

# *Implantación de sistemas operativos*

---

UT6

Sistemas Operativos libres  
y propietarios

--- Particiones---



# Índice

---

- Concepto de partición.
- Tipos de particiones
  - ▶ Primaria
  - ▶ Lógica
  - ▶ Extendida
- Programas para crear particiones

División lógica desde el  
pto vista de los SO

# 1. ¿Qué es una partición?

---

- Es el nombre genérico que recibe cada división presente en una sola unidad física de almacenamiento de datos.
- Toda partición tiene su propio sistema de archivos (formato);
- Generalmente, casi cualquier sistema operativo interpreta, utiliza y manipula cada partición como un disco físico independiente, a pesar de que dichas particiones estén en un solo disco físico.


# Tipos de particiones

---

- **Primaria.**

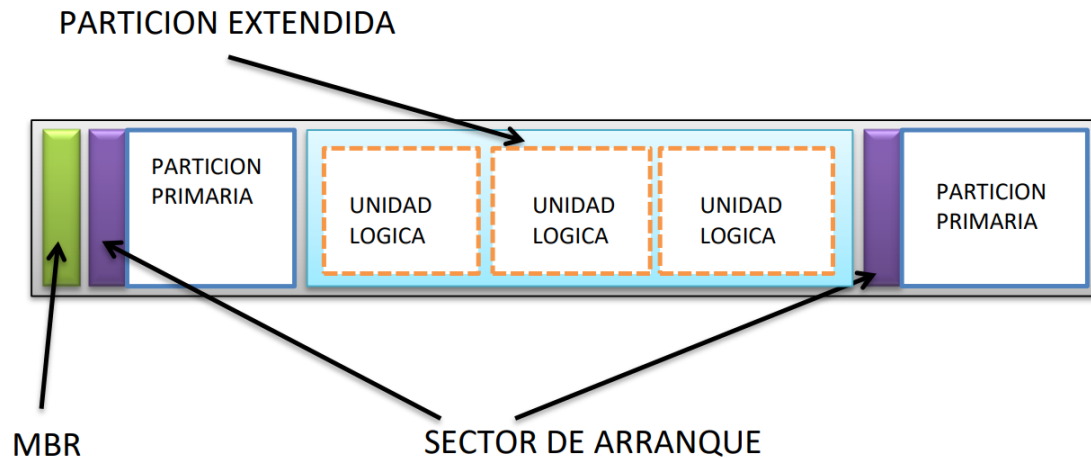
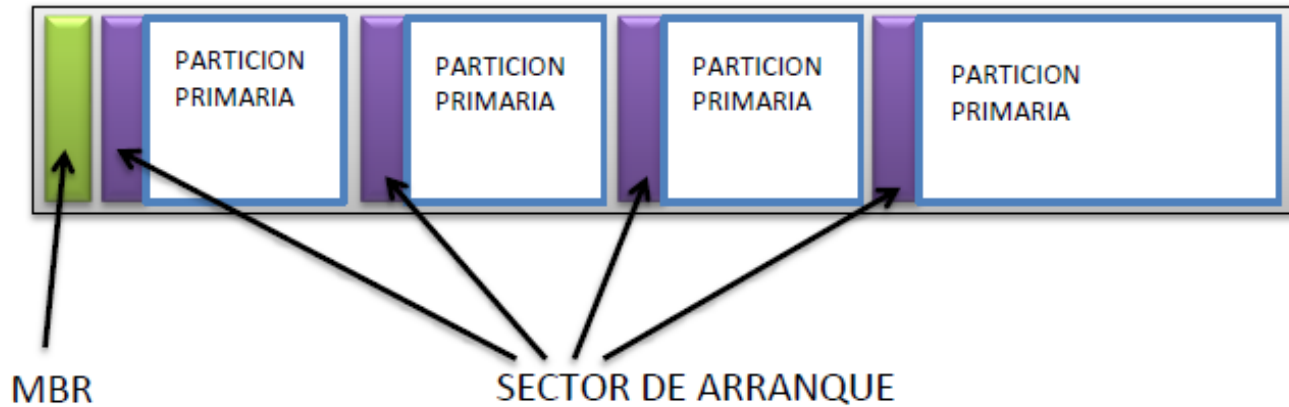
- **Extendida.**

- **Lógica.**

 **Nota:** Las particiones son independientes del sistema de archivos de una partición (FAT, ext3, NTFS, etc.)

# Sistemas basados en MBR

---



# Partición Primaria

---

- Son las divisiones crudas o primarias del disco.
- Solo puede haber 4 de éstas o 3 primarias y una extendida.
- Depende de una tabla de particiones.
- Un disco físico completamente formateado consiste, en realidad, de una partición primaria que ocupa todo el espacio del disco y posee un sistema de archivos.
- Este tipo de particiones, prácticamente cualquier sistema operativo puede detectarlas y asignarles una unidad, siempre y cuando el sistema operativo reconozca su formato (sistema de archivos)

# Partición extendida

---

- Es otro tipo de partición que actúa como una partición primaria;
- Sirve para contener infinidad de unidades lógicas en su interior.
- Fue ideada para romper la limitación de 4 particiones primarias en un solo disco físico.
- Solo puede existir una partición de este tipo por disco.
- Solo sirve para contener particiones lógicas.
- Por lo tanto, es el único tipo de partición que no soporta un sistema de archivos directamente.

# Partición lógica

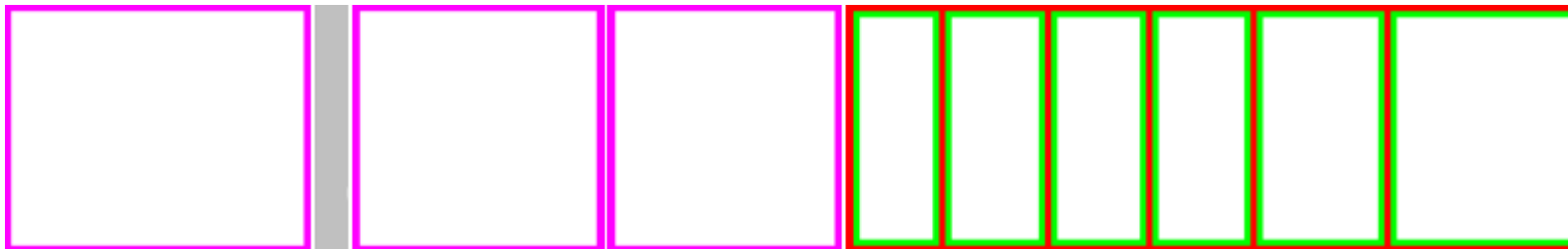
---

- Ocupa una porción de la partición extendida o la totalidad de la misma, la cual se ha formateado con un tipo específico de sistema de archivos (FAT32, NTFS, EXT2,...) y se le ha asignado una unidad.
- Inicialmente podía haber un máximo de 32 particiones lógicas en una partición extendida. Hoy alcanza hasta 128.



# Ejemplo

---



- Representación gráfica de un disco particionado.
- Cada recuadro blanco representa algún sistema de archivos vacío.
- Los espacios en gris representan los espacios sin particionar del disco.
- Las particiones rodeadas por líneas moradas o violetas representan las particiones primarias.
- Las particiones rodeadas por bordes rojos representan la partición extendida (que es un tipo de partición primaria);
- En su interior, se encuentran las particiones lógicas, rodeadas por los bordes de color verde.

# Observación sistemas basados en Unix

---

- Generalmente se usan hasta con 3 particiones:



/

Principal



/Home

Configuración usuarios



/swap

Memoria virtual temporal

- En linux:



/



Swap

# Identificación de particiones en Windows

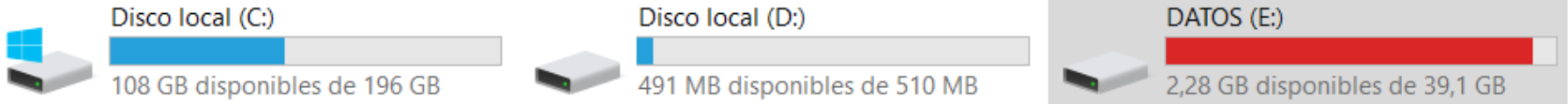
---

- Con una letra seguida por un signo de doble punto y la barra
- Ejemplo: C:\

# Identificación de particiones en Windows

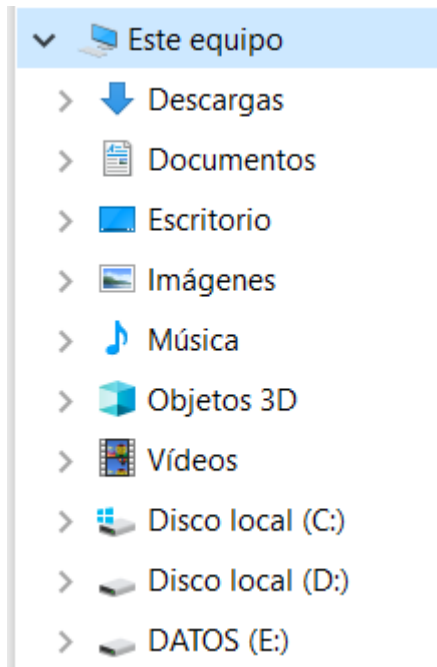
---

## Dispositivos y unidades (3)



# Carpetas Windows

---



## ■ Archivos de programa

Carpetas que contiene las aplicaciones

## ■ Usuarios

Carpetas personales de los usuarios

## ■ Windows

Carpetas del sistema

- Archivos de programa
- Archivos de programa (x86)
- carpeta
- Intel
- log
- PerfLogs
- Usuarios
- Windows

# Identificación de particiones en Linux

---

- La barra invertida
- Ejemplo: /

# Directorios en linux

---

```
|__ bin
|__ dev
|__ etc
|   |__ X11
|   |__ httpd
|   |__ mail
|   |__ ppp
|   |__ sysconfig
|__ sbin
|__ home
|   |__ shrek
|   |__ fiona
|   |__ burro
|__ lib
|__ proc
|__ tmp
|__ usr
|   |__ bin
|   |__ local
|   |__ X11R6
|   |__ etc
|   |__ include
|   |__ lib
|   |__ man
|   |__ src
|__ var
|   |__ admin
|   |__ spool
|   |__ run
|   |__ log
```

# Directorios de linux (1/3)

---

- bin Comandos y programas ejecutables
- sbin Comandos de superusuario
- boot Archivos necesarios para el arranque
- dev dispositivos de nuestro ordenador
  - ▶ cdrom Enlace al cdrom por defecto
  - ▶ sd\* Discos conectados via USB, externos
  - ▶ hd\* Discos conectados via IDE, internos
  - ▶ tty\* Terminales



# Ficheros de linux (2/3)

---

- **etc** configuración, servicios y programas
- **home** cuentas de usuario
  - ▶ **usuario** cuenta de un usuario
- **lib** librerías del sistema
- **media y mnt** punto para montar sistemas de archivos locales y removibles
- **root** cuenta del supervisor
- **tmp** archivos temporales

# Ficheros de linux (3/3)

---

- **proc**      sistema de archivos virtual con información de los procesos, hardware y el kernel
- **usr**      archivos de los usuarios y aplicaciones de usuarios
  - ▶ **local**      archivos de usuario externos a la distribución
  - ▶ **share**      archivos compartidos entre las aplicaciones
- **var**      información variable, logs, correo local

# Programas para particionar

---

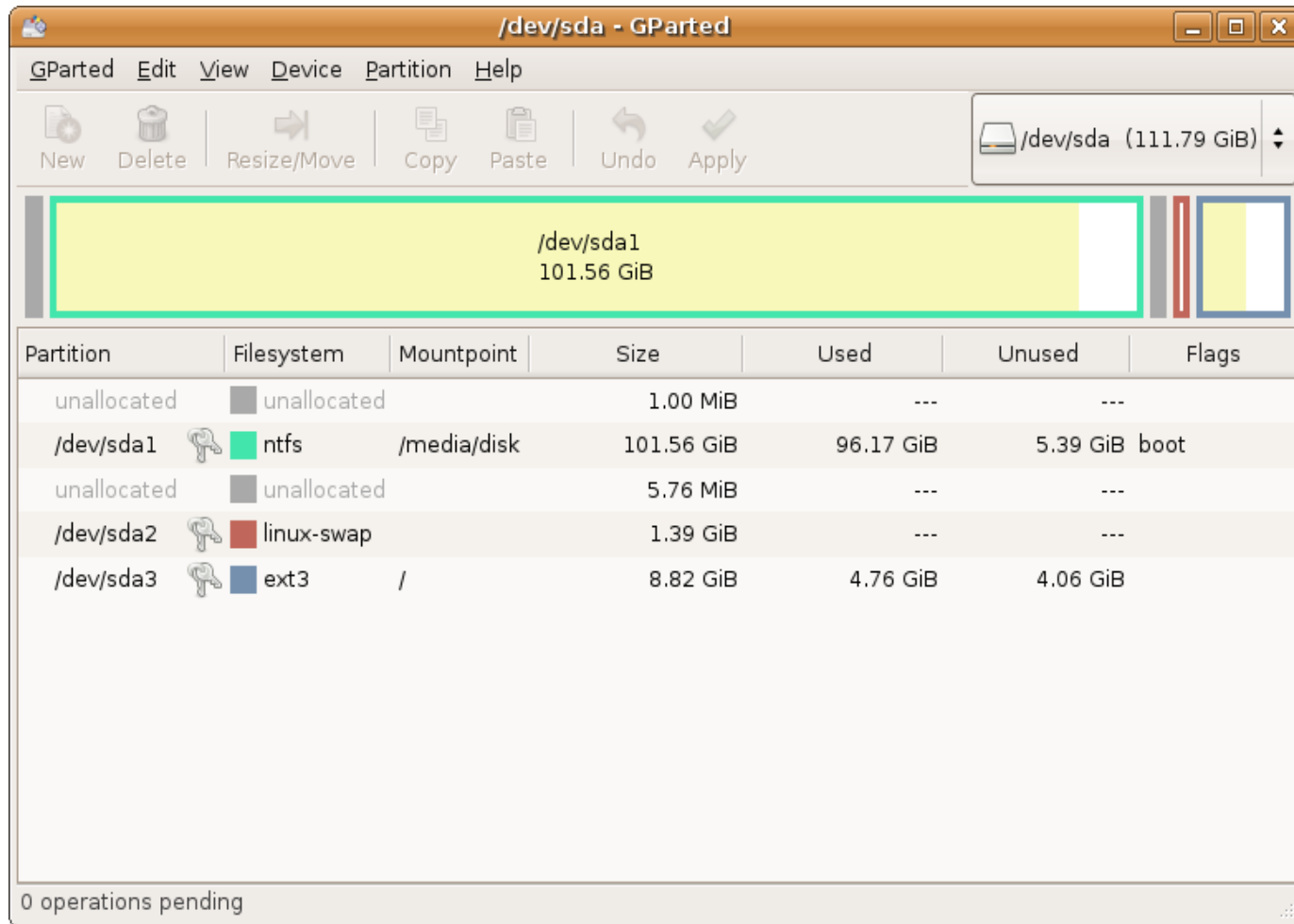
- Windows

- ▶ Antiguos `fdisk`

- ▶ Modernos `diskmgmt.msc, diskpart.msc`

- Linux `fdisk, qtparted, parted, etc.`

# Ejemplo Gparted



**dev**, significa dispositivo (device)

**sd**, se refiere al tipo de disco (disco sata en este caso)

la **letra** a indica que es la unidad a, es decir, solo hay un disco.

los **números**, significan: del 1 al 3 particiones primarias, el 4 partición extendida, del 5 en adelante unidades lógicas

# DiskPart

---

```
DISKPART> list disk
```

Núm Disco	Estado	Tamaño	Disp	Din	Gpt
Disco 0	En línea	60 GB	0 B		
* Disco 1	En línea	2048 MB	2046 MB		

```
DISKPART> select disk 1
```

El disco 1 es ahora el disco seleccionado.

```
DISKPART> create partition primary size=300
```

DiskPart ha creado satisfactoriamente la partición especificada.

```
DISKPART> list partition
```

Núm Partición	Tipo	Tamaño	Desplazamiento
* Partición 1	Principal	300 MB	64 KB

```
DISKPART> select partition 1
```

La partición 1 es ahora la partición seleccionada.

```
DISKPART> format
```

100 por ciento completado

DiskPart formateó el volumen correctamente.

```
DISKPART> assign letter=f
```

DiskPart asignó correctamente una letra de unidad o punto de montaje.

# Consideraciones Particiones

---

- Separación entre programas y datos.
- Copias backup de datos, evitando perdida de información.
- Necesidad de Linux de usar más de una partición.
- Sistemas de archivos antiguos que **no** permiten manejar discos de gran tamaño.
- La posibilidad de tener 2 sistemas de archivos diferentes y SO diferentes.
- Permite superar el límite de particiones primarias.

# Ejemplo de una tabla de particiones

Primer sector físico del disco duro (pista cero)			
440 bytes Código del gestor de arranque			
512 bytes	84 bytes	16 bytes	Primera partición
			1 byte Marca de arranque si el bit 7 está activo es una partición de arranque, los otros bits deben ser ceros.
			3 bytes CHS de inicio
			1 byte Tipo de partición
			3 bytes CHS final
			4 bytes LBA
			4 bytes Tamaño en sectores
		16 bytes	Segunda partición
			1 byte Marca de arranque si el bit 7 está activo es una partición de arranque, los otros bits deben ser ceros.
			3 bytes CHS de inicio
			1 byte Tipo de partición
			3 bytes CHS final
			4 bytes LBA
			4 bytes Tamaño en sectores
		16 bytes	Tercera partición
			1 byte Marca de arranque si el bit 7 está activo es una partición de arranque, los otros bits deben ser ceros.
			3 bytes CHS de inicio
			1 byte Tipo de partición
			3 bytes CHS final
			4 bytes LBA
			4 bytes Tamaño en sectores
	16 bytes	Cuarta partición	
		1 byte Marca de arranque si el bit 7 está activo es una partición de arranque, los otros bits deben ser ceros.	
		3 bytes CHS de inicio	
		1 byte Tipo de partición	
		3 bytes CHS final	
		4 bytes LBA	
		4 bytes Tamaño en sectores	
2 bytes	Firma de unidad arrancable ("55 AA" en hexadecimal)		

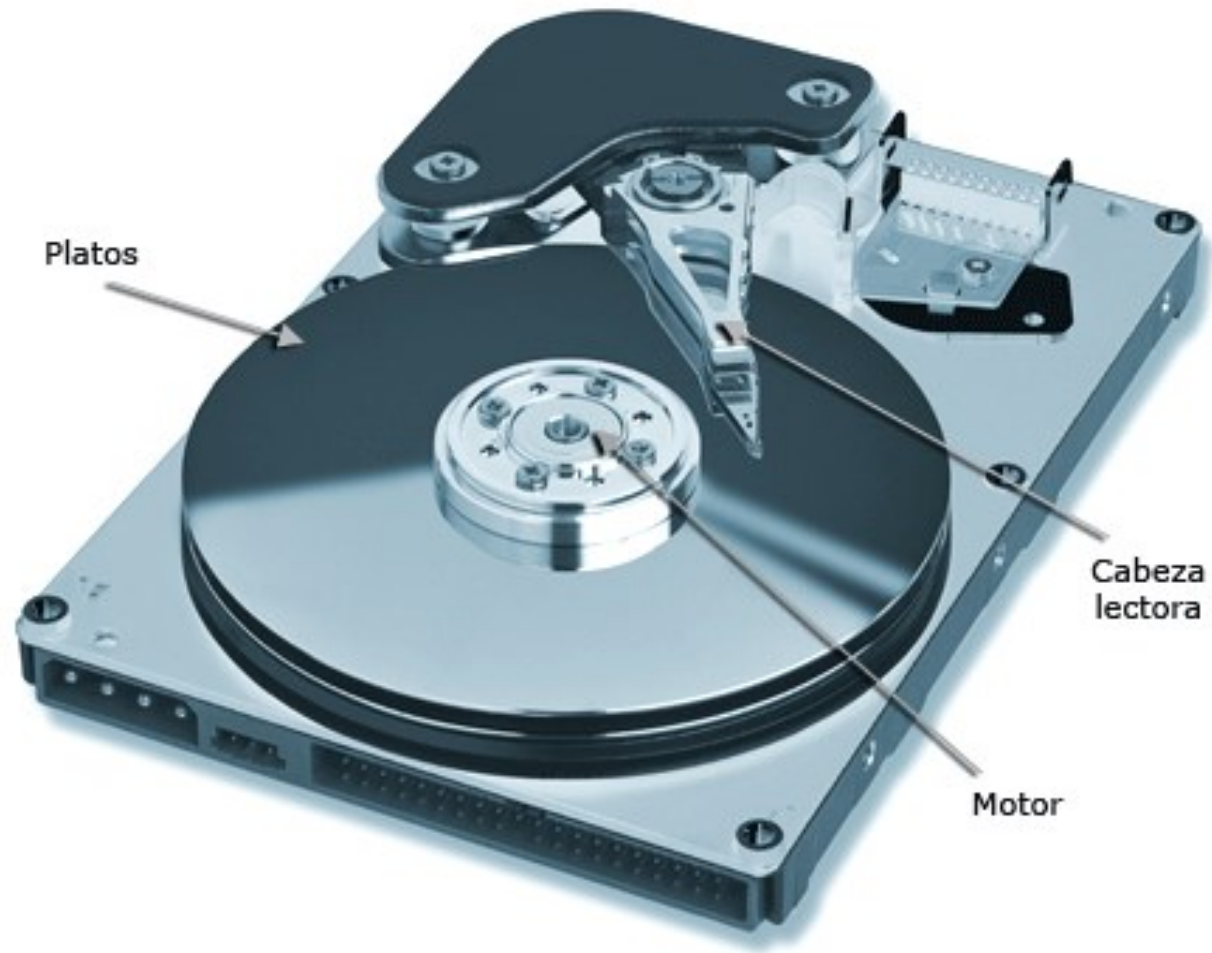
# Información adicional de interés

---

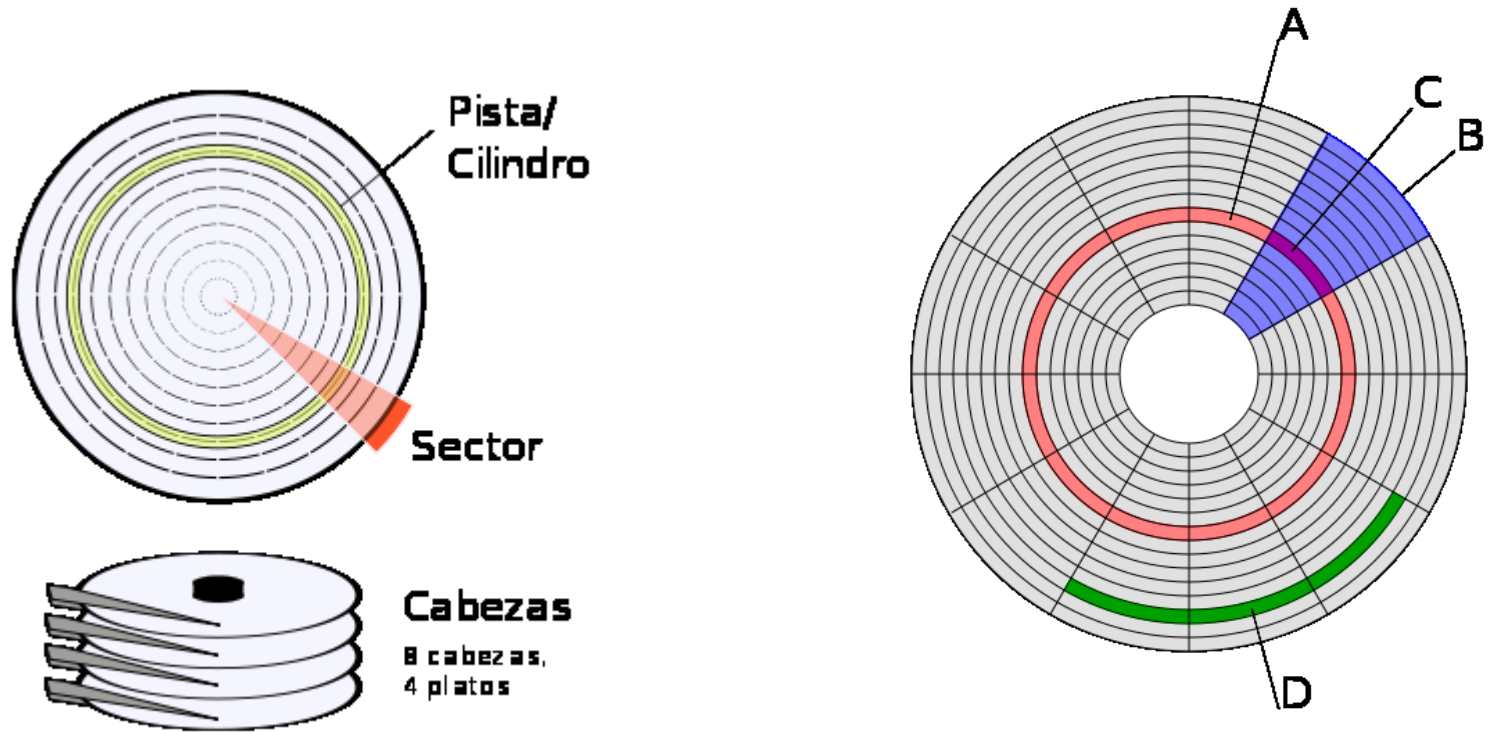
- Hay varios conceptos para referirse a zonas del disco:
  - ▶ **Plato:** cada uno de los discos que hay dentro del *disco duro*.
  - ▶ **Cara:** cada uno de los dos lados de un *plato*.
  - ▶ **Cabeza:** número de cabezales.
  - ▶ **Pista:** una circunferencia dentro de una *cara*; la *pista 0* está en el borde exterior.
  - ▶ **Cilindro:** conjunto de varias *pistas*; son todas las circunferencias que están alineadas verticalmente (una de cada *cara*).
  - ▶ **Sector:** cada una de las divisiones de una pista. El tamaño del sector no es fijo, siendo el estándar actual **512 bytes**, a 4 KB. Antiguamente el número de sectores por pista **era fijo**, lo cual desaprovechaba el espacio significativamente, ya que en las pistas exteriores pueden almacenarse más sectores que en las interiores. Así, apareció la tecnología **ZBR (grabación de bits por zonas)** que aumenta el número de sectores en las pistas exteriores, y utiliza más eficientemente el disco duro.



# Estructura Física del Disco



# Información adicional de interés



Estructura física desde el pto vista de los SO

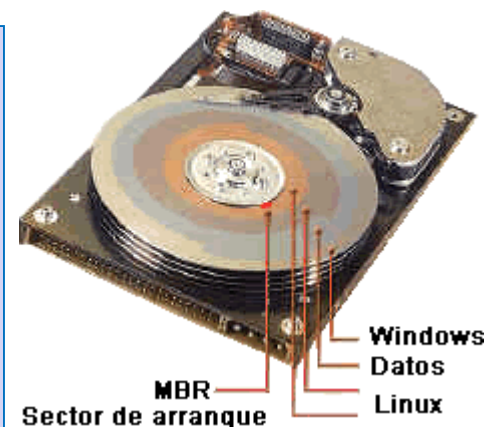
# Información adicional de interés

---

- El primer sistema de direccionamiento que se usó fue el **CHS** (**cilindro-cabeza-sector**), ya que con estos tres valores se puede situar un dato cualquiera del disco.
- Más adelante se creó otro sistema más sencillo: **LBA** (**direccionamiento lógico de bloques**), que consiste en dividir el disco entero en *sectores* y asignar a cada uno un único número. Éste es el que actualmente se usa.

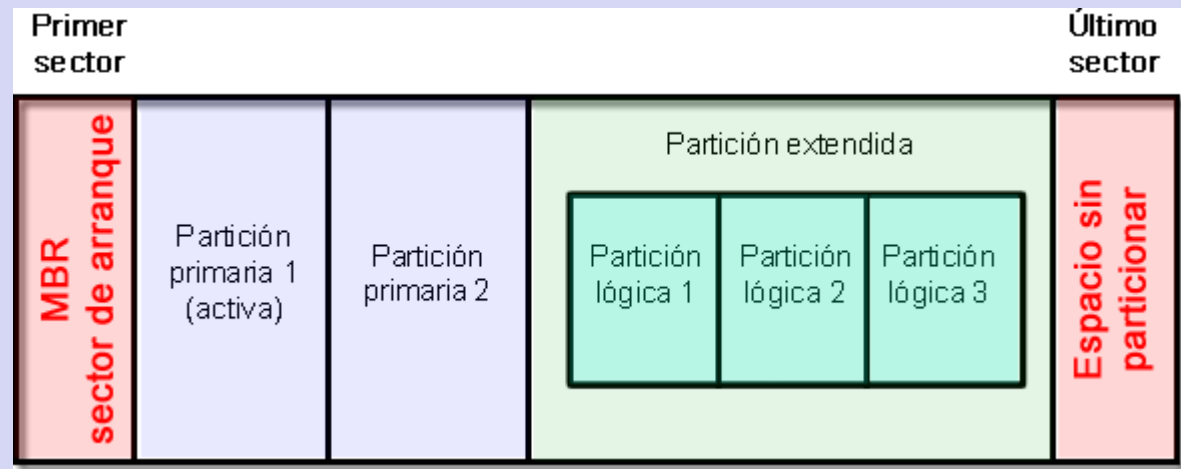
# Particiones y gestores de arranque

- El primer sector del disco duro se llama sector de arranque o **MBR (Master Boot Record)**
- En ese sector se guarda información del tamaño de las particiones y de cual es la partición activa, es decir, la de arranque.



# Partición activa

- Es donde se dirige el MBR para proceder al arranque de nuestra máquina, y puede contener un pequeño gestor de arranque cuando hay más de un SO



# MBR

---

- Soporta la mayoría de los SO
- Estos discos soportan la mayoría de las BIOS estándar
- Cada partición soporta 2TB
- Estos discos soportan 4 particiones primarias ó 3 primarias y una extendida, que puede tener actualmente hasta 128 volúmenes lógicos

# GUID (GPT)

---

- Soporta SO de tipo Servidor y SO con arquitecturas 64bits
- Estos discos usan la Interfaz Extensible del Firmware (EFI)
- Cada partición soporta 256TB
- Estos discos soportan hasta 128 particiones primarias

# Ejemplo de EFI





# Ejemplo de BIOS



# Contenido del MBR

PROGRAMA DEL MBR

TABLA DE  
PARTICIONES

Dirección.	Contenido.	Tipo.
+000h	Programa MBR.	445 Bytes.
+1BEh	1º entrada de la tabla de particiones	16 Bytes
+1CEh	2º entrada de la tabla de particiones	16 Bytes
+1DEh	3º entrada de la tabla de particiones	16 Bytes
+1EEh	4º entrada de la tabla de particiones	16 Bytes
+1FEh	Identificación (AA55h)	2 Bytes

*Contenido del Master Boot Record o MBR.*

*Longitud = 200h = 512 Bytes.*

*El código AA55h marca este sector como ejecutable.*

# Cada partición (16 Bytes)

Dirección.	Contenido.	Tipo.
+00h	Estado de la partición: 00h – Inactiva 80h – arranque (activa)	1 Byte
+01h	Cabeza de lectura / escritura donde comienza la partición.	1 Byte
+02h	Sector y cilindro donde comienza la partición.	2 Bytes
+04h	Tipo de partición: 00h – Libre 01h – DOS con la vieja FAT de 12 bits. 02h – XENIX 03h – XENIX 04h – DOS con FAT 16 05h – Partición extendida. 06h – Partición DOS > 32 Megas. 0Bh – Windows FAT32 0Ch – Windows FAT 32 LBA 0Eh – VFAT 16h – Hidden FAT 16 (Oculta) 63h – Unix 65h – Novell Netware Etc.....	1 Byte
+05h	Cabeza de lectura / escritura donde termina la partición.	1 Byte
+06h	Sector y cilindro donde termina la partición.	2 Bytes
+08h	Dirección del primer sector de la partición. (Sector de arranque).	4 Bytes
+0Ch	Número de sectores en esta partición.	4 Bytes

*Contenido de cada una de las 4 entradas de la tabla de particiones.  
Longitud = 10h = 16 Bytes.*

# Información de dicha tabla

---

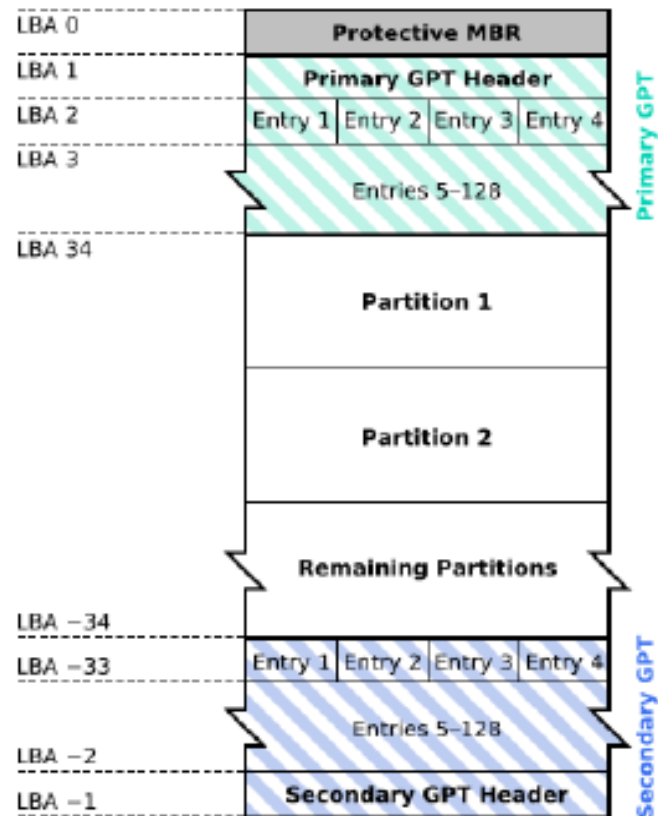
- Vemos como el 1º campo se usa para indicar si esta partición es la activa o no.
- El 2º y 3º campo se usan para indicar el cilindro, sector y cabeza donde comienza la partición.
- El 4º campo se usa para almacenar el tipo de la partición, aquí se indica que sistema operativo está instalado en la partición, si dicha partición esta oculta o no, etc.
- El 5º y 6º campo se usan para indicar el cilindro, sector y cabeza donde termina la partición.
- El 7º campo indica la dirección del primer sector de la partición (el sector de arranque) para que el POST pueda pasarle el control. Este sector siempre es el 1º sector de la partición, pero aquí indicamos su valor director (nº de sector) y no la combinación cilindro, sector y cabeza. Esto se hace para que el acceso al sector de arranque sea más rápido, y para evitar posibles errores en la carga del sistema.
- El 8º campo se usa para almacenar el número total de sectores que existen en la partición. Es un campo que se usa para comprobar que los datos de la partición son correctos.

# GPT

---

- Usa modo de direccionamiento lógico (**LBA, logical block addressing**) en lugar del modelo cilindro-cabeza-sector (CHS)
  - ▶ En el **LBA 0** o bloque lógico 0 → se almacena la información de MBR heredado,
  - ▶ En el **LBA 1** → se almacena la cabecera GPT
  - ▶ En los bloques sucesivos → se almacena la tabla de particiones
- Los SO Windows 64-bits,
  - ▶ 16.384 bytes, o lo que es lo mismo, 32 sectores, están reservados para la GPT, dejando el bloque LBA 34 como el primer sector usable del disco.

## GUID Partition Table Scheme



# Problemas MBR

---

- En MBR sólo pueden ser definidas 4 particiones primarias o 3 primarias + 1 partición extendida (con un número arbitrario de particiones lógicas dentro de la partición extendida).
- En MBR dentro de la partición extendida, los metadatos de las particiones lógicas se almacenan en una estructura de lista enlazada. Si un enlace se pierde, todas las particiones lógicas existentes, después de los metadatos, se pierden. .
- MBR almacena la información del sector de la partición con valores LBA de 32 bits. Esta longitud de LBA junto con los 512 byte del tamaño del sector (más comúnmente utilizados) limita el tamaño máximo manejable del disco hasta 2 TB.
- MBR sólo admite 1 byte para códigos de tipo de partición, lo que conlleva muchas colisiones

# Ventajas GPT

---

- Utiliza GUID (UUID) para identificar los tipos de particiones. Sin colisiones.
- Proporciona un GUID único de disco y un GUID único de partición para cada partición. Un buen sistema de archivos independiente referenciando a las particiones y discos.
- Número arbitrario de particiones (depende del espacio asignado por la tabla de particiones). No hay necesidad de particiones extendidas y lógicas. Por defecto, la tabla GPT contiene espacio para la definición de 128 particiones. Sin embargo, si el usuario desea definir más particiones, se puede asignar más espacio (de momento solo en Linux).
- Almacena una copia de seguridad del encabezado y de la tabla de particiones al final del disco que ayuda en la recuperación en el caso de que los primeros están dañados.



# Ventajas GPT

---

- Utiliza 64-bit LBA para almacenar números del Sector - tamaño máximo del disco manejable es de 2 Zeta Bytes.
- Almacena una copia de seguridad del encabezado y de la tabla de particiones al final del disco que ayuda en la recuperación en el caso de que los primeros están dañados.
- Código de reparación de errores CRC32 para detectar errores y daños de la cabecera y en la tabla de particiones.

# Arranque Windows 8/10

---

1. Se carga y ejecuta el **POST**
2. Se carga el **MBR** del disco duro (si es la opción elegida en la BIOS)
3. Se carga el **sector de arranque** de la partición primaria activa
4. Se carga el programa **bootmgr**.
5. bootmgr ajusta el procesador para trabajar a 32 bits o 64 bits.
6. bootmgr lee la base de datos **BCD** y muestra un menú si es necesario
7. El usuario selecciona un sistema operativo del menú, o se carga por defecto uno de ellos
8. bootmgr carga **winload.exe**.
9. Winload.exe carga **NTOSKRNL.EXE** (Núcleo del sistema operativo o Kernel).
10. NTOSKRNL.EXE lee el **registro** de Windows, y procede a ir cargando el sistema completo.

# Opc. a considerar arranque UEFI

---

- **SECURE BOOT.**

- ▶ El firmware del sistema comprueba la firma digital del sector de arranque, para comprobar si es de un sistema reconocido, y si se ha producido algún tipo de modificación sobre el mismo

- **TRUSTED BOOT.**

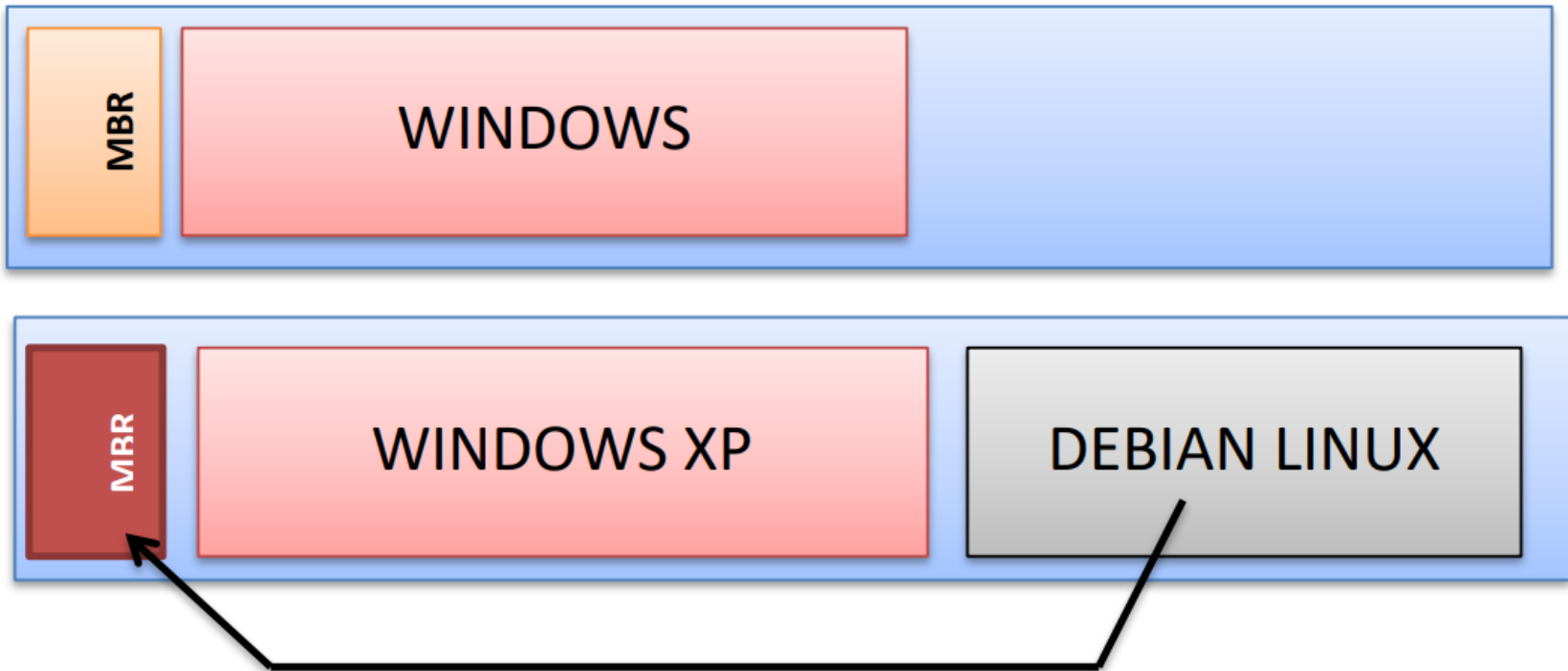
- ▶ El código de carga (bootloader) verifica el firmado del kernel de Windows antes de cargarlo.
- ▶ A su vez, el kernel de Windows verifica todos los componentes de Windows que se van cargando, incluyendo los drivers de dispositivo de la propia Microsoft que se cargan en el arranque.

- **FAST STARTUP**

- ▶ Windows 10 cierra totalmente la sesión del usuario y la vuelve a cargar en cada inicio, sin embargo, la sesión del kernel la hiberna, leyendo todo su estado en la RAM y grabándolo directamente en el disco duro.
- ▶ Esto puede ocasionar problemas de corrupción, debido al bloqueo del disco duro
  - ▶ Si se tiene varios SO instalados en la misma máquina.
  - ▶ Si se intenta usar el disco en otro Pc
  - ▶ Problemas al intentar acceder a la BIOS/UEFI
    - Solución: shutdown -s -t 0 / powercfg /hibernate off

# Instalaciones

---



## Particiones predeterminadas del disco de UEFI

Disco 0

Partición de  
sistema EFI

SISTEMA

MSR



MSR

Partición principal

WINDOWS

## Particiones predeterminadas del disco de BIOS

Disco 0

Partición  
principal

SISTEMA

Partición principal

WINDOWS