se encuentran dentro del sistema de archivos.

Las rutas se expresan mediante una cadena de caracteres, formada por los nombres de los directorios o carpetas a través de los que se puede acceder al último, que contiene el archivo y el nombre del archivo. Estos nombres de directorios se indican:

- Separados por una barra inversa (\) en entorno Windows.
- Separados por una barra normal (/) en sistemas UNIX o Linux.
- Separados por un carácter (:) en sistemas MAC OS clásicos.

La **ruta absoluta** marca el camino desde el directorio raíz del sistema de archivos, mientras que una ruta **relativa** indica el camino a partir de la posición actual en la que nos encontramos dentro del sistema de archivos. Los directorios se van anidando unos dentro de otros, formando una estructura a la que normalmente se le conoce como **árbol jerárquico** o **árbol del directorio**, y en la que encontramos carpetas reservadas por el sistema operativo, además de las que puede crear el usuario.

# Volúmenes en entorno Windows

---

En el entorno Windows los volúmenes coinciden con las unidades lógicas identificadas por alguna letra (C:, D:, F:, etc.), ya estén montadas sobre una partición del disco o sean una unidad de disco extraíble o un *pendrive*.

En los sistemas basados en Windows NT no existe un único directorio raíz, sino que se asigna una ruta individual (“path”) para montar cada volumen, que puede ser identificada de dos formas:

- Por una letra seguida de “:”, por ejemplo: “C:”.
- Por un “punto de montaje” sobre un sistema NTFS y con una letra, por ejemplo: “C:\documents”.

Para acceder a un fichero (p. ej. “Informe.doc”) almacenado en el directorio raíz de estos volúmenes, la ruta a indicar sería: “C:\Informe.doc” o bien “C:\documents\Informe.doc”.

Por defecto, Windows reserva las letras “A:” y “B:” para unidades de disco extraíbles, y el resto a partir de “C:” para los demás dispositivos (disco duro, unidades ópticas, memorias vía USB, etc.). Pero como los ordenadores ya no montan unidades de “floppy disk”, las letras A y B pueden reasignarse a otras unidades.

Hay que tener en cuenta que si una ruta de acceso a una unidad cambia o desaparece, las aplicaciones configuradas para acceder a través de esa ruta dejarán de funcionar.

# Volúmenes en entorno Unix/Linux

---

En los sistemas Unix todos los volúmenes que no sean el de arranque (“*boot volume*”) tienen un **punto de montaje** en algún lugar del sistema de archivos.

Este punto de montaje se representa también por una ruta (“*path*”), que sigue una estructura jerárquica separada por “/”.

Bajo ese punto de montaje se inserta el **árbol de directorios almacenados en el volumen**. Si se quiere usar una ruta como punto de montaje para otro volumen hay que disponer de un directorio vacío en ella, al que se suele llamar “carpeta” o “*folder*”.

Por ejemplo, si una unidad de DVD que contiene un fichero llamado “*video.avi*”, se encuentra montada en una ruta como “*/mnt/isodvd*”, se puede acceder al fichero a través de la ruta: “*/mnt/isodvd/video.avi*”.

En sistemas Unix la ruta “*~/*” es una ruta relativa que lleva al directorio personal del usuario, mientras que si se indica “*./*” se está haciendo referencia al directorio actual, y con “*../*” nos referimos a un nivel más atrás en la jerarquía (a partir de donde estemos).

## Ejercicio sobre Ubuntu

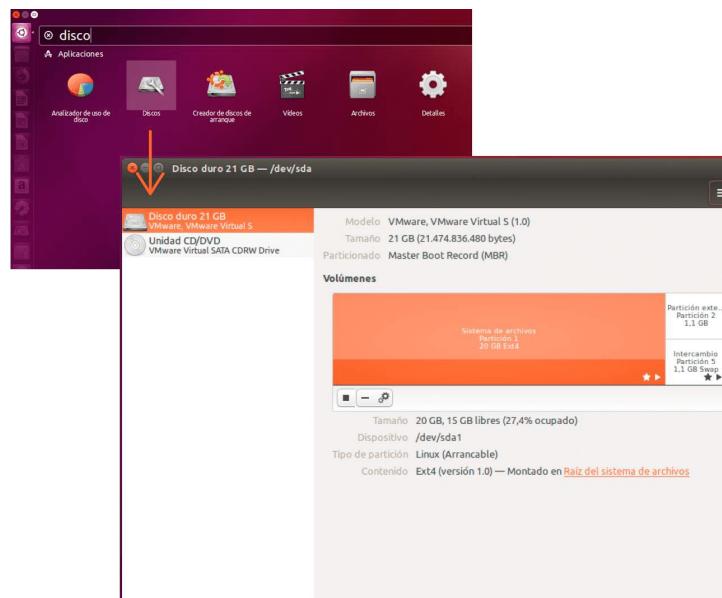
Para que compruebes si los conceptos de partición, volumen, ruta, etc. te resultan comprensibles, te pedimos que trabajes con el propio sistema haciendo lo siguiente:



Utiliza la "Guía de Escritorio de Ubuntu" (recuerda que se despliega en la esquina superior derecha) para buscar lo relacionado con "volumen".

Del resultado elige la opción de ayuda sobre "Gestionar volúmenes y particiones".

Ahí te dirá que busques una aplicación en el cuadro de búsqueda de Ubuntu (esquina superior izquierda, pinchando sobre el símbolo de Ubuntu).



Buscando la aplicación de gestión del disco y activándola te saldrá la ventana que te mostramos aquí.

Haz una lectura de la ayuda que aparece en la ventana y comprueba si entiendes todo lo que dice, y si es coherente con lo explicado en esta lección.

# Directarios y subdirectarios

Cada sistema de archivos tiene una forma de organizar el almacenamiento de los ficheros, generalmente agrupados en **directarios** y **subdirectarios** que se enlazan en árbol de forma jerárquica.



```
Símbolo del sistema
C:\Windows>dir
El volumen de la unidad C es Windows.
El número de serie del volumen es: 36F6-29C1

Directorio de C:\Windows

10/03/2018 11:48    <DIR>      .
10/03/2018 11:48    <DIR>      ..
29/09/2017 14:46    <DIR>      addins
17/12/2017 16:16    <DIR>      appcompat
14/02/2018 00:07    <DIR>      apppatch
14/03/2018 11:49    <DIR>      AppReadiness
08/01/2018 21:59    <DIR>      assembly
28/08/2016 03:22    0 ativpsrm.bin
14/02/2018 00:07    <DIR>      bcastdvr
29/09/2017 14:41    65.536 bfsvc.exe
29/09/2017 14:46    <DIR>      Boot
29/09/2017 14:46    <DIR>      Branding
27/03/2017 20:48    66 Brfaxrx.ini
27/03/2017 20:58    92 brpcfx.ini
28/03/2017 17:33    232 Brpxf04a.ini
14/03/2018 12:09    <DIR>      CbsTemp
16/12/2017 15:29    22.382 comsetup.log
29/09/2017 14:42    35.138 Core.xml
10/07/2015 12:00    31.816 CoreSingleLanguage.xml
17/12/2015 11:24    12 CSUP.txt
29/09/2017 14:46    <DIR>      Cursors
13/02/2018 19:42    <DIR>      debug
14/03/2018 11:59    <DIR>      DeliveryOptimization
16/12/2017 15:32    7.623 diagerr.xml
29/09/2017 14:46    <DIR>      diagnostics
16/12/2017 15:32    7.623 diagwrm.xml
30/09/2017 15:41    <DIR>      DigitalLocker
```

Algunos de estos directarios son creados por el propio S.O. durante su instalación, otros durante la instalación de las diferentes aplicaciones de usuario. Además el propio usuario podrá crear sus propios directarios para el almacenamiento de sus archivos y programas.

El usuario puede acceder a los directorios a través de una interfaz de comandos (como la que te mostramos en la figura) o mediante entornos gráficos (GUI).

Hemos de tener en cuenta también que **muchos dispositivos son accesibles como simples ficheros dentro el árbol de directorios**. Algunos sistemas operativos permiten incluso que un mismo archivo sea accesible desde distintos subdirectorios, o que un mismo fichero tenga varios nombres en subdirectorios diferentes, para lo cual no es necesario duplicar la información, sino establecer los enlaces al mismo recurso desde diferentes sitios.

# Ejercicio: directorios en Linux

---

Vamos a practicar cómo movernos por el árbol de directorios de Ubuntu Linux.

Dado que moverse con la interfaz gráfica es muy fácil, vamos a practicar con comandos de terminal; de esta forma verás que también es fácil moverse a través de la consola.

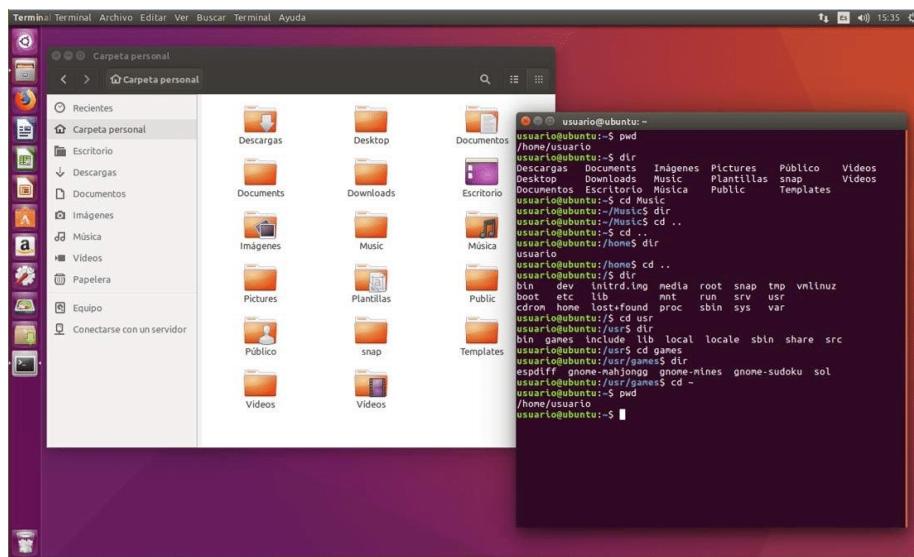
Te pedimos que introduzcas en un terminal de tu máquina virtual Ubuntu la secuencia de comandos que te mostramos abajo, y que al mismo tiempo, con el administrador de archivos gráfico, vayas comprobando que el contenido de cada subdirectorio se corresponde con el resultado de los comandos.

Secuencia de comandos:

- pwd
- dir
- cd music
- dir
- cd ..
- cd ..
- dir
- cd
- dir
- cd usr

- dir
- cd games
- dir
- cd ~
- pwd

El resultado final debe ser parecido al de la siguiente pantalla:



Te has estado moviendo de un directorio a otro, superior o inferior pero inmediatamente adyacentes. Introduce ahora el siguiente comando desde tu directorio personal:

`cd /usr/games`

¿Qué ha ocurrido? Simplemente que, como ves, es posible moverse de una carpeta a otra que puede estar en cualquier punto del sistema si especificamos la ruta completa.

Te animamos a seguir viajando por las rutas y subdirectorios del sistema.

# Resumen

---

Has finalizado esta lección.

- El sistema operativo organiza el almacenamiento de los ficheros y los agrupa en **directorios** y **subdirectorios** dentro de una estructura jerárquica (“en árbol”).
- Existen diferentes **sistemas de ficheros**, como FAT/FAT32 (MS-DOS, Windows), NTFS (Windows), UFS/FFS, ZFS (UNIX), etc.
- Las operaciones sobre las particiones del disco podemos realizarlas a través de **comandos** o mediante **aplicaciones dedicadas**.
- Las diferentes operaciones con particiones que podemos realizar son **creación**, **borrado**, **redimensionado**, **fusión**, **copia**, **recuperación**, **conversión** y **desfragmentación**.
- Un sistema de ficheros utiliza normalmente el **bloque** como unidad de acceso al dispositivo de almacenamiento.
- Para poder usar un disco duro y almacenar información, primero hay que **particionarlo**, es decir, **crear las particiones** (conjunto de cilindros contiguos que forman una unidad lógica).
- La **ruta absoluta** marca el camino desde el directorio raíz del sistema de archivos, mientras que una **ruta relativa** indica el camino a partir de la posición actual en la que nos encontramos dentro del sistema.
- Todo fichero/archivo, además de su nombre, tiene asociados una serie de datos sobre sí mismo que conocemos como **atributos del archivo** o **metadatos**.

