

# Gestión de discos, particiones y volúmenes en Ubuntu 20.04 LTS

---

En este capítulo veremos en detalle la gestión de discos y particiones. A continuación veremos los volúmenes lógicos, las cuotas de disco y las copias de seguridad.

---



*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	Discos y particiones.....	1
1.1	Identificación de dispositivos.....	2
1.2	Discos.....	3
1.3	Gparted.....	5
1.3.1	Crear una partición.....	6
1.3.2	Cambiar el tamaño de una partición.....	7
1.4	Gestión de particiones. Utilidad fdisk.....	8
1.5	Crear sistema de archivos. Comando mkfs.....	10
1.6	Información del espacio ocupado o libre en una partición.....	13
1.7	Reparación de un sistema de archivos.....	14
1.8	Etiquetas.....	15
2	Monitor del sistema.....	15
3	Copias de seguridad.....	16
3.1	duplicity.....	16
3.1.1	Realizar una copia de seguridad con duplicity.....	18
3.1.2	Restauración de una copia con duplicity.....	21
3.2	rsync.....	22
3.2.1	Sincronizar con rsync.....	25
3.2.2	La barra al final de los nombres de directorio.....	27
3.2.3	Eliminando ficheros del destino de backups anteriores.....	28
3.2.4	Impedir actualizar archivos modificados en destino.....	29
3.2.5	Backups incrementales.....	29
3.2.6	Incluyendo y excluyendo ficheros.....	31
3.2.7	Copias remotas.....	34
4	Cuotas de disco.....	34
4.1	Activar las cuotas en el sistema de archivos.....	35
4.2	Crear los archivos de cuota y la tabla de uso de espacio compartido.....	36
4.3	Configurar cuotas para usuarios y grupos.....	36
4.4	Establecer un valor para el periodo de gracia.....	38
4.5	Comprobar las cuotas de disco.....	38
4.6	Desactivar y volver a activar las cuotas.....	39
4.7	Avisos de cuotas excedidas.....	40
5	LVM.....	40
5.1	Crear los volúmenes físicos.....	42
5.2	Crear el grupo de volúmenes.....	43
5.3	Expandir el grupo de volúmenes.....	43
5.4	Crear volumen lógico lineal.....	44
5.4.1	Opciones adicionales en la creación de volúmenes lógicos.....	45
5.5	Crear sistema de archivos y montar volúmenes lógicos.....	45
5.6	Redimensionar un volumen lógico.....	46

5.6.1 Aumentar el tamaño de un volumen lógico.....	46
5.6.2 Reducir el tamaño de un volumen lógico.....	47
5.7 Visualizar información del sistema LVM.....	48
5.7.1 Visualizar información sobre volúmenes físicos.....	48
5.7.2 Visualizar información de grupos de volúmenes.....	50
5.7.3 Visualizar información de los volúmenes lógicos.....	50
5.8 Eliminar un volumen lógico.....	52
5.9 Eliminar un volumen físico.....	52
5.10 Eliminar un grupo de volúmenes.....	54
5.11 RAID.....	54
5.11.1 RAID 0.....	54
5.11.2 RAID 1.....	55
5.11.3 RAID 5.....	55
5.12 Instantáneas.....	55
5.12.1 Crear una instantánea.....	56
5.12.2 ¿Cómo funciona una instantánea?.....	58
5.12.3 Revertir el volumen lógico a la instantánea.....	59
6 Bibliografía.....	61

# Gestión de discos, particiones y volúmenes en Ubuntu 20.04 LTS

## 1 Discos y particiones

---

La gestión de discos incluye la creación y gestión de particiones en discos duros. Además, podemos establecer unas cuotas a los usuarios de forma que tengan un límite en cuanto al uso del espacio del disco duro.

En Linux existen varias herramientas para gestionar las particiones del disco duro. Desde comandos en línea hasta aplicaciones en formato gráfico que facilitan esta delicada tarea. Partiendo de que en el disco duro se dispone de espacio sin asignar, para poder disponer de dicho espacio a la hora de almacenar información hay que realizar tres operaciones:

1. Crear la partición con las herramientas que veremos a continuación.
2. Crear en la partición nueva un sistema de archivos.
3. Montar la partición en un directorio, bien manualmente o configurando el montaje de el archivo `/etc/fstab`.

Para ver la gestión de discos y particiones vamos a emplear diferentes métodos. En los siguientes epígrafes veremos algunos de ellos, comenzando por la forma que tiene Linux de identificar los dispositivos de almacenamiento.

## 1.1 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.

## 1.2 Discos

En Ubuntu disponemos de la herramienta *Discos*. Podemos abrir esta aplicación desde

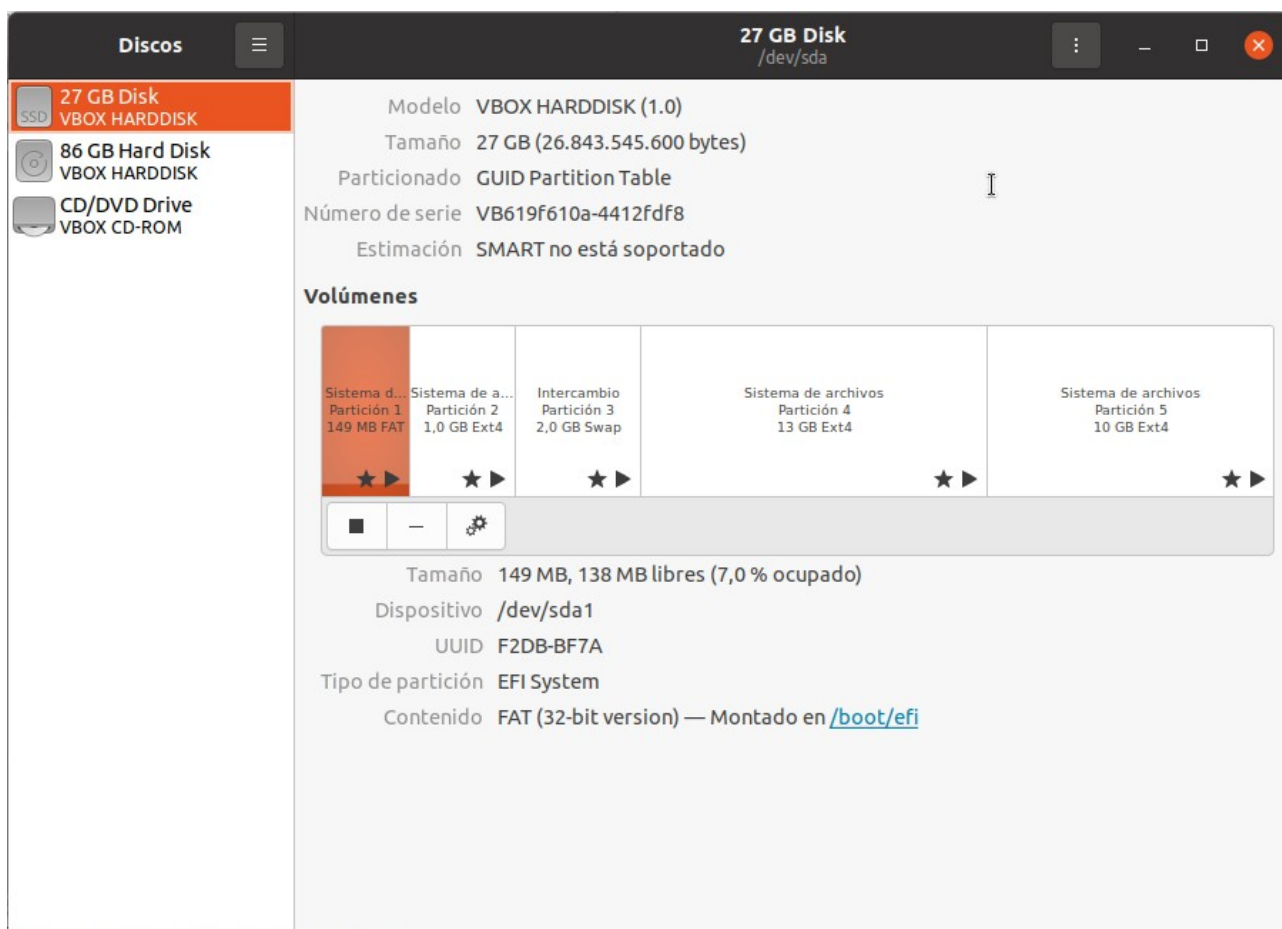


Figura 1.- Discos




el tablero.

En la parte izquierda hay un panel con las unidades de disco que hay en el sistema. Cuando seleccionamos una unidad de disco duro nos aparece en el panel de la derecha la información relativa a la unidad y las particiones que hay definidos en ella.

Como se puede apreciar en la figura, presenta un gráfico en forma de barra horizontal donde representa las particiones existentes en el disco, el sistema de archivos de cada una y su tamaño relativo.

Si hacemos clic sobre una partición veremos debajo del gráfico la información relativa a la misma. Debajo del gráfico veremos una pequeña barra de herramientas para gestionar las

particiones:

- ✓ El botón  se emplea para desmontar la partición.
- ✓ El botón  es para eliminar la partición.
- ✓ El botón  muestra un menú con las siguientes opciones:
  - x Formatear.- Da formato a la partición seleccionada.
  - x Editar partición.- Permite cambiar el tipo de partición y activar la marca de arranque.
  - x Editar sistema de archivos.- Sirve para asignar una etiqueta a la partición.
  - x Editar las opciones de montaje.-
  - x Crear imagen de disco.- Se emplea para crear un archivo que contiene una imagen exacta actual de la partición.
  - x Restaurar imagen de disco.- Restaura una partición a partir de un archivo de imagen creado con la opción anterior.


En el caso de hacer clic en el gráfico donde está el espacio libre sin asignar solamente aparece un botón  para crear una nueva partición. Cuando hacemos clic en él comienza el asistente para crear una partición y veremos la siguiente ventana.



Figura 2.- Crear partición

1. Hay que indicar el tamaño que tendrá la partición. Por defecto aparece todo el espacio libre disponible contiguo. Cuando lo hayamos especificado hacemos clic en

*Siguiente.*

2. Escribimos el nombre del volumen e indicamos el sistema de archivos a crear en la nueva partición. Si dejamos desactivado *Borrar* hará un formato rápido.

Figura 3: Nombre de volumen y sistema de archivos

3. Hacemos clic en el botón *Crear*. Antes de crearla nos pedirá que nos autentifiquemos como usuario administrador.

Si al elegir como sistema de archivos hubiéramos elegido *Otro* nos aparece una lista para seleccionar el sistema de archivos para formatear la partición.

Una vez creada la partición podemos hacer clic en el botón ▶ para montarla. Por defecto la montará en el directorio `/media/usuario/etiqueta`, siendo `usuario` el nombre de la cuenta de usuario y `etiqueta` la etiqueta de la partición que dimos cuando la creamos. En este ejemplo hemos utilizado `DATOS`. Si queremos montarla en otro directorio tendremos que editar el archivo `/etc/fstab` y añadir una línea para el nuevo volumen.

## 1.3 Gparted

Gparted es una aplicación gráfica para gestión de discos y particiones tradicional del gestor de escritorio GNOME. Por defecto no está instalada, pero podemos instalarla en el centro de software. Una vez hecho podemos ejecutarla y veríamos la siguiente pantalla.



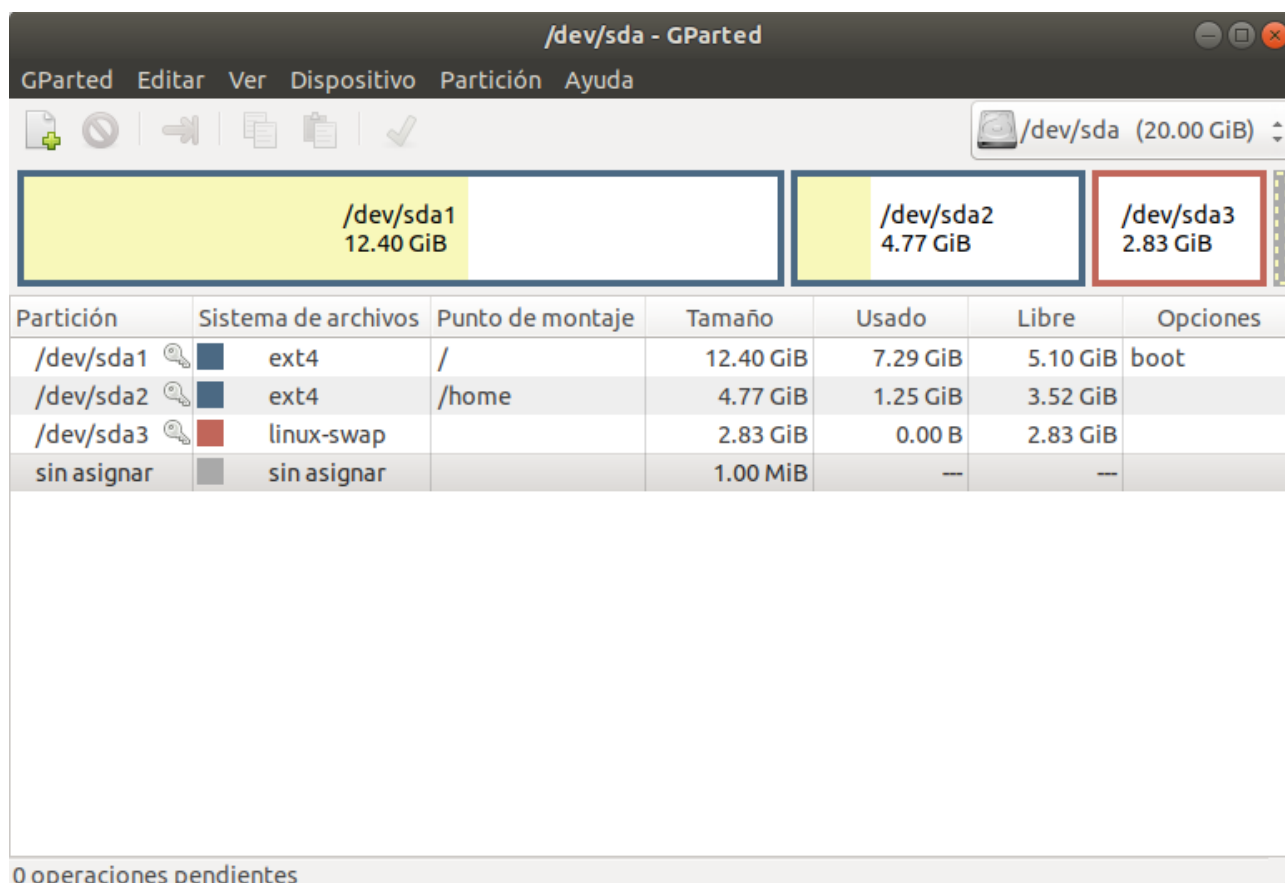


Figura 4.- Pantalla principal de Gparted

Debajo de la barra de menú aparece la barra de herramientas con botones para las operaciones clásicas en la gestión de particiones: crear, eliminar, expandir, etc. A la misma altura en la parte derecha aparece la lista de discos. Cuando seleccionamos uno nos muestra sus particiones en el área central en dos formatos: gráfico y lista. En este último nos indica información sobre la partición como el archivo de dispositivo, el sistema de archivos, espacio libre y ocupado junto con el punto de montaje.

Si el disco que estamos gestionando no disponemos de una tabla de particiones la creamos con la opción de menú *Dispositivo* → *Crear tabla de particiones...* Seleccionamos el esquema de particionamiento (*msdos* para MBR y *gpt* para GPT) y hacemos clic en el botón *Aplicar*.

### 1.3.1 Crear una partición

Para crear una partición seguiremos el siguiente proceso:

1. Seleccionar el disco en la lista de disco.
2. Hacer clic en el botón *Añadir partición*.
3. En el cuadro de diálogo que se abre rellenar los datos de la nueva partición: Tamaño, etiqueta, sistema de archivos, espacio antes y después.



Figura 5.- Creación de partición

4. Hacer clic en el botón *Añadir*.
5. En la barra de herramientas hacer clic en el botón *Aplicar todas las operaciones*.
6. En la lista de particiones seleccionarla y darle formato con la opción de menú *Partición → Formatear como → Sistema de archivos*.

### 1.3.2 Cambiar el tamaño de una partición

Para cambiar el tamaño de una partición seguiremos los siguientes pasos:

1. Seleccionar la partición a extender.
2. Hacer clic en el botón *Redimensionar/Mover partición la partición seleccionada*.

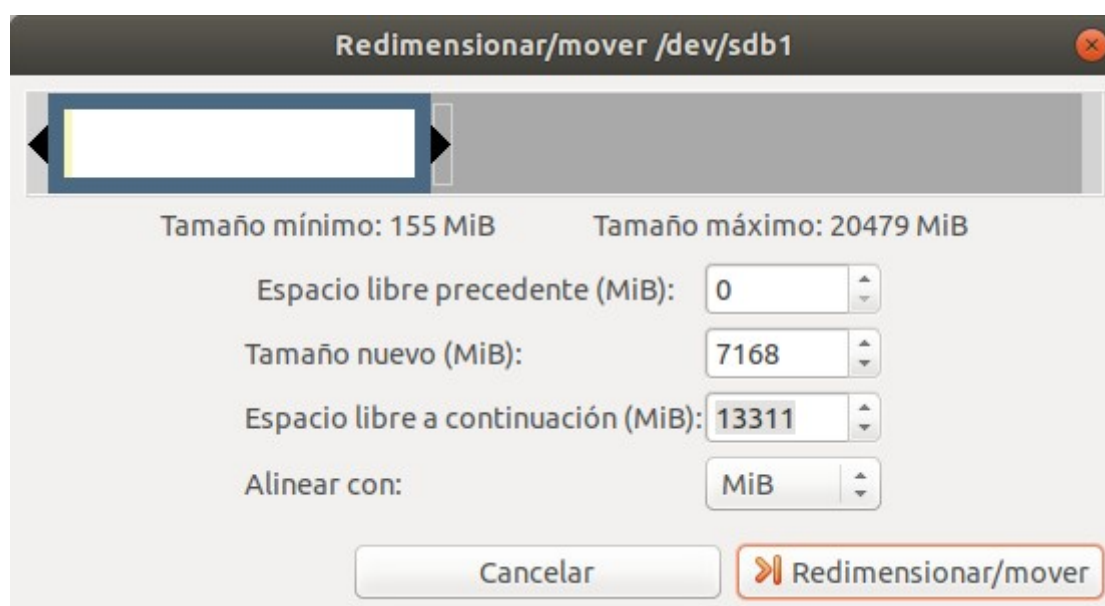


Figura 6.- Redimensionar partición

3. Hacer clic en el botón *Redimensionar/mover*.

4. En la barra de herramientas hacer clic en el botón *Aplicar todas las operaciones*.

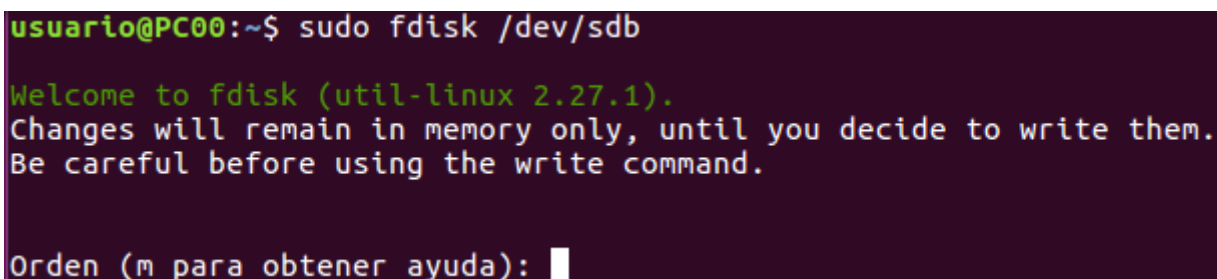
## 1.4 Gestión de particiones. Utilidad fdisk

En todas las distribuciones Linux está la utilidad en línea de comando `fdisk` la cual podemos utilizar tanto para discos con esquema de particionamiento MBR como GPT. Esta utilidad tiene una interfaz textual y presenta menús para gestionar las particiones del disco. Se podrán crear y eliminar particiones, asignarle un tipo, etc. Hay que utilizarla con mucho cuidado, ya que no es tan intuitiva como una herramienta gráfica, por tanto existe una mayor probabilidad de cometer errores que serían fatales.

Su funcionamiento es a través de un menú con órdenes a través de teclado. Para iniciarla desde la línea de comando se ejecuta:

```
fdisk [archivo_dispositivo]
```

El archivo de dispositivo será el que corresponda al disco duro del cual se gestionarán las particiones. En el caso más habitual será `/dev/sda`.



```
usuario@PC00:~$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Orden (m para obtener ayuda):
```

Figura 7.- Ejecución de fdisk

Una vez iniciada podremos ver el menú de acciones cada vez que pulsamos la letra `m` + Intro. Para salir se teclea la acción `q` + Intro.

```
Help:

DOS (MBR)
a  toggle a bootable flag
b  edit nested BSD disklabel
c  toggle the dos compatibility flag

Generic
d  delete a partition
F  list free unpartitioned space
l  list known partition types
n  add a new partition
p  print the partition table
t  change a partition type
v  verify the partition table
i  print information about a partition

Misc
m  print this menu
u  change display/entry units
x  extra functionality (experts only)

Script
I  load disk layout from sfdisk script file
O  dump disk layout to sfdisk script file

Save & Exit
w  write table to disk and exit
q  quit without saving changes

Create a new label
g  create a new empty GPT partition table
G  create a new empty SGI (IRIX) partition table
o  create a new empty DOS partition table
s  create a new empty Sun partition table
```

Figura 8.- Menú de fdisk

Podemos ver el estado actual de la tabla de particiones pulsando la acción *p* + Intro. Nos ofrece un listado en el que aparece una línea por cada partición del disco con los siguientes campos:

- ✓ Dispositivo.- Archivo de dispositivo que corresponde a la partición
- ✓ Inicio. Tendrá un \* para la partición de arranque.
- ✓ Comienzo. Número de sector donde comienza la partición.
- ✓ Fin. Número de sector donde acaba la partición.
- ✓ Bloques. Número de bloques de la partición.
- ✓ Id. Identificador del tipo de partición.

- ✓ Sistema. Sistema de archivos de la partición.

Además, antes de la tabla de particiones muestra información sobre el disco, como: Archivo de dispositivo, tamaño, geometría, tamaño del sector, etc.

Para crear una partición hay que utilizar la acción  $n$  + Intro. Habrá que indicar:

- ✓ Tipo de partición:  $p$  primaria y  $l$  lógica.
- ✓ N° de la partición.- Para particiones primarias será un número entre 1 y 3 si hay definida una partición extendida o entre 1 y 4 si no la hay.
- ✓ Primer sector.- N° de sector inicial de la partición. Por defecto aparece el primero del espacio que hay sin asignar.
- ✓ Último sector.- N° de sector final de la partición. Por defecto aparece el último del espacio sin asignar. Resulta más recomendable indicarlo con el tamaño de la partición que se podrá hacer con un  $+númeroM$  o  $+numeroK$ , el sufijo M o K sirve para indicar si este número es MB o KB.

Una vez definida la partición tendrá un tipo por defecto que es *Linux* (identificador 83). Con la acción  $t$  + Intro se puede indicar el tipo de partición. Los tipos de particiones se pueden listar con  $l$  + Intro. Se dispone de una gran cantidad de tipos de particiones reconocibles por Linux.

```
Orden (m para obtener ayuda): p
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xb013ecab

Disposit.  Inicio    Start    Final Sectores  Size Id Tipo
/dev/sda1  *          2048    487423    485376    237M 83 Linux
/dev/sda2          487424  30486527  29999104  14,3G 83 Linux
/dev/sda3    30488574  62912511  32423938  15,5G  5 Extendida
/dev/sda5    30488576  58914815  28426240  13,6G 83 Linux
/dev/sda6    58916864  62912511  3995648   1,9G 82 Linux swap / Solaris
```

Figura 9.- Listado de las particiones

Por último, para grabar los cambios, es necesario emplear la acción  $w$  + Intro, que escribe la nueva tabla de particiones en el sector de inicio del disco duro.

## 1.5 Crear sistema de archivos. Comando mkfs

El comando `mkfs` se emplea para crear un sistema de archivos en un dispositivo.

### Sintaxis

```
mkfs [-t tipo] [opciones fs] dispositivo
```

## Parámetros

### `dispositivo`

Archivo de dispositivo que corresponde a la partición donde se creará el sistema de archivos.

## Opciones

### `-t tipo`

Tipo de sistema de archivos. Pueden ser cualquiera que Linux reconoce. Los más habituales son:

- ✓ `ext4`. El actual de Linux.
- ✓ `ext3`. La versión anterior de `ext`.
- ✓ `ext2`. Por defecto si no se indica ninguno.
- ✓ `vfat`. FAT
- ✓ `msdos`. FAT pero con nombres de archivo de 8 caracteres.

### `opciones fs`

Conjunto de opciones específicas del tipo de sistema de archivos que se está creando. Comunes a todos ellos es `-c` para comprobar los bloques y `-v` para visualizar información del progreso de la operación.

En realidad, el comando `mkfs` es un front-end para varios constructores de sistemas de archivos, específicos del tipo de sistema de archivos y que son comandos en la forma `mkfs.fstype`, siendo `fstype` el tipo de sistema de ficheros. Los más habituales son `mkfs.ext4` para sistema de archivos `ext4` y `mkfs.vfat` para sistema de archivos FAT.

## Sintaxis

```
mkfs.ext4 [opciones] dispositivo
```

## Parámetros

### `dispositivo`

Archivo de dispositivo de la partición.

## Opciones

### `-b tamaño`

Tamaño de bloque. Puede ser 1024, 2048 o 4096.

### `-c`

Chequea el dispositivo buscando bloques en mal estado antes de crear el sistema de ficheros

### `-L etiqueta`

Asigna la etiqueta al dispositivo

Para sistemas de archivo FAT disponemos de los siguientes comandos.

### Sintaxis

```
mkfs.vfat [opciones] dispositivo.  
mkfs.msdos [opciones] dispositivo.  
mkdosfs [opciones] dispositivo.
```

### Parámetros

#### dispositivo

Archivo de dispositivo de la partición.

### Opciones

#### -F tamaño

Especifica el tipo de FAT. Puede ser 12, 16 o 32. Por defecto es 12 o 16 dependiendo del tamaño de la partición.

#### -f copias

Número de copias de la tabla FAT. Por defecto son 2.

#### -n etiqueta

Asigna una etiqueta al nuevo volumen

#### -c

Chequea el dispositivo buscando bloques en mal estado antes de crear el sistema de ficheros

#### -s sectores por cluster

Número de sectores de disco por cada cluster o unidad de asignación. Tiene que ser una potencia de 2.

Finalmente, el comando `mkntfs` crea un sistema de archivos NTFS.

### Sintaxis

```
mkntfs [opciones] dispositivo.
```

### Parámetros

#### dispositivo

Archivo de dispositivo de la partición.

### Opciones

#### -C, --enable-compression

Habilita la compresión para los archivos del dispositivo.

`-f, -q, --fast, --quick`

Realiza un formato rápido

`-L etiqueta`

Asigna una etiqueta al nuevo volumen

`-c, --cluster-size BYTES`

Establece el tamaño del cluster a BYTES. El valor puede estar entre 256 y 65536 bytes.

## 1.6 Información del espacio ocupado o libre en una partición.

El comando `du` informa sobre la cantidad de espacio en disco que emplea un archivo.

### Sintaxis

```
du [opciones] [archivo ...]
```

### Parámetros

`archivo ...`

Lista de archivos separados por espacios.

### Opciones

`-k`

La salida en kilobytes. Por defecto, visualiza el tamaño de cada archivo en bloques de 512 Bytes

`-m`

La salida en megabytes.

`-s`

Si el nombre de archivo es un subdirectorio estimará el tamaño de su contenido, pero no hará un recorrido recursivo.

`-c, --total`

Al final visualiza el total de espacio ocupado de todos los archivos.

Si se pasa como argumento el archivo de dispositivo de una partición, ofrece el tamaño que ocupan los archivos de la partición. Si no se indica una lista de archivos ofrece la ocupación del directorio activo.

El comando `df` se emplea para visualizar el espacio usado y libre de los sistemas de ficheros.

### Sintaxis



```
df [opciones] [archivo ...]
```

## Parámetros

`archivo ...`

Lista de archivos separados por espacios. De cada archivo ofrecerá información del sistema de archivos donde se encuentra. Si no se indica informa de todos los sistemas de archivos montados.

## Opciones

`-k`

El espacio ocupado está en Kilobytes.

`-h`

El espacio ocupado está en Megabytes y Gigabytes

`-H`

Similar al anterior, pero usando múltiplos de 1000 en lugar de 1024

`-t tipo`

Solamente informa sobre los sistemas de archivos del tipo especificado

El listado incluye una línea por sistema de archivos con los siguientes campos:

- ✓ Sistema de archivos o archivo de dispositivo.
- ✓ Bloques de 1Kb totales.
- ✓ Bloques usados.
- ✓ Bloques disponibles.
- ✓ % de uso.
- ✓ Directorio de montaje.

## 1.7 Reparación de un sistema de archivos.

El comando `fsck` chequea y repara un sistema de archivos Linux

## Sintaxis

```
fsck [opciones] [sistema_de_archivos]
```

## Parámetros

`sistema_de_archivos`

Archivo de dispositivo o directorio de montaje del sistema de archivos a chequear. Resulta conveniente que esté sin montar

## Opciones

-a

Repara si es necesario y sin preguntar al usuario.

Si no se indica ningún sistema de archivos, chequeará todos los que haya en el archivo `/etc/fstab` en el mismo orden.

## 1.8 Etiquetas

Podemos asignar una etiqueta a un dispositivo de almacenamiento. Generalmente, esta asignación se puede hacer cuando se crea el sistema de archivos. En cada tipo de sistema de archivos hay una opción que permite asignar una etiqueta al dispositivo.

Además, contamos con comandos que permiten tanto visualizar la etiqueta del dispositivo como cambiarla. En función del tipo de sistema de archivos disponemos de diferentes comandos para realizar esta tarea. Los vemos a continuación:

El comando `e2label` muestra o asigna una etiqueta a un dispositivo con sistema de archivos ext2,3 o 4.

### Sintaxis

```
e2label dispositivo [etiqueta]
```

### Parámetros

`dispositivo`

Archivo de dispositivo del cual tiene que mostrar o asignar una etiqueta

`etiqueta`

Etiqueta a asignar. Si se omite muestra la actual

## 2 Monitor del sistema

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

La información que proporciona se encuentra dividida en varias pestañas. La pestaña que nos muestra el uso de los recursos de disco es *Sistema de archivos*. Aquí podemos ver las particiones de los discos en el sistema, junto con la cantidad de espacio libre y ocupado de los mismos.



Figura 10.- Monitor del sistema de archivos

## 3 Copias de seguridad

Para realizar copias de seguridad disponemos de potentes y flexibles herramientas en línea de comando, como `rsync` y `duplicity`, así como herramientas gráficas que dependen de la distribución concreta que utilicemos. Estas herramientas suelen ser front-ends que ocultan a los usuarios la complejidad del uso de una utilidad en línea de comando.

### 3.1 duplicity

La utilidad en línea de comando `duplicity` permite realizar backups incrementales de archivos y carpetas en un formato empaquetado y encriptado para luego almacenarlos en diferentes tipos de ubicaciones, locales o remotos.

#### Sintaxis

```
duplicity [full|incremental] [opciones] directorio url_destino
duplicity [restore] [opciones] url_destino directorio
duplicity list-current-files [opciones] url_destino
```

#### Parámetros

##### full

Realiza una copia de seguridad completa. Por defecto si se omite el tipo de copia realizará una copia completa primero y las siguientes serán incrementales. Para forzar a hacer una copia completa usar esta opción.

##### incremental

Realiza una copia de seguridad incremental. Por defecto si se hizo una copia completa anteriormente.

##### restore

Realiza una restauración de una copia de seguridad.

##### directorio

Directorio que va a copiar o donde se va a restaurar, dependiendo de si se hace una copia o una restauración

## url\_destino

Destino de la copia u origen de la restauración. Puede ser una carpeta local, un servidor FTP, un servidor SSH, etc. El destino se indica con una URL con la siguiente sintaxis

protocolo://[usuario[:password]@]host[:puerto]/[/]path

El protocolo indica el tipo de destino de la copia. Duplicity admite múltiples destino de copia para los que habría que indicar un protocolo concreto. Alguno de ellos es

- ✓ file:// → Carpeta local o remota montada en el sistema de archivos.
- ✓ scp:// o sftp:// → Servidor SSH
- ✓ ftp[s]:// → Servidor FTP
- ✓ dpbx:// → Dropbox
- ✓ gdocs:// → Google Docs
- ✓ onedrive:// → OneDrive

## list-current-files

Lista los archivos actuales incluidos en la copia de seguridad indicada en el destino

### Opciones

#### --exclude patrón

Se excluyen las carpetas y ficheros cuyo nombre se ajusta al patrón indicado. Pueden especificarse varias opciones --exclude.

#### --exclude-filelist fichero

Se excluyen las carpetas y ficheros que se encuentran en el fichero indicado, donde cada línea es un path de archivo o carpeta a excluir de la copia.

#### --include patrón

Se incluye en la copia las carpetas y ficheros cuyo nombre se ajusta al patrón indicado. Pueden especificarse varias opciones --include.

#### --include-filelist fichero

Se incluye en la copia las carpetas y ficheros que se encuentran en el fichero indicado, donde cada línea es un path de archivo o carpeta a incluir de la copia.

#### --file-to-restore path

En una restauración solamente se restaurará la carpeta o fichero indicado en el path

#### --no-encryption

No se encripta la copia. Si no se indica la copia se encripta por defecto, con lo que pedirá la contraseña.

#### --progress

En copias de seguridad remotas visualiza un progreso de la copia y un tiempo estimado

de copia.

Podemos omitir la acción que indica si es una copia o una restauración ya que `duplicity` puede detectarla automáticamente en función de los parámetros pasados. Las opciones indicadas son las más habituales, pero existen muchas más. Para conocer con mayor detalle estas y otras opciones consultar la página de manual de `duplicity`.

### 3.1.1 Realizar una copia de seguridad con duplicity

En las opciones que emplean patrón podemos utilizar los metacaracteres habituales al forma patrones en UNIX/Linux. Además se dispone del metacarácter `**` el cuál actúa como `*` pero incluyendo también el carácter `/`.

Este comando se comporta de un modo un tanto particular que debemos entender para conseguir nuestro objetivo de realizar copias de seguridad. En principio vemos que el comando solamente admite un directorio para realizar la copia y luego con opciones `--include` y `--exclude` se añaden o quitan más directorios en la copia. Es fundamental entender como funcionan estas opciones ya que su significado no es tan intuitivo como parece. En principio la copia es del directorio indicando como argumento. Los directorios que ponemos con las opciones de inclusión y exclusión deben de ser subdirectorios del directorio principal, nunca directorios fuera del principal. Por ejemplo, vamos a realizar una copia del directorio `datos` que se encuentra en la carpeta principal.

```
usuario@U2004:~$ duplicity --no-encryption ~/datos
file:///media/usuario/BACKUP/
Los metadatos en local y remoto están sincronizados, no es necesario
sincronizar.
Fecha del último respaldo completo: ninguna
Firmas no encontradas, cambiando a un backup completo.
-----[ Estadísticas de respaldo ]-----
StartTime 1541345243.45 (Sun Nov  4 16:27:23 2018)
EndTime 1541345249.53 (Sun Nov  4 16:27:29 2018)
ElapsedTime 6.08 (6.08 seconds)
SourceFiles 59
SourceFileSize 144774382 (138 MB)
NewFiles 59
NewFileSize 144774382 (138 MB)
DeletedFiles 0
ChangedFiles 0
ChangedFileSize 0 (0 bytes)
ChangedDeltaSize 0 (0 bytes)
DeltaEntries 59
RawDeltaSize 144753902 (138 MB)
TotalDestinationSizeChange 144502465 (138 MB)
Errors 0
-----
```

En el ejemplo anterior hemos añadido la opción `--no-encryption` ya que por defecto encripta la copia y para ello pedirá una clave de encriptación. Esta clave será necesaria utilizarla para recuperar la copia.

Suponemos que el usuario tiene conectado un pendrive con etiqueta BACKUP. De ahí que esté montado en `/media/usuario/BACKUP` y lo usemos como destino de la copia de seguridad.

No hemos tenido que indicar el tipo de copia de seguridad. Al no haber ninguna previa crea una copia completa. A partir de ahora si omitimos el tipo de copia de seguridad empleará una incremental, al detectar una copia completa previa.

Si vemos el destino de la copia `duplicity` ha creado unos archivos empaquetados y comprimidos con toda la información a copiar y los metadatos necesarios para gestionar las copias. Sin embargo, podemos saber cuales son estos archivos ejecutando `duplicity` con la siguiente opción.

```
usuario@U2004:~$ duplicity list-current-files
file:///media/usuario/BACKUP/
Los metadatos en local y remoto están sincronizados, no es necesario
sincronizar.
Fecha del último respaldo completo: Sun Nov  4 16:27:23 2018
Sun Nov  4 16:17:17 2018 .
Sun Nov  4 16:15:50 2018 3.png
Sun Nov  4 16:15:50 2018 4.png
Sun Nov  4 16:15:51 2018 BDGC.sql
Sun Nov  4 16:15:51 2018 Control de acceso.sql
Sun Nov  4 16:15:51 2018 Creación Esquema BLOG.sql
Sun Nov  4 16:15:51 2018 Fronton (base).sb2
...
```

El listado se ha acortado para simplificar el resultado.

Veamos ahora como podemos realizar copias excluyendo algunos subdirectorios del directorio principal. En el directorio datos disponemos de cuatro subdirectorios (`sub1`, `sub2`, `sub3` y `sub4`) de los cuales vamos a excluir los dos últimos. Tendríamos que ejecutar lo siguiente

```
usuario@U2004:~$ duplicity --no-encryption --exclude ~/datos/sub3 --
exclude ~/datos/sub4 ~/datos file:///media/usuario/BACKUP/
```

Aquí es donde `duplicity` se vuelve un tanto quisquilloso. Si hubiéramos ejecutado este comando habríamos obtenido un error.

```
usuario@U2004:~$ duplicity --no-encryption --exclude '~/datos/sub3'
--exclude '~/datos/sub4' ~/datos file:///media/usuario/BACKUP/
```

El problema radica en que hemos empleado la expansión de tilde `~` para indicar los subdirectorios a excluir y además van entrecomillados. Si empleamos expansión de tilde `~` o la variable de entorno `$HOME` no podemos entrecomillar los subdirectorios. Sin embargo, en el caso de que los nombres de los subdirectorios tengan espacios en blanco no tenemos más remedio que entrecomillar o utilizar el carácter de escape. En este caso, por tanto, deberemos de utilizar path absoluto sin expansión de tilde.

Esta segunda copia, sobre una copia completa anterior, se ha hecho incremental al

detectar una copia completa previa y no indicar con el argumento `full` que la hiciera completa. Si vemos el contenido del destino de la copia







Nombre	Tamaño
 <code>duplicity-full.20181104T152723Z.manifest</code>	168 bytes
 <code>duplicity-full.20181104T152723Z.vol1.diff.tar.gz</code>	144,5 MB
 <code>duplicity-full-signatures.20181104T152723Z.sig.tar.gz</code>	1,0 MB
 <code>duplicity-inc.20181104T152723Z.to.20181104T154533Z.manifest</code>	197 bytes
 <code>duplicity-inc.20181104T152723Z.to.20181104T154533Z.vol1.diff.tar.gz</code>	292 bytes
 <code>duplicity-new-signatures.20181104T152723Z.to.20181104T154533Z.sig.tar.gz</code>	304 bytes

Figura 11.- Copias completa e incremental

Por cada copia hay tres archivos. Los nombra utilizando la fecha y hora de realización de la copia. Se observa que la primera copia ocupa mucho más (unos 145 MB) que la segunda (unos pocos bytes), ya que la segunda es la copia incremental y solo introduce lo que ha cambiado respecto a la primera.

Aparentemente, las opciones `--include` no tienen utilidad, ya que cuando se hace copia de un directorio se incluyen también sus subdirectorios, mientras que las opciones `--exclude` servirían para quitar subdirectorios de la copia que no se necesitan. Sin embargo, debemos tener en cuenta que solamente podemos incluir subdirectorios del directorio principal de la copia, por tanto, en el caso de que queramos hacer copia de seguridad de subdirectorios fuera del directorio principal deberemos indicar como directorio principal el directorio padre y mediante opciones `--include` y `--exclude` indicar que directorios queremos incluir y cuáles no. Por ejemplo, imaginemos que queremos hacer copia de los directorios `Documentos` e `Imágenes` del directorio personal del usuario. En este caso tendríamos que ejecutar el siguiente comando

```
usuario@U2004:~$ duplicity full --no-encryption --include
~/Documentos --include ~/Imágenes/ --exclude '/home/usuario/**' ~
file:///media/usuario/BACKUP/
```

Analicemos la ejecución anterior. Estamos haciendo una copia del directorio personal del usuario en el que hemos incluido los directorios `~/Documentos` e `~/Imágenes`, ya que únicamente queremos copiar estos directorios. Además, hemos tenido que poner una opción `--exclude` para que omita todo lo demás. El valor de este argumento ha sido `'/home/usuario/**'` que tiene un significado especial. Los valores de los argumentos en opciones `--include` o `--exclude` admiten metacaracteres. Hay un metacarácter adicional `**` el cuál actúa como `*` pero incluyendo la `/`. De esta forma un `**` significa cualquier cadena que se pudiera formar, incluyendo la `/` de separación de directorios en un path. Al usar metacaracteres `duplicity` obliga a entrecomillar los paths y por tanto no podemos usar la expansión de tilde.

Hemos puesto la opción `full` para realizar una copia completa ya que estamos haciendo una copia con un origen diferente al de los ejemplos anteriores, pero con el mismo

destino. Si hacemos una copia incremental provocaría un error, al ser la copia completa inicial y la incremental posterior de orígenes diferentes. Por tanto, creamos una nueva completa que cohabita con otras copias de otros orígenes.

En este comando también se plantea una duda. Si estamos excluyendo todo el contenido del directorio personal, ¿cómo es que la copia la realiza de los directorios `Documentos` e `Imágenes`? La respuesta es simple. Cuando en el conjunto de opciones hay solapamiento de directorios o conflicto por incluir y excluir el mismo directorio, las inclusiones y/o exclusiones que aparecen más a la izquierda tienen precedencia sobre las que aparecen a la derecha. Si excluimos todo y antes hemos incluido los dos directorios, estos entran en la copia.

Veamos otro ejemplo, queremos repetir la copia anterior pero en la carpeta `Documentos` se encuentra la carpeta `redes` que pretendemos excluir. Sería así

```
usuario@U2004:~$ duplicity --no-encryption --exclude
~/Documentos/redes --include ~/Documentos --include ~/Imágenes --
exclude '/home/usuario/*' ~ file:///media/usuario/BACKUP/
```

Vemos que la exclusión de `~/Documentos/redes` está antes que la inclusión de `~/Documentos`. Al hacerlo así, la exclusión tiene prioridad sobre la inclusión tener precedencia por la izquierda. Si lo hubiéramos puesto al contrario, el directorio `redes` estaría también incluido en la copia, lo cual queríamos evitar.

### 3.1.2 Restauración de una copia con duplicity

La restauración de `duplicity` también es muy flexible, pero con un funcionamiento particular que debemos entender. En principio no podemos restaurar datos en el mismo origen de la copia si los archivos y directorios que hay en la copia todavía existen ya que los sobrescribiría. Este comportamiento es lógico, ya que si se sobrescribieran datos podríamos estar perdiendo los últimos cambios de los archivos que se hicieron desde la última copia de seguridad. Por ejemplo

```
usuario@U2004:~$ duplicity restore file:///media/usuario/BACKUP/ ~
Restaurar directorio de destino /home/usuario ya existe.
No sobrescribir.
```

Hemos intentado recuperar toda la copia de la carpeta personal que hicimos en ejemplos anteriores y ha dado error por sobreescritura. Podríamos emplear la opción `--force` para forzar la sobreescritura, pero aún así podría fallar.

Generalmente lo que se hace es recuperar lo que se ha perdido. Supongamos que hemos perdido el archivo `logo.jpeg` que se encuentra en la carpeta `Imágenes`. En este caso debemos utilizar la opción `--file-to-restore` para indicar el archivo a recuperar.

```
usuario@U2004:~$ duplicity restore --file-to-restore
Imágenes/logo.jpeg --no-encryption file:///media/usuario/BACKUP/
~/Imágenes/logo.jpeg
Los metadatos en local y remoto están sincronizados, no es necesario
sincronizar.
```



```
Fecha del último respaldo completo: Sun Nov  4 17:02:55 2018
```

Muy importante que el archivo a recuperar se indica con path relativo tal y como aparece cuando hacemos un listado del contenido de la copia de seguridad con el argumento `list-current-files`. Podemos recuperar tanto un archivo como una carpeta. En el siguiente ejemplo recuperamos la carpeta `Imágenes` completa.

```
usuario@U2004:~$ duplicity restore --file-to-restore Imágenes --no-encryption file:///media/usuario/BACKUP/ ~/Imágenes
Los metadatos en local y remoto están sincronizados, no es necesario sincronizar.
Fecha del último respaldo completo: Sun Nov  4 17:02:55 2018
```

Por último decir que podemos recuperar la copia en cualquier lugar diferente a su ubicación original. Si no existe, lo más probable, entonces se restaura sin problemas. Por ejemplo, podemos recuperar toda la copia en una carpeta creada ex profeso.

```
usuario@U2004:~$ duplicity restore --no-encryption file:///media/usuario/BACKUP/ ~/temporal
Los metadatos en local y remoto están sincronizados, no es necesario sincronizar.
Fecha del último respaldo completo: Sun Nov  4 17:04:35 2018
```

## 3.2 rsync

A la hora de realizar copias de seguridad resulta fundamental optimizar el proceso de copia de forma que la operación de copia sea lo más rápida posible. Las copias incrementales permiten copiar solamente los archivos que han cambiado, dejando intactos los que no. Sin embargo, supongamos que un archivo grande, de varios GB ha cambiado en unos pocos bytes. Esto supondría la copia del archivo completo. Es evidente que se consigue una mayor eficiencia si en lugar de copiar el archivo completo se copia solamente aquellos bloques que han cambiado, dejando intacto el resto. Es decir, a nivel de archivos individuales, `rsync` analiza el fichero en origen y en destino, y sólo transmite de forma comprimida los bloques que realmente hayan cambiado. Esta es la mejor característica de `rsync`.

Esta utilidad de copia permite sincronizar archivos y directorios entre dos hosts conectados en red o entre dos ubicaciones en un mismo host, minimizando el volumen de datos transferidos. Además, puede utilizar compresión en la sincronización. `rsync` permite copias remotas pudiendo actuar como un servicio o a través de RSH o SSH.

Esta herramienta de copia tiene diferentes sintaxis en función de si la copia es local o remota, y en este último caso si se emplea una conexión RSH y SSH o un servicio.

### Sintaxis

Para copias locales (en el mismo host)

```
rsync [opciones] origen ... [destino]
```

Para copias remotas usando SSH o RSH.

```
rsync [opciones] [usuario@]host:origen ... [destino]
rsync [opciones] origen ... [usuario@]host:destino
```

Por último, si accedemos remotamente a un servicio `rsync` disponemos de la siguiente sintaxis.

```
rsync [opciones] [usuario@]host::origen ... [destino]
rsync [opciones] rsync://[usuario@]host[:puerto]/origen ...
[destino]
rsync [opciones] origen ... [usuario@]host::destino
rsync [opciones] origen ... rsync://[usuario@]host[:puerto]/destino
```

Vemos que el destino de la copia es opcional, en cuyo caso `rsync` se comporta como el comando `ls -l`, haciendo un listado de los ficheros.

El conjunto de opciones de `rsync` es numeroso. Exponemos aquí los más relevantes. Para más información podemos consultar la página de manual de `rsync`.

### Opciones

`-v, --verbose`

Muestra información de la operación en pantalla.

`-q, --quiet`

Modo silencioso. Suprime los mensajes que no son de error.

`-a, --archive`

Modo archivo. Equivale a incluir las opciones `-rlptgoD` (no `-H,-A,-X`)

`-r, --recursive`

Modo recursivo. Copia también los subdirectorios y su contenido.

`-R, --relative`

Utiliza path relativos.

`-l`

Copia los enlaces simbólicos como tal.

`-b, --backup`

Modo backup. Realiza backups para lo que conviene utilizar también las opciones `--suffix` y `--backup-dir`.

`--backup-dir=DIR`

Realiza backups dentro de una jerarquía basada en `DIR`.

`--suffix=SUFIXO`

Sufijo que se añade al archivo de copia.

`-u, --update`

No copia los archivos que ya están en el destino y son más recientes.

`-p, --perms`

Preserva los permisos actuales del archivo que copia.

`-o, --owner`

Preserva el propietario del archivo.

`-g, --group`

Preserva el grupo del archivo.

`-t, --times`

Preserva las fechas de modificación de los archivos.

`--existing`

Solamente copia el archivo si existe en el destino.

`--ignore-existing`

Solamente copia los archivos nuevos en el destino.

`--delete`

Borra los archivos en el destino que no existen en el origen.

`-z, --compress`

Comprime el archivo durante la transferencia de datos.

`--exclude=PATRÓN`

Excluye los archivos cuyo nombre coinciden con el PATRÓN.

`--exclude-from=FICHERO`

Se especifican varios patrones para excluir los archivos cuyo nombre coinciden con alguno de ellos. Los patrones están dentro de FICHERO.

`--include=PATRÓN`

Incluye los archivos cuyo nombre coinciden con PATRÓN.

`--include-from=FICHERO`

Se especifican varios patrones para incluir los archivos que coinciden con alguno de ellos. Los patrones están dentro de FICHERO.

`--stats`

Muestra alguna estadística sobre la transferencia del fichero.

### `-h, --human-readable`

Los datos sobre tamaño de archivo los da por defecto en bytes. Con esta opción los tamaños aparecen en medidas como KB, MB o GB.

### `--progress`

Muestra el progreso de la transferencia en copias remotas.

### `--list-only`

Lista los archivos en lugar de copiarlos.

### `--version`

Visualiza el número de versión.

Algunas opciones están implícitas y no es necesario indicarlás. Sin embargo, es posible que queramos anularlas. En ese caso tenemos que indicar `--no-OPCIÓN`, siendo `OPCIÓN` la opción que deseamos anular. No todas las opciones admiten el prefijo `no-`, solamente las opciones que son implícitas por otras opciones. Por ejemplo, si queremos utilizar la opción `-a` pero no queremos la opción implícita `-o`, entonces tendríamos que indicar `-a --no-o`.

El orden de las opciones es importante en este caso. Si especificamos `--no-r -a`, la ejecución de `rsync` terminará empleando `-r`, ya que va implícita con `-a`. Al poner antes `--no-r` y después `-a`, hemos hecho implícita `-r`.

## 3.2.1 Sincronizar con `rsync`

En su forma más sencilla de uso, es similar a una copia realizada con `cp`. Si queremos sincronizar un fichero en otro podemos, simplemente ejecutar el siguiente comando.

```
rsync -v fichero1 fichero2
```

Aquí hacemos una copia de `fichero1` en `fichero2`. La opción `-v` es para mostrar información sobre la operación de copia.

En el caso de querer hacer una copia de seguridad de un directorio tendríamos que emplear la opción `-r` para que incluya su contenido. Por ejemplo, supongamos que tenemos montado en `/mnt/backup` un dispositivo de almacenamiento externo o una carpeta compartida por red. Para copiar el directorio `Documentos` en este dispositivo ejecutaríamos el siguiente comando.

```
rsync -vr Documentos /mnt/backup
```

Sin embargo, para propósitos de backup, en la mayoría de los casos se utiliza con la opción `-a`, la cual equivale a utilizar juntos la opción `-r` para que el recorra toda la estructura de directorios que le indiquemos, la opción `-l` para que copie enlaces simbólicos como tales y no el archivo al que apuntan, la opción `-p` para que mantenga los permisos, la opción `-t` para que se mantenga la fecha y hora del fichero, la opción `-g` para que se mantenga el grupo, la opción `-o` para que se mantenga el propietario y la opción `-D` para que se mantengan los ficheros de dispositivo (sólo para root). Ni se mantienen los enlaces

duros (-H) ni las ACLs (-A) por defecto. En definitiva, con la opción `-a` obtenemos una copia exacta de una jerarquía de ficheros y directorios.

```
usuario@U2004:~$ rsync -av Documentos /mnt/backup/
sending incremental file list
Documentos/
Documentos/VSFTPD.docx
Documentos/el Disco duro.pdf
Documentos/httpd-docs-2.0.63.es.chm
Documentos/.mediaartlocal/
Documentos/.mediaartlocal/album-0ca74f9d6ae5d7fcb45f7e61578b1921-
e114e09486f715f89263962c3438554a.jpeg
Documentos/Rafael/
Documentos/Rafael/Documento Writer 1.odt
Documentos/manuales/
Documentos/manuales/Guia Instalación Placa GA-8I945GMF.pdf
Documentos/manuales/Manual GA-8i945gmf_e.pdf
Documentos/manuales/dir-615.pdf
Documentos/manuales/el Disco duro.pdf
Documentos/manuales/refman-5.0-es.a4.pdf
Documentos/redes/
Documentos/redes/garl-2.0.pdf

sent 48,707,092 bytes  received 261 bytes  32,471,568.67 bytes/sec
total size is 48,694,115  speedup is 1.00
```

Supongamos que después de hacer la copia anterior hemos cambiado el archivo `Documentos/Rafael/Documento Writer 1.odt`. Al repetir la copia obtendríamos el siguiente resultado.

```
usuario@U2004:~$ rsync -av Documentos /mnt/backup/
sending incremental file list
Documentos/Rafael/
Documentos/Rafael/Documento Writer 1.odt

sent 22,119 bytes  received 47 bytes  44,332.00 bytes/sec
total size is 48,694,096  speedup is 2,196.79
usuario@U2004:~$ ls -l ./Documentos/Rafael/Documento\ Writer\ 1.odt
-rw-rw-r-- 1 usuario usuario 21424 nov 11 15:50 './Documentos/Rafael/
Documento Writer 1.odt'
```

Hemos listado también el archivo para comprobar su tamaño. Se aprecia claramente que solamente ha copiado el archivo que ha cambiado, pero lo ha hecho completo, al contrario de lo que indica la definición de `rsync` como herramienta para sincronización incremental por bloques de fichero. En este caso ha copiado el archivo completo, en lugar de solamente el bloque que ha cambiado porque estamos ante una copia cuyo destino es un dispositivo local. En esta situación `rsync` considera por defecto que se emplea menos tiempo en copiar el archivo completo que en comprobar que bloques del archivo han cambiado, de ahí que emplee la opción `-W`, `--whole-file`, la cual indica que copie los archivos que han cambiado completos. Si quisiéramos que hiciera la copia solamente de los

bloques que han variado tendríamos que especificar la opción `no-whole-file`.

Repitamos el ejemplo anterior cambiando ligeramente el archivo y haciendo de nuevo la copia pero indicando que copie solo los bloques que han cambiado. Sería así.

```
usuario@U2004:~$ rsync -av --no-whole-file Documentos /mnt/backup/  
sending incremental file list  
Documentos/Rafael/  
Documentos/Rafael/Documento Writer 1.odt  
  
sent 8,235 bytes  received 233 bytes  16,936.00 bytes/sec  
total size is 48,694,121  speedup is 5,750.37
```

Vemos en el resultado del comando que se han enviado al destino muchos menos bytes de lo que ocupa el archivo.

Si además utilizamos la opción `-z`, comprimirá el bloque antes de transferirlos.

```
usuario@U2004:~$ rsync -avz --no-whole-file Documentos /mnt/backup/  
sending incremental file list  
Documentos/Rafael/  
Documentos/Rafael/Documento Writer 1.odt  
  
sent 6,705 bytes  received 233 bytes  13,876.00 bytes/sec  
total size is 48,694,122  speedup is 7,018.47
```

El uso de la opción `-z` puede ser beneficioso o perjudicial, ya que la menor transferencia de datos redunda en un mayor consumo de CPU.

¿En qué se basa `rsync` para decidir que un fichero ha cambiado? Normalmente sólo mira la fecha del fichero y su tamaño, por lo que si ninguna de las dos cosas cambia, por defecto el `rsync` no copiará el fichero. Es muy raro que dos ficheros con la misma fecha y tamaño sean diferentes, pero puede ocurrir. Si en nuestro entorno se puede dar ese caso, tendremos que usar la opción `-c` para que se determine por CRC si realmente los ficheros son iguales. Evidentemente, esto también redundará sensiblemente en el uso de CPU.

### 3.2.2 La barra al final de los nombres de directorio

Respecto a cómo indicar los nombres de los directorios, hay que tener una especial atención respecto a si ponemos una barra al final del nombre del directorio o no, ya que significan cosas diferentes.

Cuando indicamos la barra al final de un directorio origen para copiar, estamos diciendo que debe copiar el contenido del directorio, no el directorio. Esto significa que copiará en destino todos los archivos que hay dentro del directorio origen. Por ejemplo

```
rsync -av ~/Documentos/ /mnt/backup
```

En este caso no copia el directorio `~/Documentos`, sino lo que hay dentro de él. Sin embargo, si omitimos la barra final como en el siguiente ejemplo, si se copia el directorio indicado.

```
rsync -av ~/Documentos /mnt/backup
```

Por tanto, si queremos hacer copia de un directorio completo y que tenga el mismo nombre en el destino tenemos dos opciones. Siguiendo con el ejemplo anterior las indicamos aquí.

```
rsync -av ~/Documentos /mnt/backup
rsync -av ~/Documentos/ /mnt/backup/Documentos
```

No hay problema con indicar barra al final del directorio de destino. En ambos casos se está indicando que el destino de la copia es el directorio en cuestión. Si no existe lo crea, salvo que no exista algún otro directorio en el path. Solamente crea el último indicado en el path si no existe, pero no antecesores de este. En este caso resultaría un error.

### 3.2.3 Eliminando ficheros del destino de backups anteriores

En muchos casos, es posible que hayamos borrados ficheros de origen que ya no queremos que aparezcan en el destino, pero por defecto `rsync` no los elimina. Para que lo haga, debemos usar la opción `--delete` como en el siguiente ejemplo.

Supongamos que hemos borrado el `archivo~/Documentos/el disco duro.pdf`. Si hacemos una copia veremos que el archivo todavía se encuentra en el destino ya que se copió anteriormente.

```
usuario@U2004:~$ rm ./Documentos/el\ Disco\ duro.pdf
usuario@U2004:~$ rsync -av ./Documentos /mnt/backup
sending incremental file list
Documentos/

sent 649 bytes received 24 bytes 1,346.00 bytes/sec
total size is 48,694,122 speedup is 72,353.82
usuario@U2004:~$ ls -l /mnt/backup/Documentos/el\ Disco\ duro.pdf
-rw----- 1 usuario usuario 2598887 dic 22 2013
'/mnt/backup/Documentos/el Disco duro.pdf'
```

Hemos hecho una copia después de borrar el archivo y éste continua de una copia anterior. Ahora realizamos la copia con la opción `-delete`.

```
usuario@U2004:~$ rsync -av --delete ./Documentos /mnt/backup
sending incremental file list
deleting Documentos/el Disco duro.pdf

sent 646 bytes received 53 bytes 1,398.00 bytes/sec
total size is 48,694,122 speedup is 69,662.55
```

Vemos que en el destino ha eliminado el fichero.

Cuando se hagan pruebas con el `--delete` hay que llevar mucho cuidado, porque si elegimos erróneamente el directorio de destino podemos borrar en cascada muchísimos ficheros que no queríamos borrar. Es por eso que se aconseja que se use en las pruebas la opción `-n, --dry-run` para que el comando no haga nada en realidad y así podamos

depurar el comando antes de ponerlo en funcionamiento definitivamente.

### 3.2.4 Impedir actualizar archivos modificados en destino

Otra opción que puede ser interesante en algunos entornos es la `-u`, para que no se sobrescriban los ficheros del destino que son más recientes que los del origen. Esta opción es útil si es posible que en un momento dado se trabaje sobre los ficheros del backup, de modo que en ningún caso reemplacemos los archivos que se puedan haber modificado en destino.

Por ejemplo, supongamos que modificamos el archivo `/mnt/backup/Documentos/Rafael/Documento de writer1.odt`. Al hacer la copia se sobrescribe el destino, perdiendo los últimos cambios. Al hacerlo con `-u` se salta la copia del archivo origen en el destino, preservando éste último, que fue cambiado después de la última copia.

```
usuario@U2004:~$ rsync -av -u ./Documentos /mnt/backup
sending incremental file list
Documentos/Rafael/
Documentos/Rafael/Documento Writer 1.odt

sent 22,145 bytes  received 47 bytes  44,384.00 bytes/sec
total size is 48,694,122  speedup is 2,194.22
usuario@U2004:~$
```

En este caso no se ha copiado el archivo de origen en el destino ya que su fecha de modificación es anterior a la del archivo en el destino. Por tanto, sigue siendo diferente el archivo en origen y su homólogo en destino.

### 3.2.5 Backups incrementales

Si queremos tener un registro de los ficheros que vamos modificando un backup incremental, las opciones `-b`, `--backup`, `--backup-dir=DIR` y `--suffix=SUF` son de mucha utilidad.

Siguiendo con nuestro ejemplo. Tenemos dos directorios `Documentos` en origen y destino perfectamente sincronizados y actualizamos un fichero en el directorio origen. Si volvemos a realizar la sincronización veremos lo siguiente.

```
usuario@U2004:~$ rsync -av ./Documentos /mnt/backup
sending incremental file list
Documentos/Rafael/
Documentos/Rafael/Documento Writer 1.odt

sent 22,146 bytes  received 47 bytes  14,795.33 bytes/sec
total size is 48,694,122  speedup is 2,194.12
```

Ha reemplazado el archivo de destino con el de origen. En la sincronización, borra el archivo en el destino y copia el de origen. Esto implica la pérdida del fichero que teníamos previamente en el destino.



Podemos conservar los archivos que han cambiado en el destino utilizando la opción `--backup`. Con esta opción se crea un directorio en el destino al cual podemos dar nombre con la opción `--backup-dir=DIR` y al que pondremos como nombre la fecha y hora actual. Sería así

```
usuario@U2004:~$ rsync -avv --backup
--backup-dir=/mnt/pendrive/backup_`date +%y%m%d-%H%M` ./Documentos
/mnt/backup
sending incremental file list
(new) backup_dir is /mnt/pendrive/backup_181111-1710
delta-transmission disabled for local transfer or --whole-file
Documentos/VSFTPD.docx is uptodate
Documentos/edd.pdf is uptodate
Documentos/httpd-docs-2.0.63.es.chm is uptodate
Documentos/manuales/Guia Instalación Placa GA-8I945GMF.pdf is
uptodate
Documentos/manuales/Manual GA-8i945gmf_e.pdf is uptodate
Documentos/manuales/dir-615.pdf is uptodate
Documentos/manuales/el Disco duro.pdf is uptodate
Documentos/manuales/refman-5.0-es.a4.pdf is uptodate
Documentos/redes/garl-2.0.pdf is uptodate
Documentos/Rafael/
Documentos/Rafael/Documento Writer 1.odt
backed up Documentos/Rafael/Documento Writer 1.odt to
/mnt/pendrive/backup_181111-1710/Documentos/Rafael/Documento Writer
1.odt
total: matches=0 hash_hits=0 false_alarms=0 data=21458

sent 22,196 bytes  received 928 bytes  46,248.00 bytes/sec
total size is 48,694,130  speedup is 2,105.78
```

Vemos en el resultado de la copia que tenemos un directorio nuevo dentro del destino cuyo nombre es `/mnt/pendrive/backup_181111-1710` donde se ha copiado el archivo que existía antes de la copia en el destino. Supongamos que tenemos montado otro pendrive en `/mnt/pendrive`.

```
usuario@U2004:~$ ls -l
/mnt/pendrive/backup_181111-1710/Documentos/Rafael/Documento\ Writer\
1.odt
-rw-rw-r-- 1 usuario usuario 21457 nov 11 17:06
'/mnt/pendrive/backup_181111-1710/Documentos/Rafael/Documento Writer
1.odt'
```

Este archivo tiene diferente marca de tiempo y tamaño con respecto al nuevo que hay en el destino de la copia, ya que es la versión anterior del mismo.

```
usuario@U2004:~$ la -l /mnt/backup/Documentos/Rafael/Documento\
Writer\ 1.odt
-rw-rw-r-- 1 usuario usuario 21458 nov 11 17:10
'/mnt/backup/Documentos/Rafael/Documento Writer 1.odt'
```

Es importante no usar paths relativos en el parámetro `--backup-dir`, porque son relativos al directorio destino. En los ejemplos anteriores, si hubiéramos indicado `--backup-dir=backup_`date +%y%m%d-%H%M``, el directorio `backup_181111-1710` se hubiera creado dentro del `/mnt/backup` y sería eliminado en la siguiente ejecución `rsync` con `--delete`.

### 3.2.6 Incluyendo y excluyendo ficheros

La mayoría de las veces no queremos hacer backup de un único directorio, sino que de una lista determinada de directorios. En estos casos, o especificamos uno por uno en la línea de comandos o, mucho mejor, los especificamos en una lista. En este caso podríamos crear un script que invoca `rsync` con las opciones deseadas y un fichero con la lista de directorios de los que hay que hacer backup. De esta forma podemos controlar que directorios vamos a copiar modificando este último fichero. Las opciones son:

`--exclude=PATRÓN`

Excluye ficheros que coinciden con PATRÓN.

`--exclude-from=FICHERO`

Lee patrones de ficheros a excluir desde FICHERO.

`--include=PATRÓN`

Incluye ficheros que coinciden con PATRÓN.

`--include-from=FICHERO`

Lee patrones de ficheros a incluir desde FICHERO.

`--files-from=FICHERO`

Lee la lista de ficheros a copiar desde FICHERO.

Como podemos ver en las opciones previas podemos construir patrones de nombre de archivo y/o directorios que deben incluirse/excluirse de una copia. Para construir estos patrones seguiremos las siguientes reglas:

- ✓ Si empieza por `/` será una ruta absoluta. Por ejemplo `/etc/passwd` coincide exactamente con el archivo `/etc/passwd`.
- ✓ Si no empieza por `/` entonces el patrón será cualquier fichero o directorio ubicado en cualquier parte del árbol origen de la copia ya que el algoritmo se aplica recursivamente de arriba a abajo; por tanto solo se aplica al componente final del fichero o directorio. Por ejemplo, si el origen de la copia es la carpeta personal y pusiéramos como patrón `manuales` se aplicaría a cualquier directorio o archivo llamado `manuales` en cualquier lugar del árbol, siempre y cuando `manuales` esté al final del path. Así, se incluirían `~/manuales`, `~/Documentos/manuales`, `~/Documentos/redes/manuales`, ... Incluso un patrón como `subdir/manuales` se aplicaría en cualquier lugar del árbol de almacenamiento donde `manuales` estuviera dentro de un directorio llamado `subdir`. Es decir,

`subdir/manuales` deberían ser los dos componentes finales del path y se incluirían `~/subdir/manuales`, `~/Documentos/subdir/manuales`, `~/Documentos/redes/subdir/manuales`, ...

- ✓ Si el patrón acaba en `/` será un directorio y no un archivo
- ✓ Si el patrón contiene `*` coincide con cualquier cadena que no contenga el carácter barra de dividir `/`.
- ✓ Si el patrón contiene `**` coincide con cualquier cadena incluyendo el carácter barra de dividir `/`.
- ✓ Si el patrón contiene `?` coincide con cualquier carácter excepto la barra de dividir `/`.
- ✓ Podemos usar `[]`, como `[abcd]` o `[a-z]`, para poner una lista de caracteres a coincidir con uno.
- ✓ Si el patrón es `directorio/**` coincide tanto con el directorio (como si hubiéramos puesto `directorio/`) y todo lo que contiene (como si hubiéramos puesto `directorio/**`). Este comportamiento ha sido añadido a partir de la versión 2.6.7

Hemos comentado anteriormente que `rsync` recorre todo el subárbol de almacenamiento desde el directorio origen de la copia y cada nombre (archivo o directorio) visitado durante este recorrido será comprobado con la lista de patrones en el fichero y pueden darse las siguientes situaciones:

- ✓ Si el nombre coincide con el patrón y éste comienza por `-`, entonces el archivo o directorio se excluye.
- ✓ Si el nombre coincide con el patrón y éste comienza por `+`, entonces el archivo o directorio se incluye.
- ✓ Si el nombre no coincide con ningún patrón, entonces el archivo o directorio se incluye.
- ✓ Si un directorio es excluido, entonces el subárbol que cuelga de él se salta y no se examina.
- ✓ Si un directorio se incluye, se continúa recorriendo el subárbol que cuelga de él.

Imaginemos que queremos hacer un backup de los directorios `~/Documentos` y `~/Descargas`. Podríamos usar un comando como:

```
rsync -av --delete --prune-empty-dirs -include-from =  
lista_dirs_backup.txt /mnt/backup/
```

en el que el fichero `lista_dirs_backup.txt` contiene:

```
+ */  
+ /home/usuario/Documentos/**  
+ /home/usuario/Descargas/**
```

```
- *
```

Cada uno de los ficheros del directorio origen se comprueba con los patrones del fichero. El `+` indica que el directorio se incluye en la copia y `-` indica que se excluye. Si el fichero coincide con un patrón se aplica (se incluye o excluye dependiendo de si es `+` o `-`)

Para entender la primera línea debemos antes explicar la última línea. Cuando ponemos `- *` estamos indicando que cualquier otro directorio o fichero que no esté en `/home/usuario/Documentos` y `/home/usuario/Descargas` no se incluya en la copia. El problema está en que la mayoría de las copias se hacen con la opción `-r` (recursividad) y en esta situación `rsync` visita cada subcomponente de un path completo de arriba a abajo. Al visitarlo lo comprueba con los patrones del archivo y al coincidir con `- *` se excluiría, dejando de visitar el resto de los directorios que cuelgan de él. En este ejemplo, al visitar `/home` este no coincide con ningún patrón, excepto `- *`, y se excluye. No se visitaría por tanto `/home/usuario` ni `/home/usuario/Documentos` ni `/home/usuario/Descargas`.

Debido a lo anterior en la primera línea aparece `+ */` para que todos los subcomponentes se recorran. Además, esta opción implicaría una copia de estos directorios que se recorren, aunque no de su contenido, por lo que tendremos que usar la opción `--prune-empty-dirs` para que no copie directorios vacíos.

Una alternativa sin `--prune-empty-dirs` sería incluir específicamente todos los directorios involucrados:

```
+ /home/  
+ /home/usuario/  
+ /home/usuario/Documentos/  
+ /home/usuario/Documentos/**  
+ /home/usuario/Descargas/  
+ /home/usuario/Descargas/**  
- *
```

Los dos asteriscos (`**`) indican que todos los ficheros y directorios que hay debajo pasarán el filtro.

Pero es evidente que en el caso de tener muchos directorios y con mucha profundidad, esta forma se puede complicar mucho y es claramente más sencilla la anterior (aunque también es más lenta por tener que recorrer todos los directorios del origen).

En versiones de `rsync` posteriores a la 2.6.7 también existe la posibilidad de poner tres asteriscos (`***`) para especificar que el propio directorio especificado también pasa el filtro, facilitándonos nuestra labor en el caso de que queramos hacerlo sin `+ */` y sin `--prune-empty-dirs`:

```
+ /home/  
+ /home/usuario/  
+ /home/usuario/Documentos/**  
+ /home/usuario/Descargas/**
```

```
- *
```

Y también es importante tener en cuenta si los directorios especificados son relativos o absolutos. En los ejemplos anteriores, el directorio origen era el raíz (/) y por ello, los directorios se guardaban con todo el path (`/home/usuario/Documentos` y `/home/usuario/Descargas`) y así los especificábamos en el fichero de patrones. Sin embargo, también podríamos haber decidido hacerlo con referencia a `/home/usuario` para lo cual solamente tenemos que indicar patrones sin la barra inicial.

```
rsync -av --delete --prune-empty-dirs -include-from =  
lista_dirs_backup.txt /home/usuario/ /mnt/backup/
```

Con lo que en el fichero `lista_dirs_backup.txt` tendríamos que poner lo siguiente:

```
+ Documentos/**  
+ Descargas/**  
- *
```

Y el resultado sería un backup sin la parte del directorio `/home/usuario/`.

### 3.2.7 Copias remotas

Hasta ahora hemos hecho todos los ejemplos en local. Sin embargo, la máxima utilidad de `rsync` llega cuando se usa para hacer backups en una máquina remota, de forma que el backup cumpla mejor su función al estar físicamente en otro sistema.

En la máquina destino es posible usar el propio proceso `rsync` funcionando como demonio y escuchando por defecto en el puerto 873 para recibir estas conexiones, pero es mucho más cómodo y fácil hacerlo por SSH, algo para lo que `rsync` ya está preparado por defecto.

Para esto es conveniente configurar el cliente y el servidor de SSH involucrados para entrar de forma transparente usando autenticación por clave pública para evitar tener que introducir la contraseña cada vez, aunque no es estrictamente necesario. Una vez que lo tengamos así y verifiquemos que podemos entrar en la otra máquina sin introducir usuario ni contraseña, podemos usar `rsync` exactamente igual que si trabajáramos con la máquina local, solo que tenemos que especificar el prefijo `usuario@maquina:` en el origen o en el destino, nunca en ambos.

## 4 Cuotas de disco

Uno de los problemas más habituales que puede tener un administrador de sistemas cuando debe administrar un número considerable de usuarios es el uso racional de los dispositivos de almacenamiento. Si no se ponen límites, cada usuario tenderá a almacenar una cantidad considerable de información. La consecuencia inmediata es que se desborda la previsión de necesidades para la red, sobre todo cuando establecemos un almacenamiento de datos centralizado.

Por otra parte, un usuario malintencionado podría impedir el uso normal del sistema informático simplemente saturando la capacidad de su almacenamiento. En este contexto, cualquier administrador tendrá dos inquietudes: poder consultar el espacio ocupado por los usuarios de forma individual y poder establecer límites a la capacidad máxima de almacenamiento que puedan utilizar los usuarios.

En Linux, el administrador puede establecer límites de forma individual a cada usuario o puede hacerlo para todo un grupo. Además, existen dos enfoques diferentes en el momento de establecer cuotas:

- ✓ Limitar el número de bloques de disco, con lo que se restringe el tamaño máximo que se puede ocupar.
- ✓ Limitar el número de i-nodos, que restringe el número máximo de archivos que pueden crearse.

Además, cuando un administrador establece cuotas, puede fijar dos tipos de límites:

- ✓ Rígido (hard): El sistema operativo impedirá que el límite sea sobrepasado.
- ✓ Flexible (soft): El sistema operativo avisará cuando el límite sea sobrepasado.

Para poder gestionar las cuotas de disco para los usuarios debemos instalar los siguientes paquetes

```
apt-get install quota quotatool
```

Una vez instalados los paquetes podemos comenzar a configurar y establecer las cuotas de disco para los usuarios.

## 4.1 Activar las cuotas en el sistema de archivos

Para que el sistema pueda gestionar los valores de cuota de un dispositivo de almacenamiento, éste debe montarse teniendo en cuenta dicha característica. Como esta gestión debe hacerse desde el arranque del sistema, tendremos que editar el archivo `/etc/fstab` para incorporar esta características en el sistema de archivos de aquellas particiones a las que queramos controlar el espacio de almacenamiento.

En el ejemplo inferior vamos a activar las cuotas de disco para la partición que almacena las carpetas personales de los usuarios, la cual se monta en `/home`. Para ello tenemos que añadir `usrquota` y `grpquota` a las opciones de montaje de la partición montada en `/home`.

```
# /home was on /dev/sda2 during installation
UUID=32943e55-8a8e-4756-8626-fdab7e4e7176 /home ext4
defaults,usrquota,grpquota 0 2
```

Para que el sistema de cuotas esté activo, sólo nos quedará montar de nuevo la partición sobre la que se aplicarán las cuotas. Esto podemos conseguirlo reiniciando el sistema o ejecutando el siguiente comando:

```
sudo mount -o remount,rw /home
```

## 4.2 Crear los archivos de cuota y la tabla de uso de espacio compartido

El siguiente paso consistirá en preparar al sistema de archivos para que admita las cuotas. Para lograrlo, utilizaremos el comando `quotacheck`, que se encargará de examinar el sistema de archivos que le indiquemos y crear una tabla del uso actual del disco. Después, la compara con la información almacenada en los archivos de cuota de disco del sistema de archivos, actualizándolos cuando sea necesario. En nuestro caso, como es la primera vez que usamos las cuotas, los archivos de cuota no existen en el sistema de archivos. Por este motivo, usaremos la opción `-c`, que permite crearlos de forma automática, salvo si reiniciamos el sistema en el punto anterior.

```
sudo quotacheck -cmgu /home
```

Los argumentos `-g` y `-u` hacen que se comprueben tanto los grupos como los usuarios. La opción `-m` evita que se vuelva a montar el sistema de archivos como solo lectura. Después de ejecutar el comando veremos que se han creado los archivos `aquota.user` y `aquota.group` en `/home`.

Si se tuvieran más sistemas de archivos con soporte para cuotas en la raíz de cada uno estarían estos archivos, o solo uno dependiendo lo que se pidió, usuarios, grupos o ambos.

Una vez los archivos con información del uso de disco están generados debemos activar las cuotas con el siguiente comando

```
sudo quotaon -v /home
```

## 4.3 Configurar cuotas para usuarios y grupos

Para asignar cuotas a los usuarios o grupos usaremos el comando `edquota`. Por ejemplo, para asignar valores de cuota a la cuenta `usuario`, escribiríamos lo siguiente:

```
sudo edquota -u usuario
```

Para asignar valores a todo un grupo, bastaría con escribir así el comando anterior:

```
sudo edquota -g grupo
```

Incluso, en lugar del nombre de un usuario o un grupo, podemos incluir sus UID o PID correspondientes. En realidad, el comando `edquota` abre el editor nano con la información correspondiente a la cuota del usuario o grupo que hayamos indicado.

```
GNU nano 2.9.3 /tmp//EdP.a8CD5Wz
Quotas de disco para user usuario (uid 1000):
Sist. arch.      bloques      blando      duro      inodos      blando      duro
/dev/sda2        1149712      0           0          4184        0           0
```

Figura 12.- Cuota de usuario

Cada línea del archivo se refiere a la cuota del usuario en un sistema de archivos (partición) concreto. Cada línea está dividida en campos que son los siguientes:

- ✓ Sist. arch.: La partición para la que tenemos activadas las cuotas. En nuestro ejemplo es `/dev/sda2`, pero en otro equipo puede ser distinta.
- ✓ Bloques → Indica el número de bloques que está usando el usuario en estos momentos. El tamaño de bloque es de 1 KB.
- ✓ Blando → Indica un valor flexible en cuanto al número de bloques utilizados. Este límite podrá sobrepasarse durante un determinado tiempo.
- ✓ Duro → Representa el límite estricto de bloques que podrá usar el usuario (o grupo). Cuando el espacio ocupado por el usuario alcance el valor indicado en esta columna, ya no podrá guardar más información.
- ✓ Inodos → Indica el número de archivos que está usando el usuario en estos momentos.
- ✓ Blando → Indica un valor flexible en cuanto al número de archivos y/o directorios utilizados. Como en el caso de los bloques, el límite podrá sobrepasarse durante un tiempo.
- ✓ Duro → Representa el límite rígido de i-nodos que podrá usar el usuario (o grupo). Cuando el usuario alcance el número de archivos y/o directorios indicado en esta columna, ya no podrá crear archivos y/o directorios nuevos.

Para eliminar cualquier límite, basta con poner su valor a cero. La asignación de cuotas a grupos funciona de forma idéntica a la de los usuarios.

Otra forma de establecer la cuota para un usuario o grupo es emplear el comando `setquota`.

### Sintaxis

```
setquota { -u usuario | -g grupo } lbb lrb lbi lri { -a | filesystem ... }
```

### Opciones

`-u usuario`

Usuario al que se establece la cuota

`-g grupo`

Grupo al que se establece la cuota.

`-a`

Se establece la cuota para todos los sistemas de archivos con cuota activada

### Parámetros

`lbb`

Límite blando de bloques



`lrb`

Límite rígido de bloques

`lbi`

Límite blando de ficheros.

`lri`

Límite rígido de ficheros.

`filesystem ...`

Sistema de ficheros para los que se establece la cuota. Si son varios se separan por un espacio.

## 4.4 Establecer un valor para el periodo de gracia

Como hemos comentado antes, el periodo de gracia representa el tiempo durante el cual podrá sobrepasarse un límite flexible antes de que se convierta en rígido. El tiempo de gracia puede expresarse en días, horas, minutos o segundos. Para modificar el valor predeterminado del periodo de gracia sólo debemos ejecutar el siguiente comando:

```
sudo edquota -t

Período de gracia antes de imponer límites blandos para users:
La unidad de tiempo puede ser: días, horas, minutos, o segundos
Sist. arch.      Periodo gracia bloque  Periodo gracia inodo
/dev/sda2        7días                7días
```

Figura 13.- Establecer periodo de gracia

Vemos en la imagen anterior que el periodo de gracia por defecto es de 7 días.

El comando `edquota` es capaz de cambiar las unidades de tiempo en inglés, por su traducción al español, para mostrarnos el archivo. Sin embargo, luego no es capaz de hacer la conversión contraria. Esto significa que, cuando escribamos los valores para los periodos de gracia, deberemos utilizar las unidades de tiempo en inglés (days, hours, minutes o seconds).

## 4.5 Comprobar las cuotas de disco

Cuando necesitemos comprobar la cuota de un determinado usuario, empleamos el comando `edquota` y volver a editar el fichero de definición de cuotas de usuario. Sin embargo, disponemos de otras opciones que nos evitan el riesgo de modificarlo por accidente.

El comando `quota` ofrece la información relativa a las cuotas de un usuario o grupo determinado. La sintaxis sería como en el siguiente ejemplo:

```
usuario@U1804:~$ sudo quota -u usuario
Cuotas de disco para user usuario (uid 1000):
Sistema de archivos bloques  cuota límite graciaarchivos  cuota límite gracia
/dev/sda2 1149712 7864320 0 4184 0 0
```

Figura 14.- Informe de cuota para un usuario

Como se puede suponer, para consultar la cuota de un grupo, la sintaxis sería así:

```
sudo quota -g grupo
```

También podemos omitir los argumentos `-u` o `-g`. Si lo hacemos, el comando supondrá que estamos consultando la cuota de un usuario.

El comando `repquota` muestra un informe sobre el grado de ocupación y las cuotas en una determinada partición. Para el ejemplo que nos ocupa sería así:

```
usuario@U1804:~$ sudo repquota /home
*** Informe para user quotas en dispositivo /dev/sda2
Periodo de gracia de bloque: 7días; periodo de gracia de inodo: 7días
límites de Bloque Límites de archivo
Usuario usado blando duro gracia usado blando duro gracia
-----
root -- 32 0 0 5 0 0
usuario -- 1149712 7864320 0 4184 0 0
```

Figura 15.- Informe de cuota para un sistema de archivos

Un detalle que pasa casi desapercibido en el informe anterior son los dos guiones (`--`) que hay después del nombre de usuario. El primero indica que aún no se ha excedido el límite flexible para los bloques y el segundo que no se ha excedido el límite flexible en los i-nodos. Cuando se exceda cualquiera de ellos, su guión se convertirá en un signo más (+). Además, también aparecerá el periodo de gracia que aún tendrá el usuario para seguir excediendo el límite. Este es un mecanismo muy sencillo para identificar a los usuarios que han excedido alguno de sus límites, cuando el número de usuarios es elevado.

Por último, cuando un usuario trate de sobrepasar su límite rígido, recibirá un error y la operación no podrá completarse.

## 4.6 Desactivar y volver a activar las cuotas.

En ocasiones, podemos necesitar desactivar las cuotas de forma transitoria, por ejemplo para realizar alguna operación administrativa, y volver a activarlas más tarde. Desactivarlas será tan sencillo como escribir el siguiente comando:

```
sudo quotaoff /home
```

Para volverlas a activar, bastará con utilizar el comando `quotaon`:

```
sudo quotaon /home
```

## 4.7 Avisos de cuotas excedidas

Cuando un usuario llega al límite suave o soft al crear o modificar un documento, le aparecerán avisos como el siguiente:

```
usuario@U2004:~$> ls -l > directorio.txt
sda2: warning, user block quota exceeded.
```

En este instante como el usuario `usuario` no ha llegado al límite rígido ni ha expirado el tiempo de gracia, el sistema permite crear el archivo pero se le notifica con un aviso.

Pero si lo que deseamos es notificar inmediatamente y vía correo electrónico que un usuario llegó a su límite, por ejemplo, tenemos funcionando un servidor de correo electrónico cuyos buzones están en el volumen con las cuotas activadas y un usuario que recibe muchos mensajes spam está saturando su cuenta, puede ser notificado que su cuota está llegando al límite.

Para lo anterior usaremos el comando `warnquota`. Este comando simplemente se invoca desde la línea de comandos, sin argumentos, revisará los sistemas de archivos con cuotas activadas y revisará todos los usuarios buscando quien ha excedido el límite blando tanto por bloques como por inodos, y a aquellos que lo hayan excedido les enviará un correo notificándoselo.

Se puede agregar en la tabla cron una línea como la siguiente para que `warnquota` haga su trabajo cada 12 horas:

```
0 0,12 * * * root /usr/sbin/warnquota
```

`warnquota` viene con los mensajes en inglés por defecto, el archivo de configuración es `/etc/warnquota.conf`, es muy intuitivo y fácil de cambiar para personalizarlo con los mensajes a español para que sea más fácil entender a los usuarios que han excedido sus cuotas.

## 5 LVM

Un volumen lógico crea una capa de abstracción sobre el almacenamiento físico. Esto provee una flexibilidad mucho mayor que el uso directo del almacenamiento físico tradicional con particiones. Con un volumen lógico no estamos restringidos a los tamaños de los discos físicos. Además, la configuración del hardware de almacenamiento se oculta por software para poder redimensionarlo y moverlo sin parar las aplicaciones ni desmontar los sistemas de archivo. Esto reduce los costes operacionales.

Las ventajas de utilizar volúmenes lógicos son:

- ✓ Capacidad flexible.- Cuando usamos volúmenes lógicos el sistema de archivos se puede extender a través de múltiples discos ya que podemos agregar discos y particiones en un volumen lógico simple.
- ✓ Redimensionamiento flexible.- Podemos extender o reducir volúmenes lógicos con la ejecución de un comando si necesidad de volver a crear el sistema de archivos o

reparticionar los dispositivos de disco subyacentes.

- ✓ Recolocación de datos en línea.- Para desplegar nuevos y más rápidos subsistemas de almacenamiento podemos mover datos con el sistema en activo. Los datos son redistribuidos en los discos mientras están en uso.
- ✓ Nombres de dispositivos.- Los volúmenes lógicos se identifican por un nombre y podemos renombrarlos a conveniencia. Además, son gestionados en grupos.
- ✓ Volúmenes RAID.- Podemos crear volúmenes RAID.
- ✓ Instantáneas.- Podemos crear instantáneas de volúmenes para guardar copias de los mismos o probar los efectos de cambios realizados sin afectar a los datos.

Para gestionar los volúmenes lógicos en Linux disponemos de LVM (*Logical Volume Management*) el cual es un sistema de gestión de volúmenes lógicos avanzado y mucho más flexible que los tradicionales métodos de particiones de disco. En LVM se manejan tres conceptos básicos:

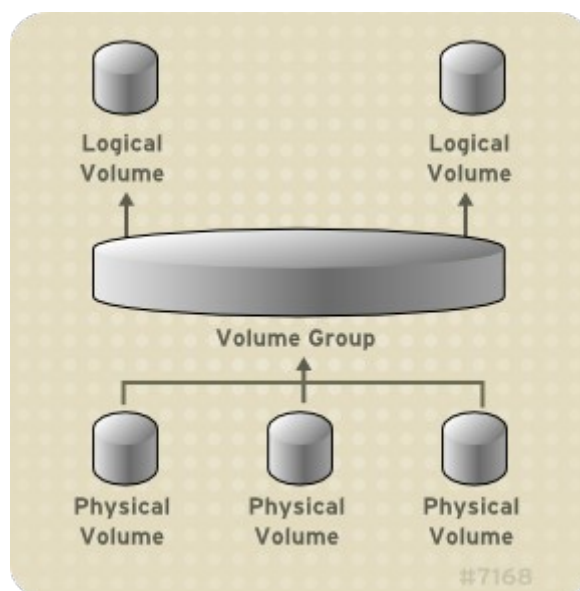


Figura 16.- Arquitectura LVM

- ✓ Volumen físico.- Se corresponde con el disco físico. Simplemente suministra el espacio en disco para los volúmenes lógicos.
- ✓ Grupo de volúmenes.- Los volúmenes físicos se agrupan en grupos de volúmenes. Es una capa intermedia que relaciona un volumen físico con uno lógico.
- ✓ Extensión.- Una extensión es la cantidad de espacio mínima que se asigna a un volumen lógico y éste tendrá un tamaño múltiplo de la extensión ya que las extensiones se asignan completas a un único volumen lógico. Dentro de un grupo de volúmenes el espacio en disco disponible desde los volúmenes físicos se divide en unidades de tamaño fijo denominadas *extensiones* las cuales se asignan a los volúmenes lógicos. En un volumen físico, las extensiones se conocen como extensiones físicas y en un volumen lógico, las extensiones se conocen como

extensiones lógicas las cuales tienen el mismo tamaño que una extensión física.

- ✓ Volúmenes lógicos.- El espacio en disco disponible en un grupo de volúmenes se divide en volúmenes lógicos. A diferencia de las particiones, los volúmenes lógicos tienen un nombre en lugar de un número y su espacio puede estar distribuido en varios discos además de que no tienen que estar físicamente contiguo. Existen tres tipos de volúmenes lógicos:
  - Lineales.- Se asigna espacio físico disponible lineal en orden. Es decir, los volúmenes lineales se crean uno detrás de otro en los volúmenes físicos.
  - Distribuidos.- El volumen lógico ocupa varios volúmenes físicos y los datos se distribuyen a lo largo de todos ellos, incrementando la tasa de transferencia en lectura y escritura. Equivale a un sistema RAID 0.
  - Reflejados.- El volumen lógico replica los datos en varios volúmenes físicos para obtener tolerancia a fallos. Si un volumen físico falla, el sistema continúa accediendo al otro ya que son copias exactas. Equivale a RAID 1

Para poder gestionar LVM debemos instalar el siguiente paquete.

```
sudo apt-get install lvm2
```

A partir de ahora vamos a ver como gestionar completamente un espacio de almacenamiento LVM.

## 5.1 Crear los volúmenes físicos

Los discos empleados para crear el grupo de volúmenes no pueden tener particiones definidas. Por tanto lo primero que tenemos que hacer antes de marcarlos como volúmenes físicos LVM es borrar las particiones que pudiera haber. Para ello emplearemos cualquier aplicación de gestión de particiones para borrarlas todas o, crear una nueva vacía.

También podemos eliminar la tabla de particiones del disco. Esto se consigue con el comando `dd` siguiente. Por ejemplo, supongamos que el disco `/dev/sdb` tuviera una tabla de particiones y queremos añadirlo como volumen físico en LVM. Ejecutaríamos el siguiente comando

```
dd if=/dev/zero of=/dev/sdb bs=512 count=1
```

Crear un volumen físico consiste en marcar un dispositivo de disco como volumen físico en LVM. Es el primer paso para la gestión del almacenamiento con LVM. En nuestro ejemplo disponemos de tres discos: `/dev/sdb`, `/dev/sdc` y `/dev/sdd` de 20, 20 y 25 GB respectivamente. Para marcar estos discos como volúmenes físicos en LVM ejecutamos el siguiente comando

```
sudo pvcreate /dev/sdb /dev/sdc /dev/sdd
Physical volume "/dev/sdb" successfully created.
Physical volume "/dev/sdc" successfully created.
Physical volume "/dev/sdd" successfully created.
```

La ejecución del comando anterior debería escribir una cabecera LVM en todos los dispositivos indicados para marcarlos como volúmenes físicos LVM.

También podemos marcar como volumen físico a particiones sueltas. En este caso el argumento para el comando `pvccreate` sería un archivo de dispositivo que representara a una partición en lugar de a un disco completo.

## 5.2 Crear el grupo de volúmenes

Una vez disponemos de los volúmenes físicos podemos crear un grupo de volúmenes desde los volúmenes físicos anteriores. Para ello utilizamos el comando `vgcreate`. Tendremos que indicar un nombre de grupo de volúmenes seguido por al menos un volumen físico. Por ejemplo,

```
sudo vgcreate grupo_lvm /dev/sdb
```

Este ejemplo creará un grupo de volúmenes con solo un volumen físico. Podemos indicar más de un volumen físico.

```
sudo vgcreate grupo_lvm /dev/sdb /dev/sdc /dev/sdd
```

Por defecto el tamaño de la extensión de un grupo de volúmenes es 4 MB, el cual es el adecuado para la mayoría de los casos. Sin embargo, podemos indicar un tamaño diferente al crear el grupo de volúmenes con la opción `-s`. En el siguiente ejemplo creamos el grupo de volúmenes con un tamaño de extensión de 8 MB.

```
sudo vgcreate -s 8M grupo_lvm /dev/sdb
```

Normalmente solo necesitamos un único grupo de volúmenes por servidor. Todo el almacenamiento se asigna en el grupo de volúmenes y ahí se colocan todos los volúmenes lógicos.

## 5.3 Expandir el grupo de volúmenes

Cuando tenemos necesidad de aumentar el espacio de almacenamiento de un grupo de volúmenes podemos hacerlo añadiendo más volúmenes físicos, sin necesidad de parar o volver a crear el sistema de archivos.

Para expandir un grupo de volúmenes añadiendo volúmenes físicos adicionales tenemos el comando `vgextend`. Este comando toma un grupo de volúmenes seguido de los volúmenes físicos a añadir. Si necesitamos añadir más de uno lo indicamos separados por espacio. Por ejemplo, si creamos el grupo de volúmenes de ejemplo anterior con solo un volumen físico podemos añadir los otros dos ejecutando el siguiente comando

```
sudo vgextend grupo_lvm /dev/sdc /dev/sdd
```

Hemos añadido al grupo de volúmenes dos volúmenes físicos al inicial que teníamos cuando lo creamos, expandiendo la capacidad disponible de almacenamiento.

## 5.4 Crear volumen lógico lineal

Una vez tenemos espacio de almacenamiento disponible en el grupo de volúmenes podemos crear volúmenes lógicos. Para crear un volumen lógico en un grupo de almacenamiento utilizamos el comando `lvcreate`. Con este comando indicamos el grupo de volumen al que se añade el nuevo volumen lógico. Además, para especificar el tamaño del volumen utilizamos la opción `-L` y con la opción `-n` especificamos el nombre del nuevo volumen.

Por ejemplo, para crear un volumen lógico de 10GB de nombre `DATOS` al grupo de volúmenes `grupo_lvm` creado anteriormente ejecutamos el siguiente comando

```
sudo lvcreate -L 10G -n DATOS grupo_lvm
```

Damos por hecho que el grupo de volúmenes tiene suficiente espacio sin asignar para soportar la capacidad del nuevo volumen lógico.

Si lo que necesitamos es crear un volumen lógico utilizando la totalidad del espacio sin asignar en el grupo de volúmenes podemos utilizar el comando `lvcreate` con la opción `-l` y el valor especial `100%FREE` en lugar de indicar un tamaño. Este valor especial indica al comando que utilice el espacio sin asignar restante dentro del grupo de volúmenes para asignárselo al nuevo volumen lógico. Por ejemplo, para añadir un nuevo volumen lógico de nombre `BACKUP` con el resto del espacio libre ejecutamos el siguiente comando

```
sudo lvcreate -l 100%FREE -n BACKUP grupo_lvm
```

Para indicar el tamaño del volumen lógico disponemos de dos opciones. Con `-L` indicamos el tamaño en medidas de capacidad tradicional como MB o GB. En este caso tendremos que indicar un número acompañado de la letra que indica la unidad empleada; M para MB y G para GB. Para otras unidades consultar la página de manual del comando.

Con la opción `-l` indicamos el tamaño en número de extensiones. El tamaño de la extensión se estableció al crear el grupo de volúmenes y por defecto es 4 MB. Además, con esta opción también podemos utilizar los valores especiales como `100%FREE` para indicar que utilice todo el espacio disponible del grupo de volúmenes o `N%grupo_lvm` para indicar un porcentaje de espacio disponible en el grupo de volúmenes.

Podemos indicar el volumen físico del cual tomará el espacio de almacenamiento que asignará al volumen lógico. Por ejemplo, el siguiente volumen lógico tendrá 10 GB de espacio que se tomará del segundo disco duro.

```
sudo lvcreate -L 10G -n DATOS grupo_lvm /dev/sdb
```

Aunque estemos creando un volumen lógico lineal, podemos indicar que reparta el espacio de almacenamiento del volumen entre varios volúmenes físicos, indicando en cada caso cuantas extensiones tomará de cada uno. En el siguiente ejemplo se indica que el volumen lógico ocupará 512 extensiones ( $512 \times 4 \text{ MB} = 2048 \text{ MB} = 2 \text{ GB}$ ) de las cuales 64 se tomarán del segundo disco duro del sistema (desde la extensión 1024 a la 1088) y el resto del tercer disco duro (desde la extensión 128 a 576).

```
sudo lvcreate -l 512 -n DATOS grupo_lvm /dev/sdb:1024-1088  
/dev/sdc:128-576
```

Debemos tener cuidado al especificar las extensiones que toma en cada volumen físico ya que si estas están ocupadas por otros volúmenes lógicos el comando producirá un error y no lo creará.

### 5.4.1 Opciones adicionales en la creación de volúmenes lógicos

Antes hemos visto la creación de volúmenes lógicos. Hay una serie de opciones avanzadas para crearlos. Estas son:

- ✓ **--type** → Especifica el tipo de volumen lógico, lo cual determina como se coloca en el espacio de almacenamiento. Algunos de los tipos no estarán disponibles si no hay suficientes volúmenes físicos subyacentes para crear correctamente la topografía elegida. Los tipos más comunes son:
  - **linear** → El tipo por defecto. El dispositivo físico subyacente utilizado (si hay más de uno) será añadido a cada uno, uno detrás de otro.
  - **striped** → Similar a RAID 0, la topología distribuida divide los datos en los volúmenes físicos subyacentes. Se mejora el rendimiento al aumentar la tasa de transferencia al disco en lectura y escritura, pero los datos son vulnerables. Requiere la opción **-i** descrita más abajo y un mínimo de dos volúmenes físicos.
  - **raid1** → Crea un volumen en espejo RAID 1. Por defecto el espejo tiene dos copias, pero se pueden especificar más con la opción **-m**. Requiere un mínimo de dos volúmenes físicos.
  - **raid5** → Crea un volumen RAID 5. Requiere al menos tres volúmenes físicos.
- ✓ **-m** → Especifica el número adicional de copias en un volumen de tipo RAID 1. Un valor de 1 indica que se mantiene una copia adicional, para un total de dos conjuntos de datos.
- ✓ **-i** → Especifica el número de volúmenes distribuidos. Solo para el tipo de volumen distribuido.
- ✓ **-s** → Indica la creación de una instantánea desde un volumen lógico existente en lugar de un volumen lógico independiente.

Más adelante veremos el uso de estas opciones cuando veamos los volúmenes RAID.

## 5.5 Crear sistema de archivos y montar volúmenes lógicos

Cuando hemos creado el volumen lógico tenemos que crearle un sistema de archivos y montarlo en una carpeta para poder acceder al mismo. En este caso utilizamos los conocidos comandos **mkfs** y **mount** vistos anteriormente. La diferencia está en el nombre del dispositivo que corresponde al volumen lógico. Al igual que las particiones tradicionales los archivos de dispositivo de los volúmenes lógicos están en **/dev**. Aquí habrá un



directorio para el grupo de volúmenes y el archivo de dispositivo con el mismo nombre que el volumen lógico. Por ejemplo, si hemos creado anteriormente el volumen lógico `DATOS` en el grupo de volúmenes `grupo_lvm`, entonces su archivo de dispositivo es `/dev/grupo_lvm/DATOS1`.

Por tanto, si queremos crearle un sistema de archivos `ext4` al volumen anterior ejecutaríamos el siguiente comando

```
sudo mkfs -t ext4 /dev/grupo_lvm/DATOS1
```

Una vez el volumen ha sido formateado podemos montarlo. Por ejemplo, en este caso lo vamos a montar en la carpeta `/mnt/datos1`.

```
sudo mkdir /mnt/datos1
sudo mount /dev/grupo_lvm/DATOS1 /mnt/datos1
```

## 5.6 Redimensionar un volumen lógico

Los volúmenes lógicos pueden cambiar de tamaño cuando se necesite. Una de las principales ventajas de LVM es la flexibilidad ya que podemos ajustar el tamaño de los volúmenes sin necesidad de parar el sistema.

Dependiendo de si lo que queremos es aumentar o disminuirlo deberemos utilizar un proceso u otro.

### 5.6.1 Aumentar el tamaño de un volumen lógico

Para aumentar el tamaño de un volumen lógico existente disponemos del comando `lvresize`. Con la opción `-L` indicamos el nuevo tamaño. También podemos utilizar un aumento del tamaño relativo anteponiendo el signo `+` al valor de la opción.

Después de aumentar el tamaño hay que redimensionar también el sistema de archivos en el caso de que el volumen lógico tuviera uno. Para ello disponemos de la opción `--resizefs`.

Para indicar el nombre del volumen lógico que queremos expandir necesitamos especificar el grupo de volúmenes seguido de una barra de dividir y el nombre del volumen lógico.

Por ejemplo, vamos a aumentar el espacio del primer volumen lógico que tenía 2 GB a 5GB para lo que tendremos que ejecutar el siguiente comando

```
sudo lvresize -L +3G --resizefs grupo_lvm/DATOS1
```

También podemos expandir el sistema de archivos manualmente. En este caso omitimos la opción `--resizefs` y utilizamos el comando en línea para expandir el sistema de archivos. Por ejemplo si el sistema de archivos es `ext4` entonces ejecutaríamos estos comandos.

```
sudo lvresize -L +3G grupo_lvm/DATOS1
```

```
sudo resize2fs /dev/grupo_lvm/DATOS1
```

### 5.6.2 Reducir el tamaño de un volumen lógico

Para reducir el tamaño de un volumen lógico utilizamos el comando `lvreduce`. Si el volumen lógico contiene un sistema de archivos hay que reducir primero el tamaño del sistema de archivos para que el volumen lógico tenga el mismo tamaño.

La reducción de la capacidad de un volumen lógico puede provocar pérdida de datos, por tanto el procedimiento para reducir el tamaño es más laborioso. Vamos a ver un ejemplo reduciendo el volumen lógico DATOS1 que hemos aumentado previamente. Queremos reducirlo en 1 GB.

Primero deberíamos hacer una copia de seguridad de los datos. Un error durante la reducción podría hacernos perder estos datos. Una vez hecha la copia de seguridad debemos comprobar cuanto espacio está actualmente en uso en el volumen lógico. Para ello contamos con el comando `df`.

```
usuario@U2004:~$ df -h /mnt/datos1
S.ficheros                Tamaño Usados  Disp Uso% Montado en
/dev/mapper/grupo_lvm-DATOS1  4,9G   217M   4,5G   5% /mnt/datos1
```

En el ejemplo anterior hemos comprobado el espacio libre y ocupado del volumen DATOS1 que hemos creado anteriormente. Tiene 217 MB en uso y disponibles 4,5 GB. Con esta información podemos estimar cual será el tamaño máximo que podemos reducir. Evidentemente podemos reducir 1GB

Ahora debemos desmontar el sistema de archivos. A diferencia del aumento de la capacidad, reducir la capacidad de un sistema de archivos debería hacerse desmontado.

```
sudo umount /mnt/datos1
```

Después del desmontaje, comprobamos el sistema de archivos para asegurar que se encuentra en buen estado. Usaremos la opción `-f` del comando `fsck` para verificar que todo está bien.

```
sudo fsck -t ext4 -f /dev/grupo_lvm/DATOS1
```

Una vez comprobado el sistema de archivos podemos reducir su tamaño utilizando el comando adecuado al tipo de sistema de archivos. El anterior es `ext4`, por tanto emplearemos `resize2fs` indicándole el tamaño final del sistema de archivos. En este punto la opción más segura es elegir un tamaño final que sea sensiblemente mayor que el uso actual del volumen.

```
sudo resize2fs -p /dev/grupo_lvm/DATOS1 4G
```

Una vez se ha completado la operación podemos reducir el volumen lógico indicando el mismo tamaño anterior con la opción `-L`.

```
sudo lvresize -L 4G grupo_lvm/DATOS1
```

Nos dará un aviso acerca de la posibilidad de perder datos. Si hemos seguido el procedimiento tecleamos y para confirmar. Podemos volver a comprobar el sistema de archivos de nuevo.

```
sudo fsck -t ext4 -f /dev/grupo_lvm/DATOS1
```

Si todo ha ido bien, podemos volver a montar el sistema de archivos.

```
sudo mount /dev/grupo_lvm/DATOS1 /mnt/datos1
```

## 5.7 Visualizar información del sistema LVM

Disponemos de varios comandos para ver la información sobre LVM en nuestro sistema.

### 5.7.1 Visualizar información sobre volúmenes físicos

Hay tres comandos para visualizar propiedades de los volúmenes físicos: `pvs`, `pvdisplay` y `pvscan`.

El comando `pvs` suministra información de los volúmenes físicos en una forma configurable y con una línea por volumen físico. Por ejemplo

```
usuario@U2004:~$ sudo pvs
PV          VG          Fmt  Attr PSize  PFree
/dev/sdb    grupo_lvm  lvm2  a--  <20,00g  10,75g
/dev/sdc    grupo_lvm  lvm2  a--  <20,00g  18,24g
/dev/sdd    grupo_lvm  lvm2  a--  <25,00g  <25,00g
```

El comando `pvdisplay` ofrece una salida más detallada por cada volumen físico. Visualiza las propiedades (tamaño, extensiones, grupo de volúmenes, etc.) en un formato fijo. El siguiente ejemplo muestra la salida para el volumen físico correspondiente al disco `/dev/sdb`.

```
usuario@U2004:~$ sudo pvdisplay
--- Physical volume ---
PV Name                /dev/sdb
VG Name                grupo_lvm
PV Size                20,00 GiB / not usable 4,00 MiB
Allocatable            yes
PE Size                4,00 MiB
Total PE               5119
Free PE                2752
Allocated PE           2367
PV UUID                mN0FfP-XTZI-de0g-SK8O-nIA0-kQSE-P6jAcV
...
```

Si lo ejecutamos con la opción `-m` veremos que también muestra las extensiones asignadas a los diferentes volúmenes lógicos.

```
usuario@U2004:~$ sudo pvdisplay -m
```

```
--- Physical volume ---
PV Name                /dev/sdb
VG Name                grupo_lvm
PV Size                20,00 GiB / not usable 4,00 MiB
Allocatable            yes
PE Size                4,00 MiB
Total PE               5119
Free PE                2752
Allocated PE           2367
PV UUID                mN0FfP-XTZI-de0g-SK8O-nIA0-kQSE-P6jAcV

--- Physical Segments ---
Physical extent 0 to 511:
  Logical volume /dev/grupo_lvm/DATOS1
  Logical extents 0 to 511
Physical extent 512 to 1023:
  Logical volume /dev/grupo_lvm/DATOS2
  Logical extents 0 to 511
Physical extent 1024 to 1086:
  Logical volume /dev/grupo_lvm/DATOS3
  Logical extents 449 to 511
Physical extent 1087 to 1598:
  Logical volume /dev/grupo_lvm/DATOS1
  Logical extents 512 to 1023
Physical extent 1599 to 1854:
  FREE
Physical extent 1855 to 2622:
  Logical volume /dev/grupo_lvm/DATOS2
  Logical extents 512 to 1279
Physical extent 2623 to 5118:
  FREE
```

El comando `pvscan` escanea todos los dispositivos de bloque LVM soportados en el sistema por volúmenes físicos. El siguiente comando muestra todos los dispositivos físicos encontrados

```
usuario@U2004:~$ sudo pvscan
PV /dev/sdb   VG grupo_lvm   lvm2 [<20,00 GiB / 10,75 GiB free]
PV /dev/sdc   VG grupo_lvm   lvm2 [<20,00 GiB / 18,24 GiB free]
PV /dev/sdd   VG grupo_lvm   lvm2 [<25,00 GiB / <25,00 GiB
free]
Total: 3 [<64,99 GiB] / in use: 3 [<64,99 GiB] / in no VG: 0 [0  ]
```

Además de los comandos anteriores también disponemos del comando `lvmdiskscan` el cual muestra información sobre los dispositivos que pueden usarse como volúmenes físicos. Si usamos la opción `-l` restringe la salida solamente a los volúmenes físicos actuales.

```
usuario@U2004:~$ sudo lvmdiskscan -l
WARNING: only considering LVM devices
```

```
/dev/sdb          [      20,00 GiB] LVM physical volume
/dev/sdc          [      20,00 GiB] LVM physical volume
/dev/sdd          [      25,00 GiB] LVM physical volume
3 LVM physical volume whole disks
0 LVM physical volumes
```

### 5.7.2 Visualizar información de grupos de volúmenes

El comando `vgscan`, que escanea todos los discos para grupos de volúmenes y reconstruye el fichero caché LVM también visualiza los grupos de volúmenes.

```
usuario@U2004:~$ sudo vgscan
Reading volume groups from cache.
Found volume group "grupo_lvm" using metadata type lvm2
usuario@U2004:~$
```

La salida del comando no ofrece mucha información, pero debería ser capaz de encontrar todos los grupos de volúmenes en el sistema.

Para obtener más información hay dos comandos para visualizar las propiedades de los grupos de volúmenes LVM: `vgs` y `vgdisplay`.

El comando `vgs` suministra información sobre el grupo de volumen en una forma configurable, visualizando una línea por grupo de volumen. Es altamente configurable para controlar el formato de salida, lo cual es útil para scripts. Por ejemplo, algunas modificaciones útiles en la salida son mostrar los volúmenes físicos y el path de los volúmenes lógicos.

```
usuario@U2004:~$ sudo vgs -o +devices,lv_path
VG          #PV #LV #SN Attr   VSize   VFree   Devices          Path
grupo_lvm   3   3   0 wz--n- <64,99g <53,99g /dev/sdb(0)
/dev/grupo_lvm/DATOS1
grupo_lvm   3   3   0 wz--n- <64,99g <53,99g /dev/sdb(1087)
/dev/grupo_lvm/DATOS1
grupo_lvm   3   3   0 wz--n- <64,99g <53,99g /dev/sdb(512)
/dev/grupo_lvm/DATOS2
grupo_lvm   3   3   0 wz--n- <64,99g <53,99g /dev/sdb(1855)
/dev/grupo_lvm/DATOS2
grupo_lvm   3   3   0 wz--n- <64,99g <53,99g /dev/sdc(128)
/dev/grupo_lvm/DATOS3
grupo_lvm   3   3   0 wz--n- <64,99g <53,99g /dev/sdb(1024)
/dev/grupo_lvm/DATOS3
```

Para una salida mejor interpretable el comando `vgdisplay` es la mejor opción. Añadiendo la opción `-v` también da información sobre los volúmenes físicos que componen el grupo de volúmenes, y los volúmenes lógicos que fueron creados utilizando del grupo de volúmenes.

### 5.7.3 Visualizar información de los volúmenes lógicos

Para visualizar información sobre los volúmenes lógicos LVM dispone de un conjunto de herramientas relacionadas.

Al igual que con los otros componentes LVM, el comando `lvscan` escanea el sistema y visualiza una mínima información sobre los volúmenes lógicos que encuentra.

```
usuario@U2004:~$ sudo lvscan
ACTIVE          '/dev/grupo_lvm/DATOS1' [4,00 GiB] inherit
ACTIVE          '/dev/grupo_lvm/DATOS2' [5,00 GiB] inherit
ACTIVE          '/dev/grupo_lvm/DATOS3' [2,00 GiB] inherit
```

Para información más completa, el comando `lvs` es flexible, potente y fácil de usar en scripts.

```
usuario@U2004:~$ sudo lvs
LV      VG      Attr      LSize Pool Origin Data%  Meta%  Move
Log Cpy%Sync Convert
DATOS1  grupo_lvm -wi-ao---- 4,00g
DATOS2  grupo_lvm -wi-a----- 5,00g
DATOS3  grupo_lvm -wi-a----- 2,00g
```

Con la opción `--segments` visualiza información sobre los segmentos y tipos de volumen lógico.

```
usuario@U2004:~$ sudo lvs --segments
LV      VG      Attr      #Str Type  SSize
DATOS1  grupo_lvm -wi-ao---- 1 linear 2,00g
DATOS1  grupo_lvm -wi-ao---- 1 linear 2,00g
DATOS2  grupo_lvm -wi-a----- 1 linear 2,00g
DATOS2  grupo_lvm -wi-a----- 1 linear 3,00g
DATOS3  grupo_lvm -wi-a----- 1 linear 1,75g
DATOS3  grupo_lvm -wi-a----- 1 linear 252,00m
```

El comando que produce mejor salida es `lvdisplay`. Con la opción `-m` la herramienta también muestra información sobre como los volúmenes lógicos están distribuidos.

```
usuario@U2004:~$ sudo lvdisplay -m
--- Logical volume ---
LV Path                /dev/grupo_lvm/DATOS1
LV Name                 DATOS1
VG Name                 grupo_lvm
LV UUID                 h1SKyk-vb0C-mj6r-AGBh-w9uh-jOYC-451AHm
LV Write Access         read/write
LV Creation host, time U2004, 2018-10-28 13:00:21 +0100
LV Status                available
# open                  1
LV Size                 4,00 GiB
Current LE              1024
Segments                2
Allocation               inherit
Read ahead sectors      auto
- currently set to     256
Block device            253:0
```

```
--- Segments ---
Logical extents 0 to 511:
  Type          linear
  Physical volume /dev/sdb
  Physical extents      0 to 511

Logical extents 512 to 1023:
  Type          linear
  Physical volume /dev/sdb
  Physical extents    1087 to 1598
```

Como se puede apreciar en la salida anterior, el volumen lógico `DATOS1` tiene un tamaño de 4 GB y ocupa dos segmentos en el mismo disco duro con un total de 1024 extensiones.

## 5.8 Eliminar un volumen lógico

Para eliminar un volumen lógico disponemos del comando `lvremove`. Debemos desmontar el volumen lógico antes de eliminarlo. El siguiente comando elimina el volumen lógico `/dev/grupo_lvm/DATOS3` del grupo de volumen `grupo_lvm`.

```
usuario@U2004:~$ sudo umount /dev/grupo_lvm/DATOS3
usuario@U2004:~$ sudo lvremove /dev/grupo_lvm/DATOS3
Do you really want to remove and DISCARD active logical volume
grupo_lvm/DATOS3? [y/n]: y
Logical volume "DATOS3" successfully removed
```

## 5.9 Eliminar un volumen físico

Si queremos eliminar un volumen físico de una gestión LVM, el procedimiento dependerá de si el dispositivo está siendo actualmente utilizado por el sistema LVM. En este caso, tendremos primero que mover las extensiones físicas localizadas en el dispositivo a una ubicación diferente. Esto requiere que el grupo de volumen tenga otros volúmenes físicos para mover las extensiones físicas del volumen que se pretende eliminar.

A modo de ejemplo supongamos que queremos eliminar el volumen físico `/dev/sdd` del grupo de volúmenes `grupo_lvm`. En este volumen físico teníamos creado un volumen lógico de la siguiente manera.

```
usuario@U2004:~$ sudo lvcreate -L 2GB -n DATOS4 grupo_lvm /dev/sdd:0-
Logical volume "DATOS4" created.
```

Vemos que todas sus extensiones están en `/dev/sdd` y ocupan 2 GB de espacio. Su nombre es `DATOS4`. Además, el volumen lógico tiene datos y se encuentra montado.

Para mover las extensiones físicas de un volumen empleamos el comando `pvmove`. En el siguiente ejemplo vamos a mover todas las extensiones del disco `/dev/sdd` que pretendemos eliminar del grupo de volúmenes.

```
usuario@U2004:~$ sudo pvmove /dev/sdd
/dev/sdd: Moved: 0,59%
/dev/sdd: Moved: 100,00%
```

Este proceso llevará un tiempo que dependerá del tamaño del volumen y la cantidad de datos a transferir. Una vez las extensiones físicas han sido recolocadas podemos eliminar el volumen físico del grupo de volúmenes.

```
usuario@U2004:~$ sudo vgreduce grupo_lvm /dev/sdd
Removed "/dev/sdd" from volume group "grupo_lvm"
```

Esto elimina el volumen físico del grupo de volúmenes. Cuando acabe, podemos eliminar el volumen físico del sistema LVM ejecutando el siguiente comando.

```
usuario@U2004:~$ sudo pvremove /dev/sdd
Labels on physical volume "/dev/sdd" successfully wiped.
```

Obviamente el dispositivo físico sigue existiendo y podemos utilizarlo para otros propósitos. Simplemente ha sido retirado del sistema de almacenamiento LVM.

Como podemos observar en este ejemplo no ha sido necesario desmontar el volumen lógico y por supuesto no se requiere eliminar los volúmenes lógicos que estuvieran ocupando extensiones en el dispositivo físico. Las extensiones de los volúmenes lógicos se han movido a otros volúmenes físicos como se comentó anteriormente. La pregunta es ¿dónde?. Si hacemos una consulta del volumen lógico con `pvdisplay` y la opción `-m` veremos lo siguiente.

```
--- Logical volume ---
LV Path                /dev/grupo_lvm/DATOS4
LV Name                 DATOS4
VG Name                 grupo_lvm
LV UUID                 1oM1r9-dr8U-jqIB-Y7Wg-VTWR-1oHU-xJZmNE
LV Write Access         read/write
LV Creation host, time  U2004, 2018-10-29 18:31:05 +0100
LV Status                available
# open                  1
LV Size                 2,00 GiB
Current LE              512
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to     256
Block device            253:2

--- Segments ---
Logical extents 0 to 511:
Type                  linear
Physical volume /dev/sdb
Physical extents      2623 to 3134
```

Ahora están ocupado las extensiones físicas de la 2623 a la 3134 del volumen físico



/dev/sdb. Si ejecutamos el comando `mount` veremos que el volumen lógico sigue montado y por tanto podemos acceder a la información que almacena.

```
...  
/dev/mapper/grupo_lvm-DATOS4 on /mnt/DATOS4 type ext4  
(rw,relatime,data=ordered)
```

## 5.10 Eliminar un grupo de volúmenes

Para eliminar un grupo de volúmenes completo, incluyendo todos los volúmenes lógicos dentro de él, utilizamos el comando `vgremove`.

Antes de eliminar el grupo de volúmenes, deberíamos eliminar los volúmenes lógicos utilizando el procedimiento que vimos en un apartado anterior. Como mínimo, deberíamos asegurarnos de que hemos desmontado todos los volúmenes lógicos que contiene el grupo de volúmenes.

```
sudo umount /dev/grupo_lvm/DATOS1  
sudo umount /dev/grupo_lvm/DATOS4
```

Después, podemos borrar el grupo de volúmenes completo indicando como argumento su nombre al comando `vgremove`.

```
sudo vgremove grupo_lvm
```

El comando nos pedirá confirmación para el borrado. Si el grupo de volúmenes todavía contiene volúmenes lógicos, nos pedirá confirmación para borrarlos individualmente antes de eliminar el grupo de volúmenes.

## 5.11 RAID

Vamos a ver a continuación aplicaciones concretas de LVM para crear y mantener volúmenes RAID. Damos por hecho que ya tenemos creado los volúmenes físicos y el grupo de volúmenes de los epígrafes anteriores para a partir de ahora mostrar los ejemplos de creación de los volúmenes RAID.

### 5.11.1 RAID 0

Para crear un volumen RAID 0 debemos especificar al menos dos divisiones (stripes). Recordemos que un volumen RAID 0 divide el almacenamiento en dos discos en los que los archivos se almacenan en uno de ellos. De esta forma se aumenta la tasa de lectura y escritura, aunque no hay tolerancia a fallos.

Para crear un volumen RAID 0 debemos incluir la opción `--type` con el valor `striped` y el número de divisiones en la opción `-i` con el valor `2`. Luego indicamos su tamaño y nombre como vimos anteriormente y el grupo de volúmenes donde se crea.

Por ejemplo, supongamos que necesitamos un volumen RAID 0 de 10 GB. Tendríamos que teclear el siguiente comando:

```
usuario@U2004:~$ sudo lvcreate --type striped -i 2 -L 10G -n vraid0
grupo_lvm
Using default stripesize 64,00 KiB.
Logical volume "vraid0" created.
```

Esta topología requiere de al menos dos volúmenes físicos con suficiente capacidad en los que se colocará cada una de las divisiones del volumen.

### 5.11.2 RAID 1

Para crear un volumen reflejado (RAID 1) debemos utilizar el tipo `raid1` de la opción `--type`. Para indicar el conjunto de copias de datos establecemos la opción `-m`. El siguiente ejemplo crea un volumen RAID 1 con dos copias (la original y el espejo). En este caso necesitamos dos volúmenes físicos, ya que cada copia se almacena en un disco físico diferente.

```
usuario@U2004:~$ sudo lvcreate --type raid1 -m 1 -L 10G -n vraid1
grupo_lvm
Logical volume "vraid1" created.
```

Vemos que la opción `-m` indica número de copias sin contar con la original, es decir, se indican el número de espejos. Si queremos tres copias de datos (la original y dos espejos) tendríamos que haber ejecutado el siguiente comando.

```
usuario@U2004:~$ sudo lvcreate --type raid1 -m 2 -L 10G -n vraid1
grupo_lvm
Logical volume "vraid1" created.
```

En este caso necesitaríamos tres volúmenes físicos.

### 5.11.3 RAID 5

Para crear un volumen RAID 5 necesitamos 2 divisiones más una implícita que se emplea para la información de paridad. En el siguiente ejemplo vamos a crear un volumen de 5GB. Debemos emplear el tipo `raid5` en la opción `--type` y con la opción `-i` el número de divisiones (2 como mínimo). El número de unidades para la información de paridad se añade automáticamente.

```
usuario@U2004:~$ sudo lvcreate --type raid5 -i 2 -L 5GB -n vraid5
grupo_lvm
Using default stripesize 64,00 KiB.
Logical volume "vraid5" created.
```

## 5.12 Instantáneas

La característica de instantánea de LVM permite crear volúmenes lógicos, que son imágenes de un volumen en un instante concreto, sin necesidad de interrumpir su servicio. Cuando se produce un cambio en el dispositivo original (el volumen lógico origen) después de crear un volumen instantánea, este realiza una copia del área de datos cambiada antes del cambio para que se pueda reconstruir el estado del volumen lógico origen.

Ya que en una instantánea se copia solo las áreas de datos que cambian después de que el volumen instantánea se crea, este requiere una mínima cantidad de almacenamiento. Por ejemplo, con un volumen lógico origen que raramente se actualiza entre el 3 y el 5% de la capacidad del volumen lógico origen es suficiente para mantener la instantánea.

Si un volumen instantánea se llena, se vuelve corrupta y no podríamos utilizarla. Esto se hace para asegurar que hay bastante espacio en el sistema de archivos del volumen lógico origen. Deberemos, por tanto, monitorizar el tamaño de la instantánea. Podemos cambiar el tamaño de la instantánea cuando se necesite para evitar esta situación. Por otro lado, si la instantánea es más grande de lo que necesitamos, podemos reducir su tamaño para liberar espacio que se necesitan para otros volúmenes lógicos.

Cuando creamos un sistema de archivos del volumen instantánea, el acceso completo en lectura y escritura en el volumen lógico origen permanece. Si un trozo de una instantánea cambia, ese trozo es marcado y nunca se volvería a copiar desde el volumen lógico origen.

Las instantáneas tienen varios usos:

- ✓ El más típico es para realizar una copia de seguridad de un volumen lógico sin parar el sistema de archivos mientras se realiza la copia.
- ✓ Se puede realizar un chequeo del sistema de archivos de una instantánea para comprobar su integridad y determinar si el sistema de archivos original necesita ser reparado.
- ✓ Podemos probar aplicaciones con datos de producción tomando una instantánea antes de ejecutar pruebas y recuperarla posteriormente dejando los datos reales originales.

### 5.12.1 Crear una instantánea

Supongamos que disponemos de un volumen lógico llamado `datos`, que está montado en el directorio `/mnt/datos`, sobre el que queremos crear una instantánea y que tiene el siguiente contenido.

```
usuario@U2004:~$ ls -l /mnt/datos
total 32
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo3.txt
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo1.txt
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo2.txt
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo.txt
drwx----- 2 root      root    16384 oct 31 11:01 lost+found
```

Para crear una instantánea utilizamos la opción `-s` o `--snapshot` del comando `lvcreate`. El siguiente comando crea un volumen lógico instantánea con un tamaño de 200 MB con nombre `inst_datos` que es una imagen del volumen lógico `datos`.

```
usuario@U2004:~$ sudo lvcreate -L 200M -n inst_datos -s
/dev/grupo_lvm/datos
Using default stripesize 64,00 KiB.
```

```
Logical volume "inst_datos" created.
```

Nótese que para crear una instantánea de un volumen debemos indicar el volumen lógico origen en lugar del grupo de volúmenes.

Si el volumen lógico origen contiene un sistema de archivos, podemos montar la instantánea para acceder a los contenidos actuales y, si es necesario, para ejecutar una copia mientras el volumen lógico original continua siendo actualizado. Por ejemplo, si montamos la instantánea que acabamos de crear veremos lo siguiente.

```
usuario@U2004:~$ sudo mount /dev/grupo_lvm/inst_datos /mnt/inst_datos
usuario@U2004:~$ ls -l /mnt/inst_datos
total 32
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo3.txt
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo1.txt
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo2.txt
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo.txt
drwx----- 2 root      root    16384 oct 31 11:01 lost+found
```

Se aprecia que es el mismo contenido del volumen lógico original en el momento de crear la instantánea.

Después de crear la instantánea, cuando utilizamos el comando `lvdisplay` para visualizar información del volumen lógico original veremos que nos informa sobre la existencia de una instantánea sobre este volumen lógico. Por ejemplo, el siguiente comando muestra la información del volumen lógico `datos` que hemos utilizado para crear una instantánea y también muestra su instantánea.

```
usuario@U2004:~$ sudo lvdisplay /dev/grupo_lvm/datos
--- Logical volume ---
LV Path                /dev/grupo_lvm/datos
LV Name                 datos
VG Name                grupo_lvm
LV UUID                MGXmFD-s1KG-r0Lv-cZ3b-KyRS-wdnH-7Vujph
LV Write Access         read/write
LV Creation host, time U2004, 2018-10-31 10:58:41 +0100
LV snapshot status      source of
                        inst_datos [active]
LV Status               available
# open                  1
LV Size                 2,00 GiB
Current LE              512
Segments                1
Allocation              inherit
Read ahead sectors      auto
 - currently set to     256
Block device            253:13
```

El comando `lvs` por defecto muestra el volumen lógico origen y el porcentaje actual de la instantánea que está siendo usado. El siguiente ejemplo muestra la salida por defecto del comando `lvs` para un sistema que incluye el volumen lógico `/dev/grupo_lvm/`

`datos` para el cual se ha creado un volumen instantánea `/dev/grupo_lvm/inst_datos`.

```
usuario@U2004:/mnt/datos1$ sudo lvs
  LV          VG          Attr          LSize   Pool Origin Data%  Meta%
Move Log Cpy%Sync Convert
  datos        grupo_lvm owi-aos---    2,00g
  inst_datos    grupo_lvm swi-a-s--- 200,00m      datos  0,01
```

Las instantáneas no necesitan mucho espacio inicialmente, pero crecen en tamaño conforme se realizan cambios en el volumen lógico origen. El tamaño utilizado durante el procedimiento es el tamaño máximo que la instantánea puede tener. Las instantáneas que crecen más allá de este tamaño se rompen y no pueden ser utilizadas, aunque cuando se aproximen a su capacidad total podemos aumentarlas de tamaño.

### 5.12.2 ¿Cómo funciona una instantánea?

Cuando creamos una instantánea sobre un volumen lógico, suponemos que tenemos dos volúmenes lógicos (el original y la instantánea), pero en realidad tenemos cuatro volúmenes lógicos. Si ejecutamos el comando `dmsetup` veremos la siguiente información.

```
usuario@U2004:~$ sudo dmsetup table
grupo_lvm-datos: 0 20971520 snapshot-origin 253:1
grupo_lvm-ins_datos-cow: 0 1024000 linear 8:32 20973568
grupo_lvm-ins_datos: 0 20971520 snapshot 253:1 253:2 P 8
grupo_lvm-datos-real: 0 20971520 linear 8:32 2048
usuario@PC00:~$
```

Lo que ha ocurrido es que el volumen original (`datos`) se ha renombrado a `datos-real` y se ha creado otro con el nombre del volumen lógico origen. Posteriormente ha creado la instantánea `ins_datos` y otro volumen llamado `ins_datos-cow`. ¿Para que sirve tanto volumen?

En principio los datos originales están en `datos-real`. El volumen `datos` solamente está para que el usuario interactúe con él. De igual forma, el volumen `ins_datos` (la instantánea) también está para interactuar con ella. El volumen `ins_datos-cow` es el que se emplea para copiar los datos que han cambiado en el volumen original antes de su actualización. El sufijo `cow` significa *Copy On Write*, lo que significa que cada vez que se actualiza un bloque del volumen original, antes de su sobreescritura se copia en el volumen `ins_datos-cow` para conservar el que había antes. A partir de ahora las operaciones de acceso a disco se realizan de la siguiente manera:

- ✓ Lectura sobre el volumen original → El usuario utiliza el volumen `datos` para leer archivos y la petición se adelanta al volumen `datos-real`.
- ✓ Escritura sobre el volumen original → El usuario crea, modifica o borra archivos en el volumen `datos`. Primero se copian los bloques originales del volumen desde `datos-real` a `ins_datos-cow` y los nuevos bloques sobrescriben los originales en `datos-real`. Si estos bloques ya fueron actualizados previamente,

simplemente se sobrescriben los originales en `datos-real`.

- ✓ Lectura sobre la instantánea → Cuando el usuario lee datos de `ins_datos` primero se comprueba si esos datos fueron actualizados. Si es así, se leen desde el volumen `ins-datos-cow`, y si no, se leen los datos originales desde el volumen `datos-real`. En cualquier caso el usuario tendrá los datos originales en el momento de crear la instantánea.
- ✓ Escritura sobre la instantánea → Cuando el usuario realiza una escritura sobre `ins_datos` los datos modificados son los de `ins_datos-cow`. En realidad, primero se comprueba si habían sido previamente modificados en cuyo caso contrario se copiarían desde `datos-real` a `ins_datos-cow`, y luego se sobrescriben. Una posterior actualización de los mismos datos provocaría una sobrescritura en `ins_datos-cow`.

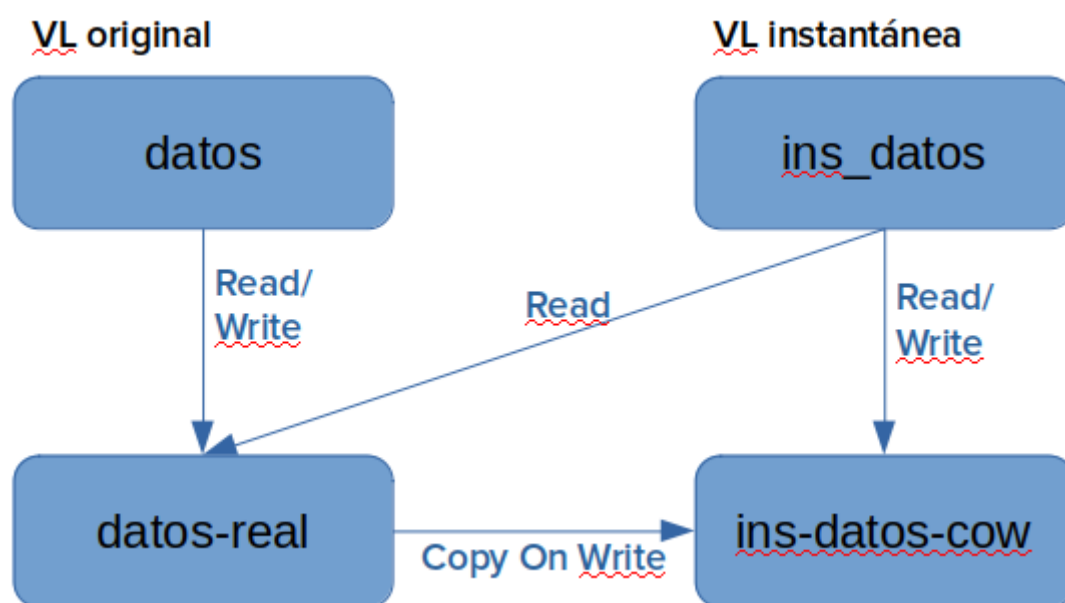


Figura 17: Funcionamiento de las instantáneas

### 5.12.3 Revertir el volumen lógico a la instantánea

Cuando queremos recuperar un volumen lógico a su estado original con la instantánea tenemos que utilizar el comando `lvconvert` con la opción `--merge`. Antes, es recomendable desmontar el volumen lógico original en el caso de que estuviera montado.

```
usuario@U2004:~$ sudo umount /dev/grupo_lvm/datos
```

Posteriormente, reversionamos el volumen lógico a la instantánea. Por ejemplo, si queremos restaurar el volumen lógico `datos` al cual creamos una instantánea anteriormente tendríamos que ejecutar el siguiente comando.

```
usuario@U2004:~$ sudo lvconvert --merge grupo_lvm/inst_datos
```

También, podríamos haber hecho la reversión sin desmontar el volumen lógico original.

```
usuario@U2004:~$ sudo lvconvert --merge grupo_lvm/inst_datos
Can't merge until origin volume is closed.
Merging of snapshot grupo_lvm/inst_datos will occur on next
activation of grupo_lvm/datos.
```

En este caso la ejecución del comando indicará que se revertirá el volumen lógico en la próxima activación. Para activar de nuevo el volumen tenemos que desmontar el volumen lógico origen, desactivarlo y volverlo a activar con la siguiente sucesión de comandos.

```
usuario@U2004:~$ sudo umount /dev/grupo_lvm/datos
usuario@U2004:~$ sudo lvchange -an /dev/grupo_lvm/datos
usuario@U2004:~$ sudo lvchange -ay /dev/grupo_lvm/datos
```

Esto llevará al volumen lógico al estado en que se encontraba en el momento que se hizo la instantánea. Si volvemos a montar el volumen lógico original y listamos su contenido veremos que ha vuelto a su estado original.

```
usuario@U2004:~$ sudo mount /dev/grupo_lvm/datos /mnt/datos
usuario@U2004:~$ ls -l /mnt/datos
total 32
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo3.txt
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo1.txt
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo2.txt
-rw-r--r-- 1 usuario usuario    5 oct 31 11:02 archivo.txt
drwx----- 2 root      root    16384 oct 31 11:01 lost+found
```

En el caso de que no se pueda revertir el volumen lógico con este procedimiento se realizará la próxima vez que se reinicie el sistema. Obviamente, si desmontamos el volumen lógico original antes de revertirlo a su estado original, no será necesario la reactivación.

## 6 Bibliografía

---

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

MAN, *Páginas de manual de Linux*

UBUNTU 18.04, *Guía del escritorio de Ubuntu*

RUIZ, P, *Copias de seguridad integradas en Ubuntu 16.04 LTS (parte I)* [accedido diciembre 2017]. Disponible en <<http://somebooks.es/copias-de-seguridad-integradas-en-ubuntu-16-04-lts-parte-i/>>

JELLINGWOOD, J. *How To Use LVM To Manage Storage Devices on Ubuntu 16.04* [accedido octubre 2018]. Disponible en <<https://www.digitalocean.com/community/tutorials/how-to-use-lvm-to-manage-storage-devices-on-ubuntu-16-04>>

RUIZ, P. *Cuotas de disco* [accedido octubre 2018]. Disponible en <<http://somebooks.es/9-6-cuotas-de-disco/>>

GONZALEZ, S. *Implementar y administrar cuotas de disco en Linux* [accedido octubre 2018]. Disponible en <[https://www.linuxtotal.com.mx/index.php?cont=info\\_admon\\_018](https://www.linuxtotal.com.mx/index.php?cont=info_admon_018)>

LANDMANN, R. *Red Hat Enterprise Linux 4 - Cluster Logical Volume Manager - LVM Administrator's Guide - Edition 1.0* [accedido octubre 2018] Disponible en <[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/4/html/Cluster\\_Logical\\_Volume\\_Manager/index.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Cluster_Logical_Volume_Manager/index.html)>

DEEPAK,P. *Understanding LVM snapshots (create, merge, remove, extend)* [accedido octubre 2018] Disponible en <<https://www.golinuxhub.com/2017/09/understanding-lvm-snapshots-create.html>>

RAFFO, D. *LVM SNAPSHOTS EXPLAINED* [accedido abril 2020] Disponible en <<https://www.clevernetsystems.com/lvm-snapshots-explained/>>