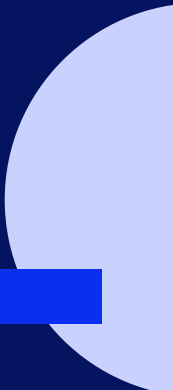

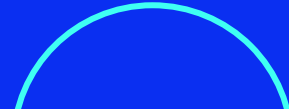


# Bootcamp DevOps Engineer

Fase 1 - SysAdmin  
Módulo 1



# Crear particiones y sistemas de archivos

# Manejo de particiones

En este tópico, veremos cómo realizar particiones, crear sistemas de archivos, entre otras cosas.



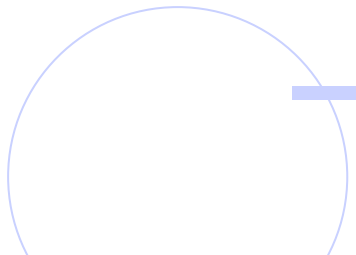
# Identificación de discos

En Linux, los discos rígidos son listados y representados dentro de la **carpeta /dev**. Para ser identificados, el kernel les agrega un identificador que comienza con **hd** para el caso de **discos IDE**, o **sd** para el caso de **discos SATA o SCSI**. Adicionalmente, a cada uno de estos identificadores se les agrega una **letra del alfabeto para su posición**.



## Identificador del Disco Duro Dispositivo Lógico

<b>hda</b>	Maestro Interfaz Primaria
<b>hdb</b>	Esclavo Interfaz Primaria
<b>hdc</b>	Maestro Interfaz Secundaria
<b>hdd</b>	Esclavo Interfaz Secundaria



## SATA, SCSI y dispositivos de almacenamiento USB

<b>sda</b>	Primer Disco
<b>sdb</b>	Segundo Disco
<b>sdc</b>	Tercer Disco



## Identificador de volúmenes lógicos

Los volúmenes lógicos se pueden listar de la siguiente manera (suponiendo que el **VG** -*Volume Group*- existente se llama **vglinux**):

```
# ls -l /dev/fedora/+( [a-z A-Z0-9+_.- ] )
```

Otra manera de mostrar los dispositivos de este tipo es así:

```
# ls -l /dev/mapper
```

**Nota:** el archivo */dev/mapper/control* usado por herramientas de visualización y configuración de LVM's y otros tipos de dispositivos.

# Configuración de particiones

Así como los discos rígidos son listados y representados, **también las particiones correspondientes a cada disco duro son representadas y listadas**. Para ser identificados, el *kernel* agrega un número al final de cada **identificador** para diferenciar una partición primaria de una lógica o extendida.

**Nota:** en el esquema de particiones MBR los discos rígidos permiten un máximo de 4 particiones primarias, de las cuales una puede ser extendida. **Las particiones extendidas pueden contener particiones lógicas.**



# Aplicaciones gráficas para particionar discos duros

Existen **aplicaciones libres** como alternativas a las aplicaciones propietarias (*Partitioning Magic*, entre otras). A continuación veremos tres de los mejores exponentes:

- **Gparted.**
- **KDE Partition Manager.**
- **Parted Magic.**



## Gparted

GParted es el editor de particiones de GNOME. Esta aplicación es usada para **crear, eliminar, redimensionar, inspeccionar y copiar particiones, como también sistemas de archivos**. Esto es útil para crear espacio para nuevos sistemas operativos, reorganizar el uso del disco y crear imágenes de un disco en una partición. La aplicación usa la librería **libparted** para detectar y manipular dispositivos y tablas de partición, mientras varias herramientas de sistema de archivos dan mantenimiento a sistemas de archivos no incluidos en **libparted**.

- Está escrito en **C++** y utiliza **gtkmm** como herramienta gráfica. Este acercamiento es para mantener la interfaz gráfica de usuario lo más simple posible, conforme con las *Human Interface Guidelines*.
- **Gparted LiveCD:** se encuentra disponible en LiveCD, basado en Slackware y construido sobre la última rama estable núcleo de Linux (2.6). LiveCD es actualizado con cada lanzamiento de GParted. El LiveCD de Ubuntu incluye esta aplicación entre sus utilidades. También se encuentra disponible en versión LiveUSB.
- **Capacidades y limitaciones:** GParted no puede incrementar el tamaño de las particiones sin existir un espacio vacío después de dicha partición, es decir, si existen dos particiones juntas no se podrá aumentar el tamaño de una en detrimento de la otra; esto es más bien una limitación técnica. En esta tabla se muestran las capacidades de GParted, de acuerdo con cada sistema de archivos.

Si quieres probar esta aplicación, la puedes descargar directamente de la página oficial del proyecto o usando los repositorios de la distribución. <http://gparted.sourceforge.net>.



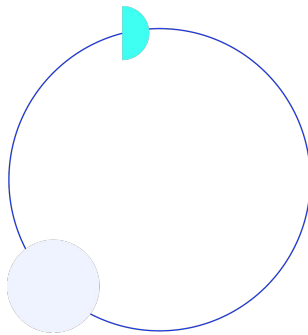
## KDE Partition Manager

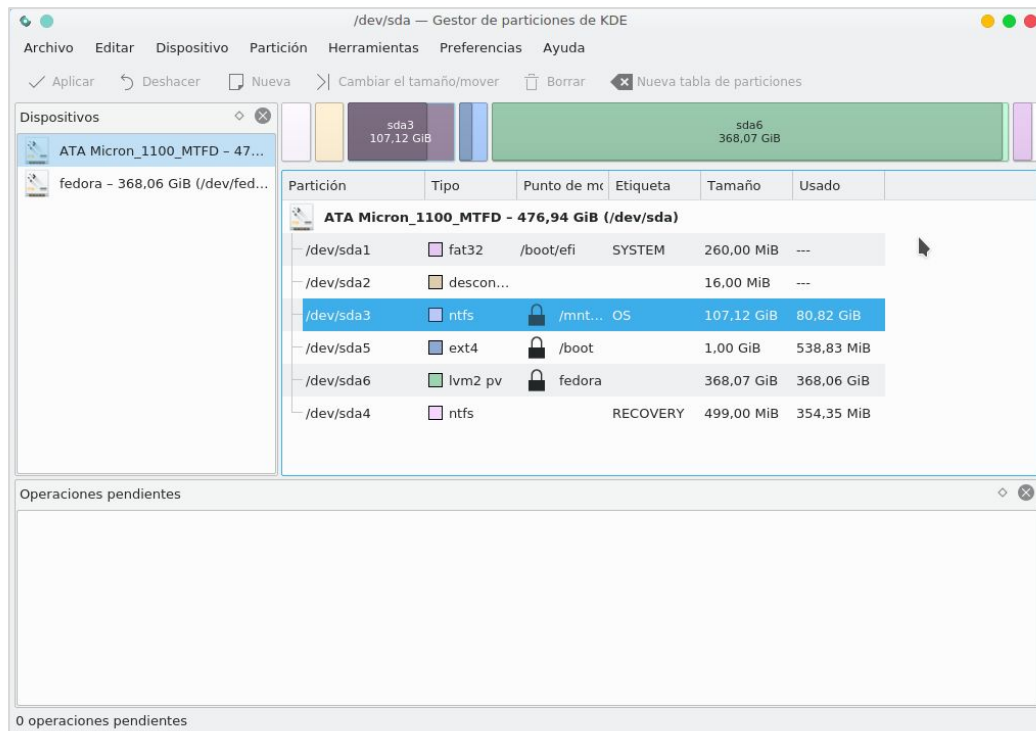
Se basa en el programa de consola Parted.

- Puede administrar discos, particiones y sistemas de archivos. Permite **crear, copiar, mover, borrar, redimensionar sin pérdida de datos, hacer copias de respaldo y recuperar particiones**.
- Soporta los siguientes sistemas de archivos: *ext2, ext3, ext4, reiserfs, NTS, FAT16/32, jfs, xfs* y más.

- Lo más probable es que lo encuentres en el repositorio de tu distribución.

Además existen distintas maneras de descargarlos desde la siguiente URL: [KDE Partition Manager](#).

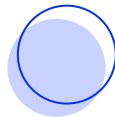




## Parted Magic

- **Parted Magic es un LiveCD que incluye una distribución Linux a medida**, para poder operar directamente con ella, sin necesidad de ser instalada en el disco duro. No estamos hablando solo de un particionador de discos. En este LiveCD podemos encontrar varias herramientas, entre ellas, un editor de particiones llamado **VisParted** basado en el genuino Gparted, con el que podremos **crear, redimensionar y borrar nuestras particiones del disco duro**.
- Soporta los siguientes sistemas de archivos: *ext2, ext3, ext4, fat16, fat32, hfs, hfs+, jfs, linux-swap, ntfs, reiserfs, reiser4* y *xfs*.

Si quieres probar esta aplicación, la puedes descargar directamente de la página oficial del proyecto, previo abono del mismo: [Parted Magic](#).



# Formatear y particionar discos rígidos desde BASH: dando formato de bajo nivel con dd

Esta herramienta **sirve para dar formato de bajo nivel a un disco rígido (escribirá cada sector del disco)**. El proceso puede variar dependiendo del tamaño de almacenamiento del disco duro o del tipo de interfaz (IDE o SATA). La forma de implementar esta herramienta es la siguiente:

```
# dd if=/dev/zero of=/dev/sda
```

- **dd:** El comando dd (*duplicate disk*) es un comando bastante útil para transferir datos desde un dispositivo o archivo, hacia otro dispositivo o archivo.
- **if=/dev/zero:** if significa *input file*, es decir, el origen a copiar. En este caso, el origen es el dispositivo zero, para escribir el carácter zero en todo el disco.
- **of=/dev/sda:** of significa *output file*, o sea, el dispositivo o archivo destino donde se van a copiar los datos. El ejemplo se refiere al disco rígido.

# Particionando un disco duro con Fdisk

Fdisk es una aplicación disponible para varios sistemas operativos. Esta utilidad **permite dividir en forma lógica un disco duro, siendo denominado este nuevo espacio como partición.**

La descripción de las particiones se guarda en la tabla de particiones que se localiza en el **sector 0 de cada disco.** La versión fdisk de Linux permite crear particiones en 94 sistemas de archivos distintos, incluyendo FAT32, ext3, Solaris y QNX. Esta versión de fdisk cuenta con un menú de texto de ayuda en línea para realizar las operaciones.



## Opciones

-1

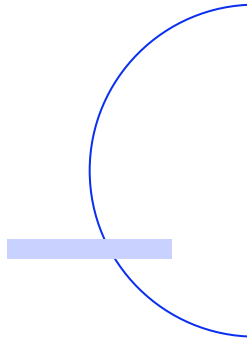
Este parámetro listará la tabla de particiones.

Ingresa en el modo de particionamiento en el disco  
**/dev/sda**

```
# fdisk /dev/sda
```

Al lanzar la aplicación, esta nos arrojará información referente al número de cabezas, sectores, cilindros, particiones o tamaño del dispositivo.

```
Disk /dev/sda: 146.8 GB, 146815737856 bytes  
255 heads, 63 sectors/track, 17849 cylinders  
Units = cylinders of 16065 * 512 = 8225280 bytes
```



Una vez que la aplicación está iniciada, se nos presenta el siguiente mensaje:

```
Command (m for help):
```

Si presionamos la tecla '**m**', se imprimirá el menú con las herramientas propias del comando **fdisk**. Estas herramientas son las detalladas en la tabla de la derecha:

<b>a</b>	Conmuta el indicador de iniciable
<b>b</b>	Modifica la etiqueta de disco <b>bsd</b>
<b>c</b>	Conmuta el indicador de compatibilidad con DOS
<b>d</b>	Suprime una partición
<b>l</b>	Lista los tipos de particiones conocidos
<b>m</b>	Imprime este menú
<b>n</b>	Añade una nueva partición
<b>o</b>	Crea una nueva tabla de particiones DOS vacía
<b>p</b>	Imprime la tabla de particiones
<b>q</b>	Salir sin guardar los cambios
<b>s</b>	Crea una nueva etiqueta de disco <b>Sun</b>
<b>t</b>	Cambia el identificador de sistema de una partición
<b>u</b>	Cambia las unidades de visualización/entrada
<b>v</b>	Verifica la tabla de particiones
<b>w</b>	Escribe la tabla en el disco y sale
<b>x</b>	Funciones adicionales (solo para usuarios avanzados)

Por ejemplo, con “p” obtendremos la tabla de particiones actual del disco duro en cuestión:

```
Command (m for help): p
```

```
Disk /dev/sda: 146.8 GB, 146815737856 bytes  
255 heads, 63 sectors/track, 17849 cylinders  
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	25	200781	83	Linux
/dev/sda2		26	2575	20482875	83	Linux
/dev/sda3		2576	2706	1052257+	82	Linux swap / Solaris
/dev/sda4		2707	4000	10394055	8e	Linux LVM



Si lo que queremos es **borrar una partición** (digamos la 3):

```
Command (m for help): d
Partition number (1-4): 3
Command (m for help)
```

En el ejemplo, hemos borrado la partición correspondiente a la memoria extendida (swap). Para crearla nuevamente usamos “n”:

```
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Selected partition 3
First cylinder (2576-17849, default 2576):
Using default value 2576
Last cylinder or +size or +sizeM or +sizeK (2576-2706, default 2706):
Using default value 2706
```

- **Tipo de partición:** Aquí se nos pide elegir entre *partición primaria* y *partición extendida*.
- **Selección de la partición:** Escribir el número de la partición que se va a crear. "3" en nuestro ejemplo.
- **Primer cilindro:** Es el cilindro en el que comienza la partición. Normalmente elegimos el que el sistema nos marca de manera predeterminada pulsando *Enter*.
- **Último cilindro o tamaño de la partición:** En el ejemplo, se pulsó *Enter*, para tomar el valor predeterminado. Generalmente resulta más cómodo proporcionar el tamaño en megas (o en K's). Una opción sería entonces dar: +2048M para asignar 2 gigas a una partición.



Tabla de particiones actual:

Command (m for help): p

Disk /dev/sda: 146.8 GB, 146815737856 bytes  
255 heads, 63 sectors/track, 17849 cylinders  
Units = cylinders of 16065 \* 512 = 8225280 bytes

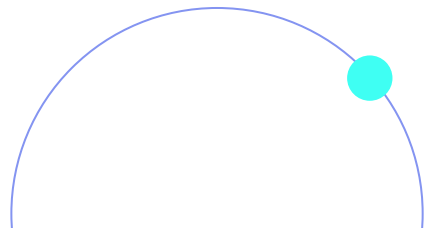
Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	25	200781	83	Linux
/dev/sda2		26	2575	20482875	83	Linux
/dev/sda3		2576	2706	1052257+	83	Linux
/dev/sda4		2707	4000	10394055	8e	Linux LVM

- En la **primera columna**, se encuentra el dispositivo resultado de la **partición de disco sda**. Podemos observar que en este momento contamos con **4 particiones: sda1, sda2, sda3 y sda4**.
- La **segunda columna** nos informa que es la primera partición del disco la que contiene la **parte bootable**.
- La **tercera y cuarta columna** brindan información acerca del **cilindro** en el cual comienza y termina la partición respectivamente.
- La **quinta** nos dice en **bloques cuánto ocupa**.
- Por último, con la **sexta y séptima columna** sabemos el **tipo de partición** (Identificador y sistema, respectivamente).



De nuestro ejemplo, sabemos que contamos con 3 particiones “Linux” (Id 83) y una “Linux LVM” (Id 8e). Si pensamos utilizar la partición recién creada como **SWAP**, hay que **cambiar el tipo de partición pulsando la letra t**.

```
Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 82
Changed system type of partition 3 to 82 (Linux swap / Solaris)
```



Verificar la tabla actual:

```
Command (m for help): p
Disk /dev/sda: 146.8 GB, 146815737856 bytes
255 heads, 63 sectors/track, 17849 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	25	200781	83	Linux
/dev/sda2		26	2575	20482875	83	Linux
/dev/sda3		2576	2706	1052257+	82	Linux swap / Solaris
/dev/sda4		2707	4000	10394055	8e	Linux LVM

Con **fdisk** podemos asignar muchos otros tipos de sistema. Con “**l**” podemos obtener la **lista completa de opciones**, como vemos en la siguiente pantalla.



Command (m for help): 1

0 Empty	1e Hidden W95 FAT1	80 Old Minix	be Solaris boot
1 FAT12	24 NEC DOS	81 Minix / old Lin	bf Solaris
2 XENIX root	39 Plan 9	82 Linux swap / So	c1 DRDOS/sec (FAT-
3 XENIX usr	3c PartitionMagic	83 Linux	c4 DRDOS/sec (FAT-
4 FAT16 <32M	40 Venix 80286	84 OS/2 hidden C:	c6 DRDOS/sec (FAT-
5 Extended	41 PPC PReP Boot	85 Linux extended	c7 Syrix
6 FAT16	42 SFS	86 NTFS volume set	da Non-FS data
7 HPFS/NTFS	4d QNX4.x	87 NTFS volume set	db CP/M / CTOS / .
8 AIX	4e QNX4.x 2nd part	88 Linux plaintext	de Dell Utility
9 AIX bootable	4f QNX4.x 3rd part	8e Linux LVM	df BootIt
a OS/2 Boot Manag	50 OnTrack DM	93 Amoeba	e1 DOS access
b W95 FAT32	51 OnTrack DM6 Aux	94 Amoeba BBT	e3 DOS R/O
c W95 FAT32 (LBA)	52 CP/M	9f BSD/OS	e4 SpeedStor
e W95 FAT16 (LBA)	53 OnTrack DM6 Aux	a0 IBM Thinkpad hi	eb BeOS fs
f W95 Ext'd (LBA)	54 OnTrackDM6	a5 FreeBSD	ee EFI GPT
10 OPUS	55 EZ-Drive	a6 OpenBSD	ef EFI (FAT-12/16/
11 Hidden FAT12	56 Golden Bow	a7 NeXTSTEP	f0 Linux/PA-RISC b
12 Compaq diagnost	5c Priam Edisk	a8 Darwin UFS	f1 SpeedStor
14 Hidden FAT16 <3	61 SpeedStor	a9 NetBSD	f4 SpeedStor
16 Hidden FAT16	63 GNU HURD or Sys	ab Darwin boot	f2 DOS secondary
17 Hidden HPFS/NTF	64 Novell Netware	b7 BSDI fs	fd Linux raid auto
18 AST SmartSleep	65 Novell Netware	b8 BSDI swap	fe LANstep
1b Hidden W95 FAT3	70 DiskSecure Mult	bb Boot Wizard hid	ff BBT
1c Hidden W95 FAT3	75 PC/IX		

Para **guardar los cambios** pulsar la **tecla w**.

```
Command (m for help): w\\
```

```
¡Se ha modificado la tabla de particiones!
```

```
Llamando a ioctl() para volver a leer la tabla de particiones.
```

```
Se están sincronizando los discos.
```





## Particionando de forma No-Interactiva

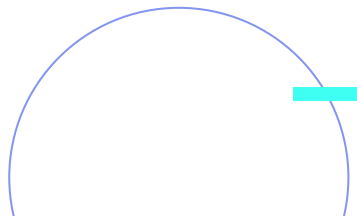
Se puede crear un archivo con los comandos de **fdisk** para **particionar múltiples dispositivos de la misma manera**.

```
[root@oc6127656113 ~]# cat comandos-fdisk.txt
d
n
p
1

t
83
w
```

En el archivo, primero se borra la partición única que posee (**d**), luego crea una nueva (**n**) y la deja lista para usarla con Linux (**t 83**).

Finalmente, si suplantamos la **w** (escribir la tabla en el disco y salir) por una **p** (imprimir la tabla de particiones) y luego agregamos una **q** (salir sin guardar), podemos ver cómo funciona sin romper nada, como se muestra en el siguiente slide.



```
# fdisk /dev/sdb < comandos-fdisk.txt
```

```
Command (m for help): Selected partition 1
```

```
Command (m for help): Command action
```

```
  e   extended
```

```
  p   primary partition (1-4)
```

```
Partition number (1-4, default 1): First sector (2048-31252023, default 2048): Using default value 2048
```

```
Last sector, +sectors or +size{K,M,G} (2048-31252023, default 31252023): Using default value 31252023
```

```
Command (m for help): Selected partition 1
```

```
Hex code (type L to list codes):
```

```
Command (m for help):
```

```
Disk /dev/sdb: 16.0 GB, 16001036288 bytes
```

```
50 heads, 24 sectors/track, 26043 cylinders, total 31252024 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
```

```
Disk identifier: 0x0002f2b2
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	31252023	15624988	83	Linux

```
Command (m for help):
```

# Uso de gdisk y parted con GPT

La herramienta **gdisk** sirve para **particionar un sistema que usa GPT en lugar de MBR**. GPT proporciona soporte para mayor número de particiones, más confiabilidad y discos más grandes.

El modo de uso es similar al de **fdisk**, por ejemplo:

```
# gdisk -l /dev/sda
GPT fdisk (gdisk) version 1.0.1

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.
```



...

Disk /dev/sda: 976773168 sectors, 465.8 GiB  
Logical sector size: 512 bytes  
Disk identifier (GUID): 4F89526A-5F1B-4BF9-8CD8-2ABAD827386D  
Partition table holds up to 128 entries  
First usable sector is 34, last usable sector is 976773134  
Partitions will be aligned on 2048-sector boundaries  
Total free space is 8550 sectors (4.2 MiB)

Number	Start (sector)	End (sector)	Size	Code	Name
1	2048	1026047	500.0 MiB	EF00	EFI System Partition
2	1026048	1107967	40.0 MiB	FFFF	Basic data partition
3	1107968	1370111	128.0 MiB	0C01	Microsoft reserved ...
4	1370112	2373631	490.0 MiB	2700	Basic data partition
5	2373632	173729791	81.7 GiB	0700	Basic data partition
6	956454912	976766598	9.7 GiB	2700	Microsoft recovery ...
7	235169792	956454911	343.9 GiB	8300	
8	173729792	235169791	29.3 GiB	8300	usr

sergio ~ I/O size (minimum/optimal): 512 bytes / 512 bytes

Los comandos principales son:

<b>b</b>	Hacer una copia de seguridad de la GPT a un archivo
<b>c</b>	Cambiar el nombre de una partición
<b>d</b>	Borrar una partición
<b>i</b>	Mostrar información detallada en una partición
<b>l</b>	Mostrar un listado de tipos de particiones conocidas
<b>n</b>	Agregar una nueva partición
<b>o</b>	Crear una nueva tabla de particiones GUID (GPT) vacía
<b>p</b>	Imprimir la tabla de particiones
<b>q</b>	Salir sin guardar los cambios
<b>r</b>	Opciones de recuperación y transformación (expertos solamente)
<b>s</b>	Ordenar particiones
<b>t</b>	Cambiar el código del tipo de una partición
<b>v</b>	Verificar el disco
<b>w</b>	Guardar tabla en el disco
<b>x</b>	Funcionalidad extra (expertos solamente)
<b>?</b>	Imprimir este menú



En el siguiente ejemplo, el disco usa MBR pero tiene una GPT no válida:

```
# gdisk -l /dev/sda | head
GPT fdisk (gdisk) version 0.8.6
```

Partition table scan:

MBR: MBR only  
BSD: not present  
APM: not present  
GPT: not present

```
*****
Found invalid GPT and valid MBR; converting MBR to GPT format.
*****
```

Warning! Secondary partition table overlaps the last partition by  
33 blocks!  
You will need to delete this partition or resize it in another utility.



Si bien el comando **gdisk** tiene la capacidad para pasar de MBR a GPT, hay ciertos casos como este, en el que es necesario **borrar o cambiar el tamaño de alguna partición** (en particular la última).

La herramienta **parted** soporta tanto MBR como GPT y ofrece muchas más funcionalidades que **fdisk**. Para listar la tabla de particiones se debe hacer lo que figura en el código debajo:

```
# parted /dev/sda print
```

```
Model: ATA VBOX HARDDISK (scsi)
```

```
Disk /dev/sda: 8590MB
```

```
Sector size (logical/physical): 512B/512B
```

```
Partition Table: msdos
```

```
Disk Flags:
```

Numero	Inicio	Fin	Tamaño	Typo	Sistema de ficheros	Banderas
1	1049kB	525MB	524MB	primary	xf	arranque
2	525MB	8590MB	8065MB	primary	lvm	

En el siguiente ejemplo, el disco tiene GPT:

```
# parted /dev/sda print
```

```
Model: ATA WDC WD5000LPVX-7 (scsi)
```

```
Disk /dev/sda: 500GB
```

```
Sector size (logical/physical): 512B/4096B
```

```
Partition Table: gpt
```

```
Disk Flags:
```

Numero	Inicio	Fin	Tamaño	Sistema de ficheros	Nombre	Banderas
1	1049kB	525MB	524MB	fat32	EFI System Partition	arranque, esp
2	525MB	567MB	41,9MB	fat32	Basic data partition	oculta
3	567MB	701MB	134MB		Microsoft reserved partition	msftres
4	701MB	1215MB	514MB	ntfs	Basic data partition	oculta, diag
5	1215MB	88,9GB	87,7GB	ntfs	Basic data partition	msftdata
8	88,9GB	120GB	31,5GB	ext4	usr	
7	120GB	490GB	369GB	xf		
6	490GB	500GB	10,4GB	ntfs	Microsoft recovery partition	oculta, diag



Para **crear un disco con formato gpt** se debe ejecutar el siguiente comando:

```
(parted) mklabel gpt
```

Tener en cuenta que este comando si bien no borra los datos, quedarían inaccesibles, ya que **parted** elimina la tabla de particiones MBR que pudiera existir antes.



## Resguardando mi MBR y GPT

Copiar el primer sector del disco. Es el que contiene el MBR:

```
# dd if=/dev/sda of=/tmp/mbr.bkp bs=512 count=1
```

Para poder recuperarlo:

```
# dd if=/tmp/mbr.bkp of=/dev/sda bs=512 count=1
```

Para respaldar un disco con GPT:

```
# sgdisk -b gptbackup.img /dev/sda
```

Para recuperar la tabla de particiones a partir del backup:

```
#sgdisk -l /dev/sda
```



**¡Sigamos  
trabajando!**