

UEFI trabaja con GPT y actualmente es el mejor y trabaja con 128 particiones
MBR/DOS trabaja con BIOS LEGACY un sistema ya viejo y solo trabaja con 4 particiones
No se puede trabajar GPT con BIOS LEGACY
Todo depende de tu placa si acepta UEFI o BIOS LEGACY
De preferencia al hacer esta conversión hagan una copia de seguridad de sus datos.

Convertir discos de MBR/DOS a GPT

fdisk -l

/*Para ver nuestro **Disklabel**

/*Vemos que esta en **dos** y tiene **4** particiones

```
Arch Linux 5.3.13-arch1-1 (tty1)
archiso login: root (automatic login)

root@archiso ~ # loadkeys es
root@archiso ~ #
root@archiso ~ # loadkeys la-latn1
root@archiso ~ #
root@archiso ~ # fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: VBOX HARDISK
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: 0x67287096

Device      Boot      Start          End      Sectors  Size Id Type
/dev/sda1   *          2048       1026047       1024000    500M 83 Linux
/dev/sda2                1026048       53454847       52428800    25G 83 Linux
/dev/sda3          53454848       96446463       42991616    20.5G 83 Linux
/dev/sda4          96446464      104857599       8411136     4G 82 Linux swap / Solaris
```

/*Ejecutamos **gdisk [ruta-de-disco]**

/*Y podemos ver las opciones presionando **> ?**

gdisk /dev/sda

```
root@archiso ~ # gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.4

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
in memory. THIS OPERATION IS POTENTIALLY DESTRUCTIVE! Exit by
typing 'q' if you don't want to convert your MBR partitions
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.

Command (? for help): ?
b      back up GPT data to a file
c      change a partition's name
d      delete a partition
i      show detailed information on a partition
l      list known partition types
n      add a new partition
o      create a new empty GUID partition table (GPT)
p      print the partition table
q      quit without saving changes
r      recovery and transformation options (experts only)
s      sort partitions
t      change a partition's type code
u      verify disk
w      write table to disk and exit
x      extra functionality (experts only)
?      print this menu

Command (? for help):
```

```
Command (? for help): o
Proceed? (Y/N): Y
Command (? for help): w
Do you want to proceed? (Y/N): Y
```

```
n    add a new partition
o    create a new empty GUID partition table (GPT)
p    print the partition table
q    quit without saving changes
r    recovery and transformation options (experts only)
s    sort partitions
t    change a partition's type code
u    verify disk
w    write table to disk and exit
x    extra functionality (experts only)
?    print this menu

Command (? for help): o
This option deletes all partitions and creates a new protective MBR.
Proceed? (Y/N): Y
```

```
Command (? for help): o
This option deletes all partitions and creates a new protective MBR.
Proceed? (Y/N): Y
```

```
Command (? for help): ?
b    back up GPT data to a file
c    change a partition's name
d    delete a partition
i    show detailed information on a partition
l    list known partition types
n    add a new partition
o    create a new empty GUID partition table (GPT)
p    print the partition table
q    quit without saving changes
r    recovery and transformation options (experts only)
s    sort partitions
t    change a partition's type code
u    verify disk
w    write table to disk and exit
x    extra functionality (experts only)
?    print this menu
```

```
u    verify disk
w    write table to disk and exit
x    extra functionality (experts only)
?    print this menu
```

```
Command (? for help): w
```

```
Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!
```

```
Do you want to proceed? (Y/N): Y
OK: writing new GUID partition table (GPT) to /dev/sda.
The operation has completed successfully.
root@archiso ~ #
```

```
root@archiso ~ #
root@archiso ~ # fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: VBOX HARDDISK
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: 955ACCC3-BDA4-46D9-B7C5-959EEAEEB71A

Disk /dev/loop0: 524.49 MiB, 549953536 bytes, 1074128 sectors
```

/*Y como vemos al convertir nuestro disco a GPT elimino toda partición
/*Dejando el Disco Vacío, así que tengan cuidado al convertirlo

Convertir discos de GPT a MBR/DOS

```
fdisk -l
```

/*Para ver nuestro Disklabel

/*Vemos que esta en GPT y tiene 1 partición

/*Ejecutamos gdisk [ruta-de-disco]

/*Y podemos ver las opciones presionando > ?

```
gdisk /dev/sda
```

```
root@archiso ~ # fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: VBox HARDDISK
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: FE8A36D6-39B2-4E31-8158-BFE8D90F30EC

Device      Start      End  Sectors Size Type
/dev/sda1   2048 104857566 104855519   50G Linux filesystem

Disk /dev/loop0: 524.49 MiB, 549953536 bytes, 1074128 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
root@archiso ~ #
root@archiso ~ # gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.4

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

Found valid GPT with protective MBR; using GPT.

Command (? for help): ?
b        back up GPT data to a file
c        change a partition's name
d        delete a partition
i        show detailed information on a partition
l        list known partition types
n        add a new partition
o        create a new empty GUID partition table (GPT)
p        print the partition table
q        quit without saving changes
r        recovery and transformation options (experts only)
s        sort partitions
t        change a partition's type code
v        verify disk
w        write table to disk and exit
x        extra functionality (experts only)
?        print this menu

Command (? for help): r_
```

```

Command (? for help): r
Recovery/transformation command (? for help): g
MBR command (? for help): p
MBR command (? for help): w
Converted 1 partitions. Finalize and exit? (Y/N): Y

```

```

Command (? for help): r
Recovery/transformation command (? for help): ?
b      use backup GPT header (rebuilding main)
c      load backup partition table from disk (rebuilding main)
d      use main GPT header (rebuilding backup)
e      load main partition table from disk (rebuilding backup)
f      load MBR and build fresh GPT from it
g      convert GPT into MBR and exit
h      make hybrid MBR
i      show detailed information on a partition
l      load partition data from a backup file
m      return to main menu
o      print protective MBR data
p      print the partition table
q      quit without saving changes
t      transform BSD disklabel partition
v      verify disk
w      write table to disk and exit
x      extra functionality (experts only)
?      print this menu
Recovery/transformation command (? for help): g_

```

```

Recovery/transformation command (? for help): g
MBR command (? for help): ?
a      toggle the active/boot flag
c      recompute all CHS values
l      set partition as logical
o      omit partition
p      print the MBR partition table
q      quit without saving changes
r      set partition as primary
s      sort MBR partitions
t      change partition type code
w      write the MBR partition table to disk and exit
MBR command (? for help): p
** NOTE: Partition numbers do NOT indicate final primary/logical status,
** unlike in most MBR partitioning tools!
** Extended partitions are not displayed, but will be generated as required.
Disk size is 104857600 sectors (50.0 GiB)
MBR disk identifier: 0x00000000
MBR partitions:

```

Number	Boot	Start Sector	End Sector	Status	Can Be Logical	Can Be Primary	Code
1		2048	104857566	primary	Y	Y	0x83

```

MBR command (? for help): ?
a      toggle the active/boot flag
c      recompute all CHS values
l      set partition as logical
o      omit partition
p      print the MBR partition table
q      quit without saving changes
r      set partition as primary
s      sort MBR partitions
t      change partition type code
w      write the MBR partition table to disk and exit
MBR command (? for help): _

```



```

Disk size is 104857600 sectors (50.0 GiB)
MBR disk identifier: 0x00000000
MBR partitions:

Number  Boot  Start Sector    End Sector    Status    Can Be    Can Be    Code
      1                2048      104857566    primary      Y          Y         0x83

MBR command (? for help): ?
a      toggle the active/boot flag
c      recompute all CHS values
l      set partition as logical
o      omit partition
p      print the MBR partition table
q      quit without saving changes
r      set partition as primary
s      sort MBR partitions
t      change partition type code
w      write the MBR partition table to disk and exit

MBR command (? for help): w

Converted 1 partitions. Finalize and exit? (Y/N): Y
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
root@archiso ~ #

MBR command (? for help): w

Converted 1 partitions. Finalize and exit? (Y/N): Y
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
root@archiso ~ # fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: VBOX HARDDISK
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: 0x00000000

Device      Boot Start      End  Sectors Size Id Type
/dev/sda1                2048 104857566 104855519  50G 83 Linux

Disk /dev/loop0: 524.49 MiB, 549953536 bytes, 1074128 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
root@archiso ~ #

```

/*Aún no estoy seguro si todos los datos de sda1 estén bien o haya sufrido cambios
/*Pero ya tenemos nuestro disco en MBR/Dos

/*Consejo si tu placa puede trabajar con UEFI es mejor que uses tu disco en GPT

/*Dejo algunas diferencias.

BIOS LEGACY (Antiguo) - EFI o UEFI (Moderno)

Aquí depende del modelo de tu placa (Se puede actualizar de BIOS y pasar a UEFI)

Diferencias: (Existen más pero esto es el resumen :D)

- UEFI es el acrónimo de Unified Extensible Firmware Interface.
- UEFI tiene una mayor velocidad de arranque.
- UEFI Permite arrancar HDD y SSD de más de 2 TB utilizando GPT.
- UEFI puede ejecutarse en 32 o 64 bits, lo que posibilita un mayor rendimiento y una mayor velocidad de arranque o apagado.
- BIOS es el acrónimo de Basic Input Output System.
- BIOS viene dada por el sistema MBR (Master Boot Record).
- BIOS no puede leer otro sistema que no sea este MBR, GPT no es reconocible.

Para ver entrar al BIOS o UEFI son pulsando suprimir (del), o F2, o F1 o en algunos casos F10 son las más comunes.



MBR (Antiguo) - GPT (Moderno)

- MBR acrónimo de Master Boot Record.
- MBR el tamaño máximo de estas unidades es los 2 TB.
- MBR solo permite un máximo de cuatro particiones por cada disco duro.
- MBR Solo es posible 4 particiones, la 4 partición es extendida para tener más particiones.
- MBR funciona en sistemas operativos de 64 y 32 bits.
- GPT acrónimo de GUID Partition Table.
- GPT es capaz de ser empleado con unidades de 9,4 ZB.
(Cada Zettabyte equivale a mil billones de Gigabytes)
- GPT su límite es de 128 particiones por cada disco duro.
- GPT sólo funciona en sistemas operativos de 64 bits.

Se puede usar migrar su disco MBR a GPT pero debes tener el riesgo de perder toda su información en el disco.