

CURSOS
INTERSEMESTRALES



PROTECO

Particiones y Volumen logico



¿Qué es particionado?

- Dividir un disco duro en múltiples sub discos.
- Cada región será tratada de forma independiente
- Dos tipos:
 - Lógico
 - Físico (primario)





¿Por qué hacer particiones?

- Encapsular tu información bajo distintos sistemas de archivos (Administración)
- Da mayor eficiencia (A nivel de bloques) de tu memoria.
- Elegir el espacio que queremos dar a cierto sistema operativo.
- Separar información de SO de usuario.



Lifewire



¿Que es un sistema de archivos?

Parte del Sistema Operativo que :

Estructura, asigna y da acceso a la memoria



NTFS? FAT32? ext2?

FAT32

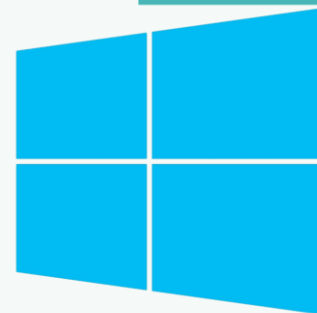
Tablas de asignación de Archivos

NTFS

Sistema de archivos de nueva tecnología
windows

ext2

Segundo de sistema de archivo



Tipos

- ext4

Es la última versión de la familia de sistemas de ficheros ext

Eficiencia (menor uso de CPU, mejoras en la velocidad de lectura y escritura) }

1024PB (PetaBytes).

Todos los directorios pueden estar en su propia partición.





Tipos de particionado

- ¿Que es la tabla de particiones?

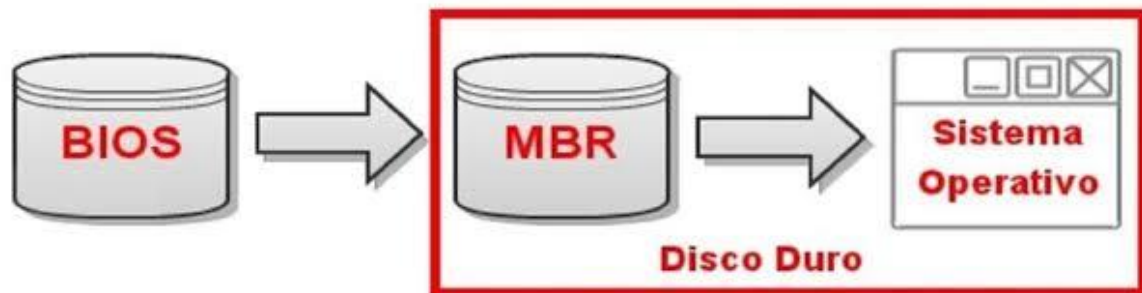
Contiene donde se encuentra cada partición.

Es lo primero que es leído en el sistema operativo

- El como esta hecho puede ser:

MBR (Master Boot Record) (Maximo 4 particiones primarias)

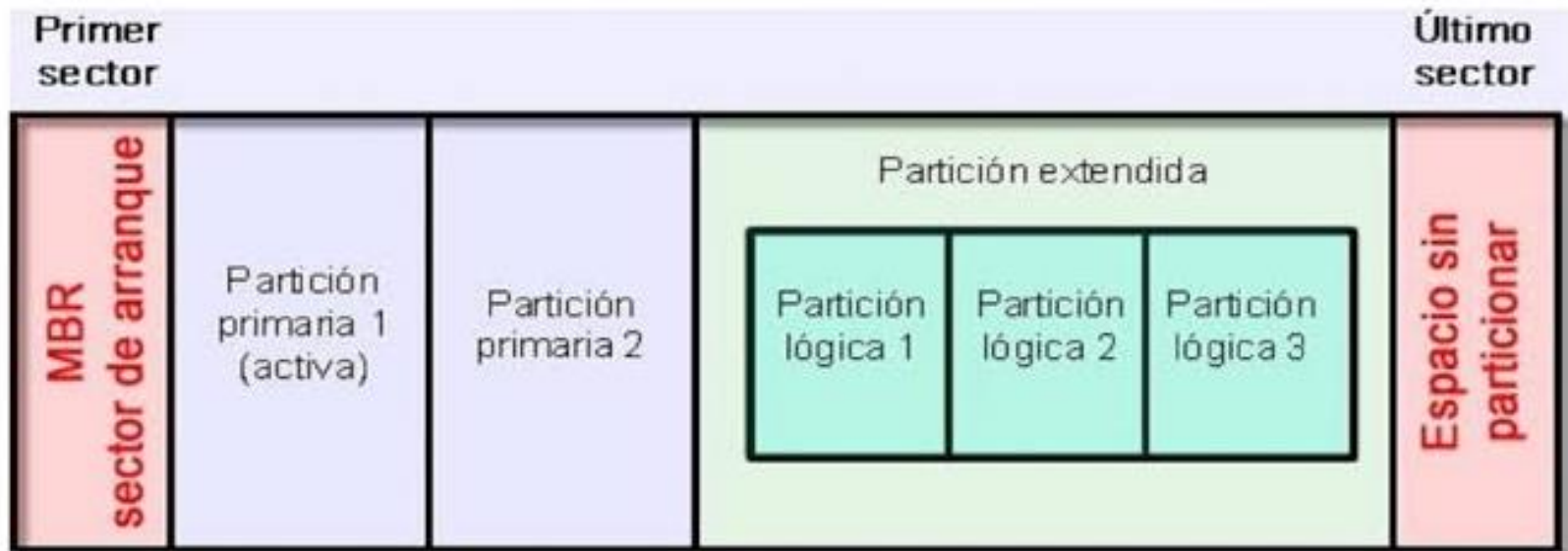
GPT (GlobalUID Partition Table) (Numero ilimitado de particiones)





Tipos de particionado

- Partición primaria: (MBR only)
- Partición extendida (Contiene las lógicas en MBR)
- Partición Lógica (MBR only)

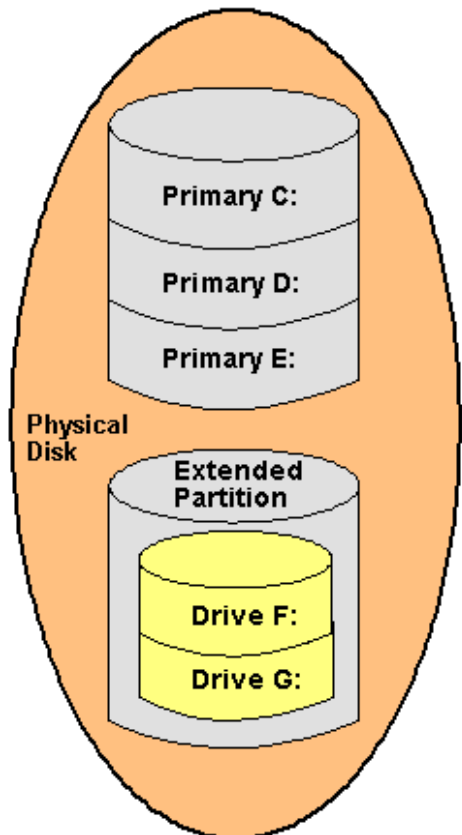




Tipos de particionado (MBR)

- Partición primaria: (MBR)
 - Contiene UN sistema de archivos.
 - Puede ser booteable.
 - Deben ser solo 4 por disco
- Partición extendida
 - Es única por disco.
 - Es un contenedor para las particiones lógicas
 - Ocupa el espacio de una PP.
- Partición lógica
 - Pueden ser múltiples
 - Contenidos dentro de la extendida

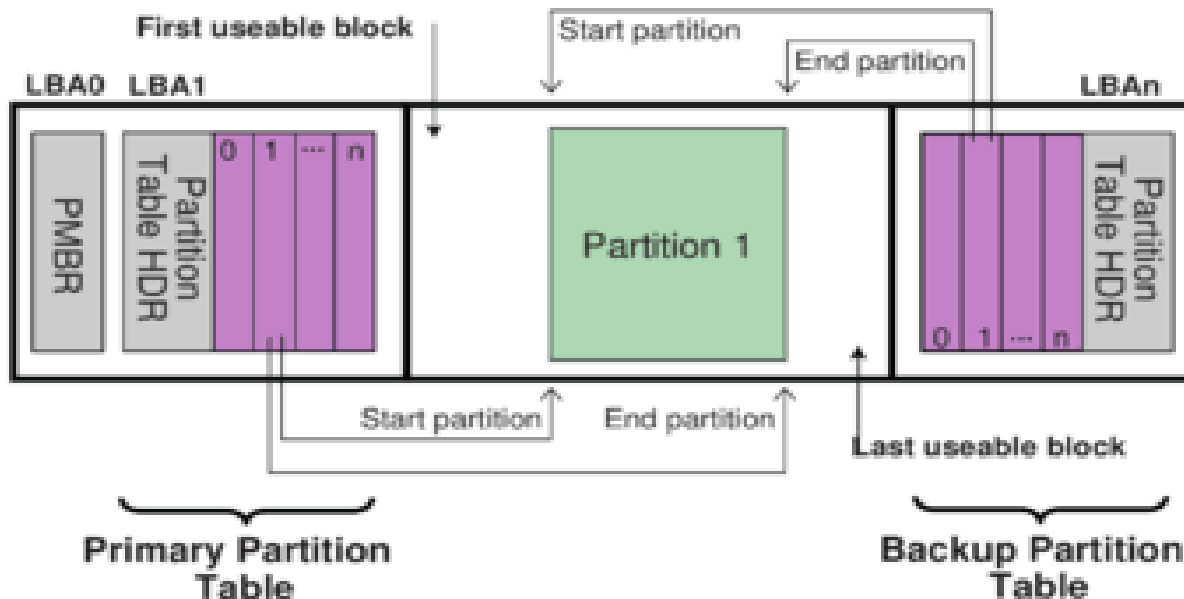
BIOS Firmware MBR Booting



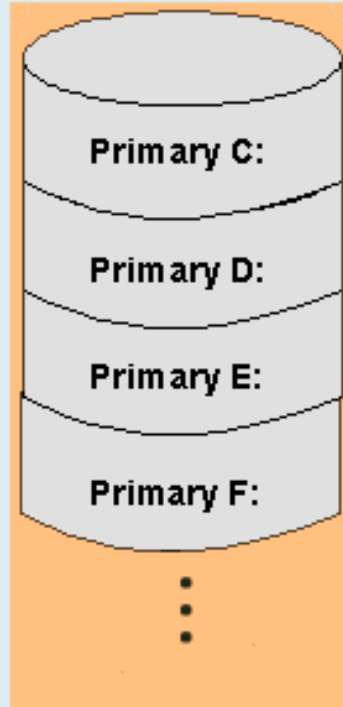


Tipos de particionado (GPT)

- Fue agregada junto con UEFI
- Utiliza identificadores únicos globales (GUID)
- Todas las particiones son equivalentes a las primarias de MBR
- Pueden usarse infinitas particiones y máximo trabaja con 2 Zbytes



GPT Disk





fdisk

Programa que nos permitirá modificar nuestra tabla de particiones (MBR o GPT)

```
root@campesino-VirtualBox:/home/campesino# fdisk -l
```

```
Disk /dev/sda: 18 GiB, 19287818240 bytes, 37671520 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: 0x30d555b9
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	37670911	37668864	18G	83	Linux

```
Disk /dev/sdb: 3.8 GiB, 4102028288 bytes, 8011774 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: 0x00054b78
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		128	8007807	8007680	3.8G	e	W95 FAT16 (LBA)





fdisk

```
root@campesino-VirtualBox:/home/campesino# fdisk /dev/sdb
```

```
Welcome to fdisk (util-linux 2.31.1).  
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.
```

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

```
Help:
```

```
DOS (MBR)
```

- a toggle a bootable flag
- b edit nested BSD disklabel
- c toggle the dos compatibility flag





fdisk

Notar que el tamaño del sector es de 512 bytes, por lo que al agregar 3000 sectores = $512 \times 3000 = 1,536,512$ bytes

```
Command (m for help): g
Created a new GPT disklabel (GUID: E588DBFC-1D27-2F43-92AA-44F34D656659).
The old dos signature will be removed by a write command.

Command (m for help): p

Disk /dev/sdb: 3.8 GiB, 4102028288 bytes, 8011774 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: gpt
Disk identifier: E588DBFC-1D27-2F43-92AA-44F34D656659

Command (m for help): n
Partition number (1-128, default 1):
First sector (2048-8011740, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-8011740, default 8011740): +300
0

Created a new partition 1 of type 'Linux filesystem' and of size 1.5 MiB.
Partition #1 contains a vfat signature.

Do you want to remove the signature? [Y]es/[N]o: Y

The signature will be removed by a write command.
```





fdisk

t -> Cambiar el tipo de una partición.

w -> Guardar las modificaciones

Ejercicio:

Crear 5 particiones

1 swap 10K

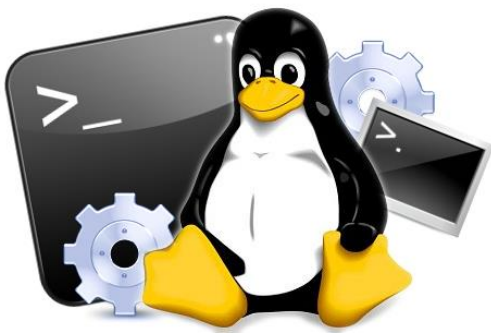
2 100M

3 1.5 M

4 250.5 K

Nota: Actualmente tenemos una librería llena de estantes, falta llenarlos de libros.





fdisk

Ejercicio:

Quemar todo y volverlo a una usb compatible con Windows.

¿Cuántas particiones?, ¿Qué File system?



PROTECO



fdisk

Disk /dev/sdb: 3.8 GiB, 4102028288 bytes, 8011774 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: gpt

Disk identifier: E588DBFC-1D27-2F43-92AA-44F34D656659

Device	Start	End	Sectors	Size	Type
/dev/sdb1	2048	5048	3001	1.5M	Linux filesystem
/dev/sdb2	5049	6049	1001	500.5K	Linux swap
/dev/sdb3	6144	6644	501	250.5K	Linux filesystem
/dev/sdb4	8192	8391	200	100K	Linux filesystem
/dev/sdb5	10240	30719	20480	10M	Linux filesystem

root@campesino-VirtualBox:/home/campesino#





mkfs

Make File system: Conjunto de herramientas que creará un sistema de archivos

De otro modo no podremos montarlo

```
root@campesino-VirtualBox:/home/campesino# mkfs
mkfs          mkfs.cramfs  mkfs.ext3     mkfs.fat
mkfs.bfs      mkfs.ext2    mkfs.ext4     mkfs.minix
```

```
root@campesino-VirtualBox:/home/campesino# mkfs.ext4 /dev/sdb5
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 10240 1k blocks and 2560 inodes
Filesystem UUID: f1c60809-e31e-4fe2-8e64-96055e58f0c3
Superblock backups stored on blocks:
    8193
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done
```





mount

Permite añadir o conectar a nuestro sistema de archivos otro sistema de archivos.

```
Model: JetFlash Transcend 4GB (scsi)
Disk /dev/sdb: 4102MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	2585kB	1537kB			
2	2585kB	3098kB	513kB			
3	3146kB	3402kB	257kB			
4	4194kB	4297kB	102kB			
5	5243kB	15.7MB	10.5MB	ext4		

```
root@campesino-VirtualBox:/mnt# sudo mount /dev/sdb5 /mnt/montado/
root@campesino-VirtualBox:/mnt# cd montado/
root@campesino-VirtualBox:/mnt/montado# ls
lost+found
```





mount

Para desmontar o separar el sistema de archivos nuevo, podemos usar umount

```
root@campesino-VirtualBox:/mnt# ls
montado
root@campesino-VirtualBox:/mnt# umount montado/
root@campesino-VirtualBox:/mnt# cd montado/
root@campesino-VirtualBox:/mnt/montado# ls
```





LVM

Logical Volume Manager:

Permite crear bloques virtuales (O dispositivos virtuales), independientemente de los discos físicos.
+Fácil de administrar y usar.

Disque Physique / partition :
PV(Physical Volume)

sdb1



sdc1



VG (Group Volume)

VG01

LV (Logical Volume)

LV01

LV02

LV03

Système de fichier
+
montage

EXT4



/

EXT3



/var

EXT4



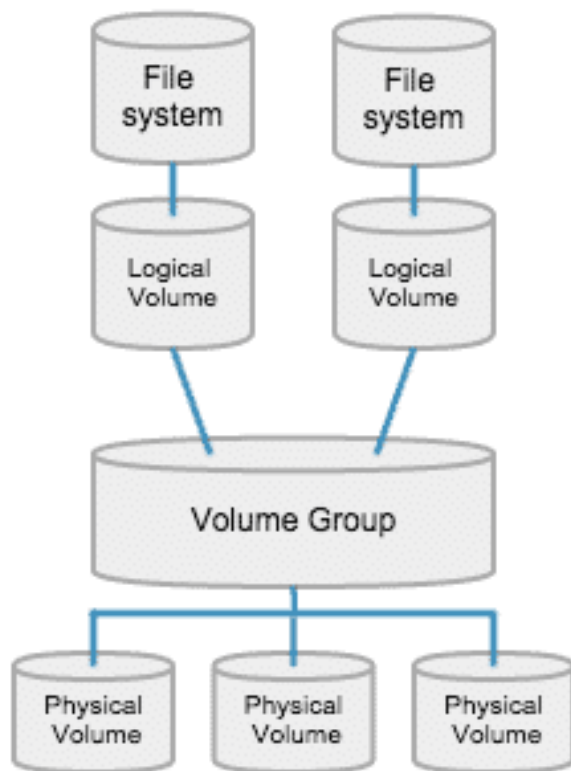
Volume1



LVM

Volume Group (VG) esta conformado por volúmenes físicos(PV) (dispositivos físicos) .

El VG se puede subdividir en distintos volúmenes lógicos (LV)





LVM

Ventajas:

- Usar múltiples discos como si fueran uno solo (grande).
- Modificar los LV a placer
- Migrar datos de forma fácil y rápida entre LV

Desventaja:

- No es soportado por Windows
- Mayor complejidad al crear nuestro sistema





LVM

¿Qué se necesita?

*Discos -> Particiones para los PV (Basta con una por disco) -> Crear PV (Puede y es recomendable sea una sola en todo el disco) -> Crear VG -> Añadir PV al VG
-> Crear LV sobre el VG -> Hacer uso de esto,

```
root@campesino-VirtualBox:/mnt# lvmdiskscan
/dev/loop1 [      <14.75 MiB]
/dev/sda1  [      17.96 GiB]
/dev/loop2 [      42.80 MiB]
/dev/loop3 [     <54.43 MiB]
/dev/loop4 [      88.48 MiB]
/dev/loop5 [       4.02 MiB]
/dev/loop6 [     149.90 MiB]
/dev/loop7 [      <3.65 MiB]
/dev/sdb1  [     <3.82 GiB]
0 disks
9 partitions
0 LVM physical volume whole disks
0 LVM physical volumes
```



LVM

¿Qué se necesita?

Crear PV (Puede y es recomendable sea una sola en todo el disco)

```
root@campesino-VirtualBox:/mnt# pvcreate /dev/sdb1
WARNING: vfat signature detected on /dev/sdb1 at offset 82. Wipe it? [y/n]: y
Wiping vfat signature on /dev/sdb1.
WARNING: vfat signature detected on /dev/sdb1 at offset 0. Wipe it? [y/n]: y
Wiping vfat signature on /dev/sdb1.
WARNING: vfat signature detected on /dev/sdb1 at offset 510. Wipe it? [y/n]: y
Wiping vfat signature on /dev/sdb1.
Physical volume "/dev/sdb1" successfully created.
```



PROTECO



LVM

Comprobando:

```
root@campesino-VirtualBox:/mnt# pvdisplay
"/dev/sdb1" is a new physical volume of "<3.82 GiB"
--- NEW Physical volume ---
PV Name                /dev/sdb1
VG Name
PV Size                <3.82 GiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                Umv22w-rJ21-3axp-enzS-odKI-asLt-1GbuGt

root@campesino-VirtualBox:/mnt# pvscan
PV /dev/sdb1            lvm2 [<3.82 GiB]
Total: 1 [<3.82 GiB] / in use: 0 [0   ] / in no VG: 1 [<3.82 GiB]
```



LVM

¿Qué se necesita?

Crear VG (En este caso es servidor y le debemos dar un PV inicial

```
root@campesino-VirtualBox:/mnt# vgcreate servidor /dev/sdb1
Volume group "servidor" successfully created
```

Añadir los distintos PV al VG:

```
root@campesino-VirtualBox:/mnt# vgextend servidor /dev/sda
```

```
root@campesino-VirtualBox:/mnt# vgdisplay
--- Volume group ---
VG Name                servidor
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No   1
VG Access              read/write
VG Status              resizable
```





LVM

-> Crear LV sobre el VG -> Hacer uso de esto,

```
# lvcreate -L <tamaño> <VG> -n <nombre_volLog>
```

```
root@campesino-VirtualBox:/mnt# lvcreate -l 50%FREE servidor -n Tareas
Logical volume "Tareas" created.
root@campesino-VirtualBox:/mnt# lvcreate -L 10M servidor -n Escuela
Rounding up size to full physical extent 12.00 MiB
Logical volume "Escuela" created.
```





LVM

Comprobando:

```
root@campesino-VirtualBox:/mnt# lvdisplay
--- Logical volume ---
LV Path                /dev/servidor/Tareas
LV Name                 Tareas
VG Name                 servidor
LV UUID                 zbYFCq-Bfxb-FPOI-A0uP-RNZf-D0Y8-WXayVl
LV Write Access         read/write
LV Creation host, time  campesino-VirtualBox, 2020-01-16 05:37:30 -0600
LV Status                available
# open                  0
LV Size                 <1.91 GiB
Current LE               488
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to      256
Block device            253:0

--- Logical volume ---
LV Path                /dev/servidor/Escuela
LV Name                 Escuela
VG Name                 servidor
LV UUID                 zCYrKU-hNNS-KyR3-uTKi-SuuL-KhA9-k3NWIw
LV Write Access         read/write
LV Creation host, time  campesino-VirtualBox, 2020-01-16 05:38:03 -0600
LV Status                available
# open                  0
LV Size                 12.00 MiB
```



LVM

Podemos ahora darle un sistema de archivos:

```
root@campesino-VirtualBox:/mnt# mkfs.ext4 /dev/servidor/Tareas
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 499712 4k blocks and 124928 inodes
Filesystem UUID: 8fd817f8-9cde-463f-bbe9-d5cdafcc4081
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Y montarlo:

```
root@campesino-VirtualBox:/mnt# mount /dev/servidor/Tareas montado/
root@campesino-VirtualBox:/mnt# cd montado/
root@campesino-VirtualBox:/mnt/montado# ls
lost+found
```

