

Arch Linux

Una distribución de Linux® liviana y flexible que intenta mantenerlo simple. Actualmente tenemos paquetes oficiales optimizados para la arquitectura x86-64 - 64bits

Pero también contamos con la distribución no-oficial optimizados para la arquitectura 32bits
<https://archlinux32.org/>

Contamos con Paquetes Oficiales con un gestor de paquetes **Pacman** es una de las principales características distintivas de Arch Linux y un repositorio de paquetes no-oficiales (**AUR**) - **Arch User Repository** operado por la comunidad que crece en tamaño y calidad todos los días.



La Descarga:

Desde la página oficial pueden descargar la ISO:
<https://www.archlinux.org/download/>

Es recomendable descargarlo por **BitTorrent**

Arch Linux Downloads

Release Info

The image can be burned to a CD, mounted as an ISO file, or be directly written to a USB

- **Current Release:** 2019.09.01
- **Included Kernel:** 5.2.11
- **ISO Size:** 623.0 MB
- **Installation Guide**
- **Resources:**
 - [Bug Tracker](#)
 - [Mailing List](#)

Existing Arch Users

If you are an existing Arch user, there is no need to download a new ISO to update your

BitTorrent Download (recommended)

If you can spare the bytes, please leave the client open after your download is finished,

- [Magnet link for 2019.09.01](#)
- [Torrent for 2019.09.01](#)

También se puede descargar por Navegador escoge el país donde vive o el país más cercano

Ecuador <ul style="list-style-type: none">▪ cedia.org.ec▪ esepoch.edu.ec▪ uta.edu.ec	Colombia <ul style="list-style-type: none">▪ edatel.net.co▪ udenar.edu.co▪ upb.edu.co	Chile <ul style="list-style-type: none">▪ archlinux.cl▪ ufro.cl▪ ufro.cl
Mexico <ul style="list-style-type: none">▪ mirror.pkgbuild.com		

Y descargar la imagen .ISO

México

Index of /iso/2019.09.01/

../		
arch/	01-Sep-2019 05:13	-
archlinux-2019.09.01-x86_64.iso	01-Sep-2019 05:13	653262848
archlinux-2019.09.01-x86_64.iso.sig	01-Sep-2019 05:15	310
archlinux-2019.09.01-x86_64.iso.torrent	01-Sep-2019 05:15	40793
archlinux-bootstrap-2019.09.01-x86_64.tar.gz	01-Sep-2019 05:14	154047390
archlinux-bootstrap-2019.09.01-x86_64.tar.gz.sig	01-Sep-2019 05:15	310
md5sums.txt	01-Sep-2019 05:15	145
shasums.txt	01-Sep-2019 05:15	161

Ecuador

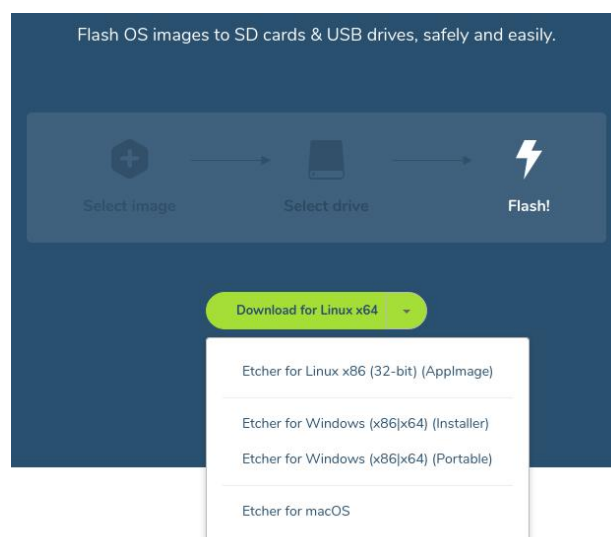
../archlinux/iso/2019.09.01/

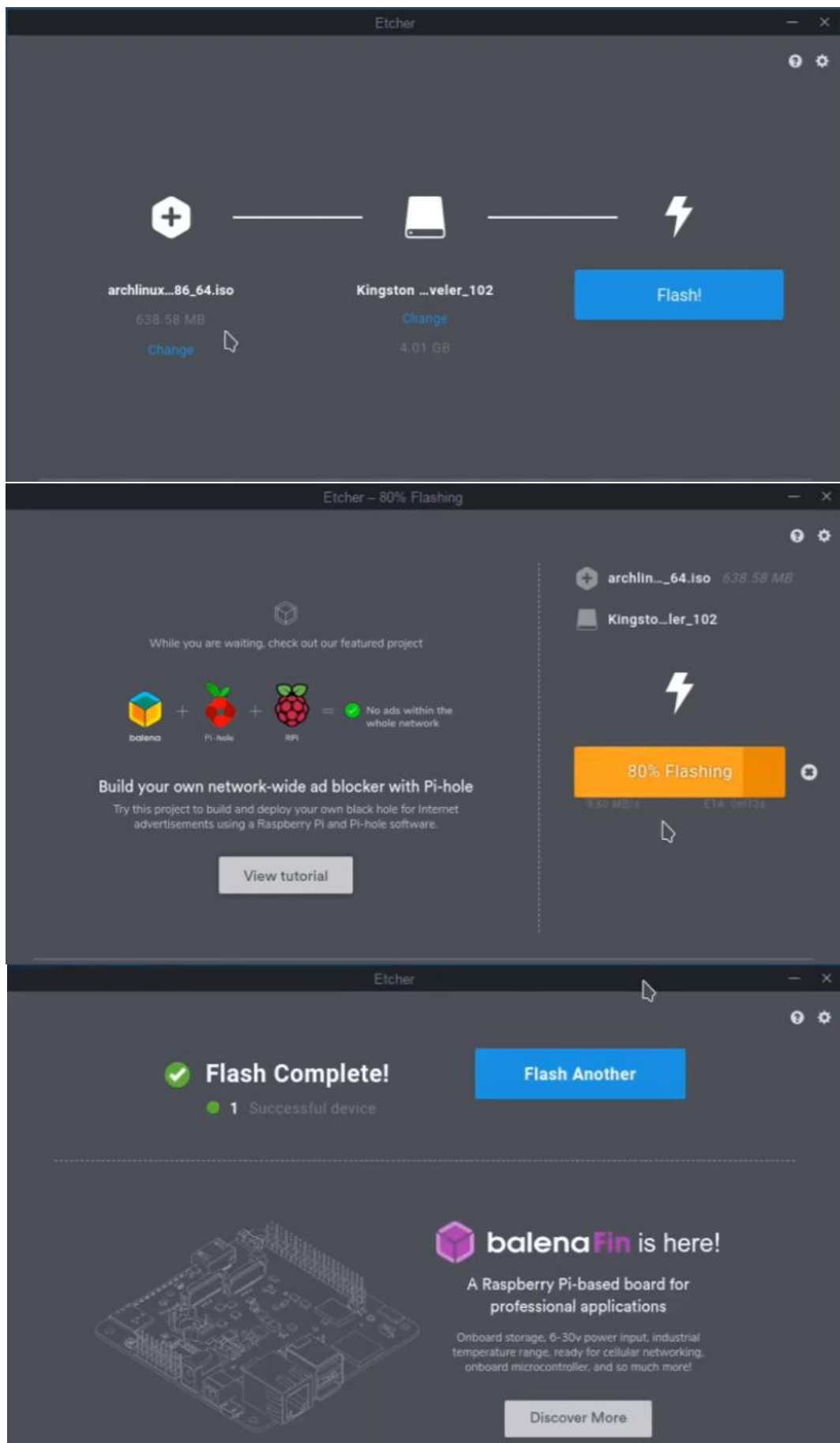
filename	timestamp	filesize
..	01 Sep 2019, 00:37:25	
arch	01 Sep 2019, 00:13:06	
archlinux-2019.09.01-x86_64.iso	01 Sep 2019, 00:13:47	623.0 MB
archlinux-2019.09.01-x86_64.iso.sig	01 Sep 2019, 00:15:01	310 byte
archlinux-2019.09.01-x86_64.iso.torrent	01 Sep 2019, 00:15:04	39.8 kB
archlinux-bootstrap-2019.09.01-x86_64.tar.gz	01 Sep 2019, 00:14:58	146.9 MB
archlinux-bootstrap-2019.09.01-x86_64.tar.gz.sig	01 Sep 2019, 00:15:02	310 byte
md5sums.txt	01 Sep 2019, 00:15:03	145 byte
shasums.txt	01 Sep 2019, 00:15:02	161 byte
9 archivos		769.9 MB

Para crear una USB-booteable en lo personal y mi recomendación es:

- Tener USB de 4GB o más
- Descargar Etcher - <https://www.balena.io/etcher/>

Etcher es multiplataforma, el procedimiento muy sencillo y el resultado es el mismo en cualquier plataforma, seleccionas la ISO, seleccionas la USB y le das en FLASH.





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

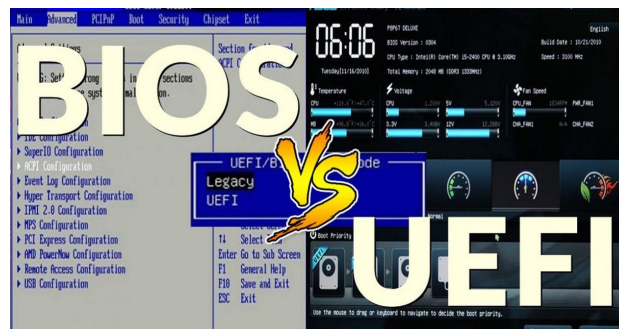
Aquí depende del modelo de tu placa (Se puede actualizar de BIOS y pasar a UEFI)

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.

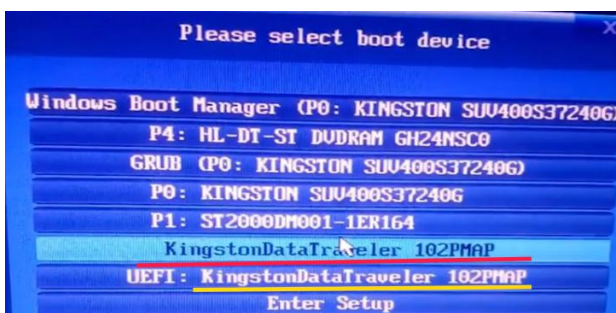
Para ver entrar al BIOS o UEFI son pulsando suprimir (Del), o F2, o F1 o en algunos casos F10 son las más comunes.



Para iniciar el Boot Menú y seleccionar el CD o USB
La tecla depende de la placa madre, puede ser ESC, F2, F7, F9, F12



Hay placas que aceptan UEFI y Bios, otras que solo aceptan BIOS

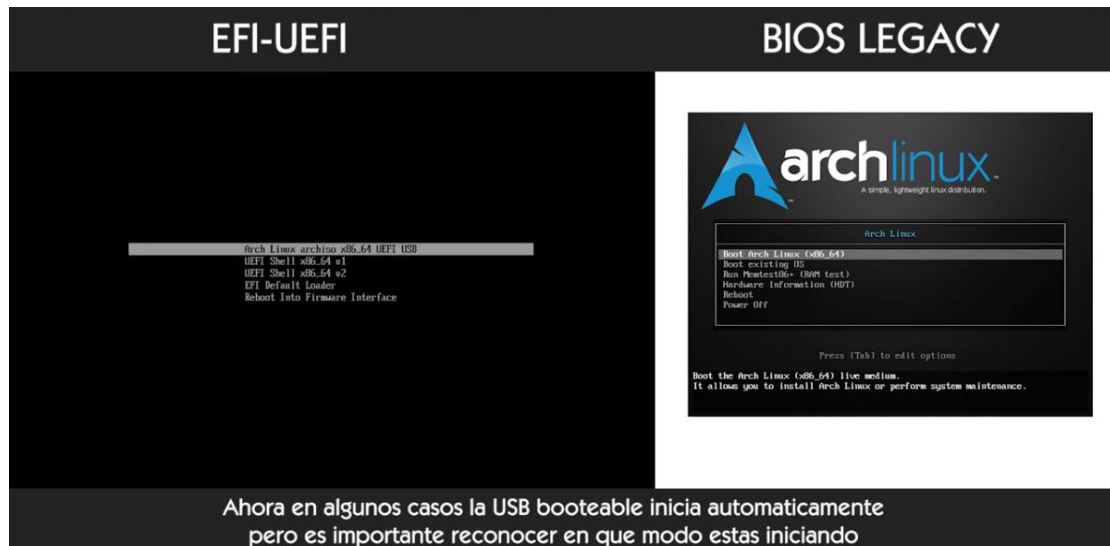


Arch Linux (Bios Legacy / UEFI)

Si vamos a instalar Arch Linux en nuestra PC.

La USB-Bootleable con la ISO **detectara automáticamente** si estamos usando UEFI o BIOS LEGACY.

*/*Es importante diferenciar esto para la instalación del GRUB*/*



MBR/Dos (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 migrar su disco MBR a GPT pero debes tener en cuenta

Si tu Placa Madre acepta UEFI porque BIOS NO trabaja con GPT.

Y podemos revisar en nuestro **Distro Linux (Kde Neon)** cual estamos usando MBR/Dos o GPT con **sudo fdisk -l**

```
pc: bash — Konsole <2>
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
pc@pc-VirtualBox:~$ sudo fdisk -l
Disco /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x497bd2b8

Dispositivo Inicio Comienzo      Final  Sectores Tamaño Id Tipo
/dev/sda1 *      2048  104855551 104853504 50G 83 Linux
```

Vemos que el Tipo de etiqueta de disco estamos usando: **MBR/Dos**

Entonces vamos a iniciar nuestra **USB en Bios Legacy** porque nuestro disco ya tiene formado **MBR/Dos**

Iniciamos la USB con el Logo de Arch que es Bios.

ARCHLINUX - UEFI

/*Ponemos nuestro teclado en español

`loadkeys es`

/*Ponemos nuestro teclado en latino

`loadkeys la-latin1`

/*Si tenemos conexión cableada inmediato tenemos internet y lo confirmamos con.

`ping -c 3 archlinux.org`

```
root@archiso ~ #  
root@archiso ~ # ping -c 3 archlinux.org  
PING archlinux.org (138.201.81.199) 56(84) bytes of data.  
64 bytes from apollo.archlinux.org (138.201.81.199): icmp_seq=1 ttl=63 time=241 ms  
64 bytes from apollo.archlinux.org (138.201.81.199): icmp_seq=2 ttl=63 time=242 ms  
64 bytes from apollo.archlinux.org (138.201.81.199): icmp_seq=3 ttl=63 time=243 ms  
  
--- archlinux.org ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2067ms  
rtt min/avg/max/mdev = 240.803/241.776/242.568/0.732 ms  
root@archiso ~ # _
```

/*El parámetro -c 3 establece que se ejecute ping tres veces.

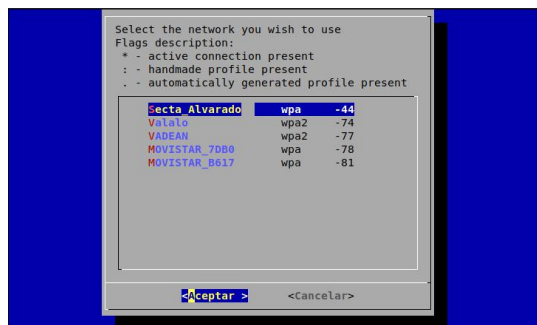
/*Por defecto en la ISO habilita el servicio de dhcpd para el uso de red cableada

/*Si no hay conexión con Ethernet debemos activar el servicio de red

`systemctl enable dhcpd`
`systemctl start dhcpd`

/*Si tenemos conexión Wi-Fi.

`wifi-menu`



fdisk -l

/*Listado de discos existentes

/*Aquí podemos saber cual es la ruta de nuestro disco `/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: dos
Disk identifier: 0x497bd2b8

Device      Boot  Start        End  Sectors  Size Id Type
/dev/sda1   *      2048    104855551  104853504   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 ~ #
```

/*Recordemos que `MBR/Dos` es para BIOS Legacy y `GPT` es para UEFI.

/*Si iniciaste la USB con la pantalla negra y tu disco marca `DOS` debes convertir tu disco a `GPT`

/*O mejor cambia el modo e ingresa tu usb con el logo de Arch que es para BIOS Legacy

/*En este caso `/dev/loop0` es la imagen ISO de ArchLinux

/*En GitHub tengo un PDF de como convertir tu disco a `GPT` para trabajar con UEFI

cfdisk /dev/sda

/*`cfdisk` es para crear particiones nuevas dependiendo nuestra necesidades

/*Teniendo ya nuestro disco en `GPT` podemos particionar nuestro disco

```

Disk: /dev/sda
Size: 50 GiB, 53687091200 bytes, 104857600 sectors
Label: gpt, identifier: 8C3E0CC5-46C8-4736-A78D-D9AC2CD155AF

Device
>> /dev/sda1

[ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ] [ Write ] [ Dump ]

Quit program without writing changes
```

/*Vamos empezar eliminando las particiones en `[Delete]` y creando nuevas en `[New]`

/*Aquí para UEFI es necesario tener una partición en `FAT32` para boot

/*Podemos crear varias particiones y solo se generará: `sda1`, `sda2`, `sda3`...

/*Podemos crear las particiones y cambiar el tipo de partición en `[Type]`

/*Al finalizar el particionado le damos en `[Write]` para escribir los cambios

/*`Flecha Arriba / Abajo - Flechas Derecha / Izquierda` - para movernos en `cfdisk`

Esquemas de particionado

/boot >> Gestor de arranque tanto para BIOS o UEFI (500MB Aprox)
/ >> Root - Archivos de administrador (Programas Instalados)
/home >> Archivos del usuario (Datos)
/swap >> memoria swap (Ram virtual)

El sistema Android esta basado en linux y / tambien es root es la raíz del sistema.



MEMORIA SWAP

Menos de 1GB RAM física >> 2GB de SWAP
Entre 2GB a 4GB RAM física >> 2GB a 4GB de SWAP
8GB de RAM física >> 4GB de SWAP
Más de 8GB de RAM física >> 2GB a 4GB de SWA

/*Algunos ejemplos de particiones

Sda1 > boot > [Type] > EFI System

Sda2 > root - home

Disk: /dev/sda					
Size: 50 GiB, 53687091200 bytes, 104857600 sectors					
Label: gpt, identifier: 8C3E0CC5-46C8-4736-A78D-D9AC2CD155AF					
Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	1026047	1024000	500M	EFI System
>> /dev/sda2	1026048	104857566	103831519	49.5G	Linux filesystem

Sda1 > boot > [Type] > EFI System

Sda2 > root - home

Sda3 > Swap > [Type] > Linux swap

Disk: /dev/sda					
Size: 50 GiB, 53687091200 bytes, 104857600 sectors					
Label: gpt, identifier: 8C3E0CC5-46C8-4736-A78D-D9AC2CD155AF					
Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	1026047	1024000	500M	EFI System
/dev/sda2	1026048	96446463	95420416	45.5G	Linux filesystem
>> /dev/sda3	96446464	104857566	8411103	4G	Linux swap

/*Ahora si quieren tener sus Archivos y datos si eres nuevo puedes hacer muchos errores que por andar aprendiendo puedes romper el sistema y toca formatear de nuevo el sistema, para esas situaciones es mejor asignarle un partición para el Home

/*Incluso el Home puede ser compartida con otras distros si le instalas más adelante

Sda1 > boot > [Type] > EFI System

Sda2 > root

Sda3 > home

Sda4 > Swap > [Type] > Linux swap

/*Si nos toca formatear solo formateamos sda1 = boot, sda2 = root y sda4 = swap
 /***Home**, están todos nuestros archivos o documentos **no lo formateamos** y solo montamos

Disk: /dev/sda
 Size: 50 GiB, 53687091200 bytes, 104857600 sectors
 Label: gpt, identifier: 8C3E0CC5-46C8-4736-A78D-D9AC2CD155AF

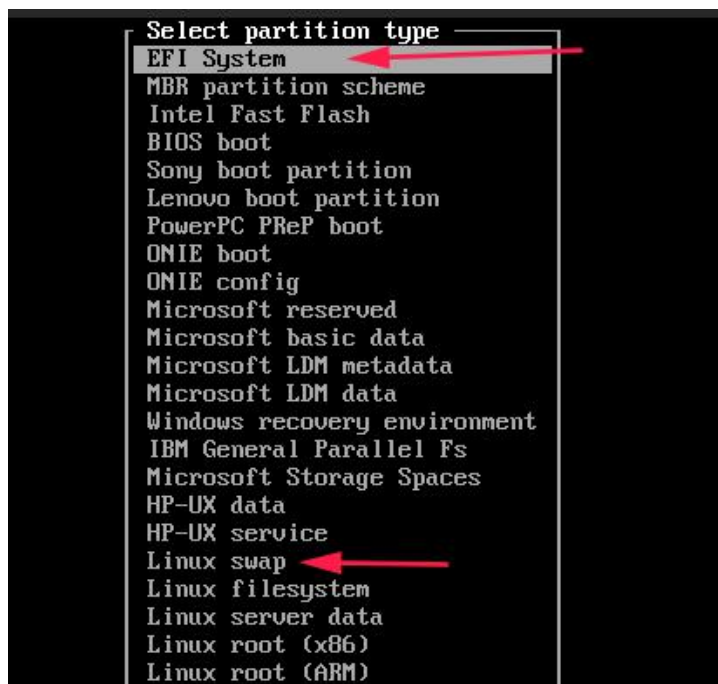
Device	Start	End	Sectors	Size	Type
>> /dev/sda1	2048	1026047	1024000	500M	EFI System
/dev/sda2	1026048	53454847	52428800	25G	Linux filesystem
/dev/sda3	53454848	96446463	42991616	20.5G	Linux filesystem
/dev/sda4	96446464	104857566	8411103	4G	Linux swap

/*Siendo UEFI y GPT permite muchas particiones así que no nos preocupamos.

/*Para cambiarle el [Type] y seleccionamos **Linux swap**

/*Para cambiarle el [Type] y seleccionamos **EFI system**

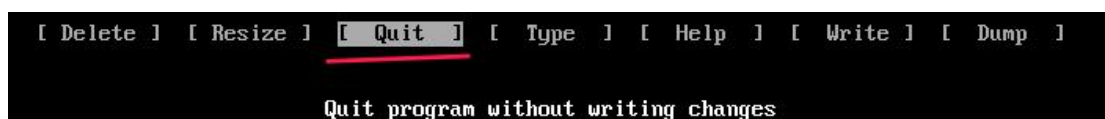
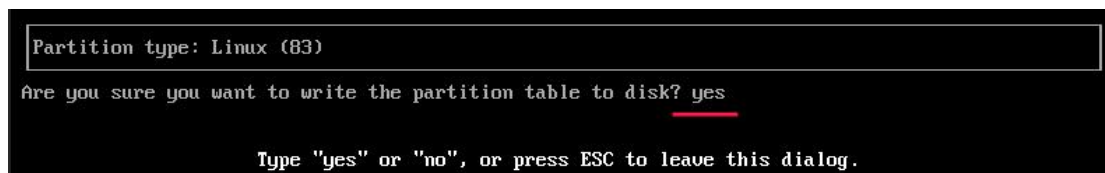
/*A la partición correspondiente.



/*Una vez que tenemos todo particionado y escogido el esquema correcto

/*Le damos en [Write] y **yes**

/*Luego de confirmar los cambios [Quit]



`fdisk -l`

/*Volvemos hacer un listado de particiones para revisar que los cambios fueron realizados
/*Es importante saber que función tiene sda1 para que servirá y así con todas las particiones...

```
root@archiso ~ #  
root@archiso ~ # fdisk -l  
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors  
Disk model: UBOX 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: 8C3E0CC5-46C8-4736-A78D-D9AC2CD155AF  
  
Device            Start       End   Sectors  Size Type  
/dev/sda1         2048      1026047  1024000  500M EFI System  
/dev/sda2      1026048   53454847  52428800  25G Linux filesystem  
/dev/sda3      53454848   96446463  42991616 20.5G Linux filesystem  
/dev/sda4      96446464  104857566  8411103   4G Linux swap  
  
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 ~ # _
```

/*Formateamos las particiones y luego montaremos
/*En el sistema UEFI la partición para **boot debe ser formateada en FAT32**

`mkfs.fat -F 32 /dev/sda1`

/*Luego formateamos las particiones Linux en Ext4

`mkfs.ext4 /dev/sda2`

`mkfs.ext4 /dev/sda3`

/*Ext4 es un sistema de archivos transaccional (en inglés journaling) de Linux

/*Formateamos la memoria SWAP (Ram virtual)

`mkswap /dev/sda4`

/*Activamos la memoria SWAP (Ram virtual)

`swapon /dev/sda4`

/*Montamos nuestras particiones formateadas para eso hay que identificar cual es la partición para root, boot, home...

/*Ahora para **root** en este ejemplo es **sda2** lo cual **debe ser montado primero**

`mount /dev/sda2 /mnt/`

/mnt - sistemas de archivos montados manualmente en el disco duro.
/ - diagonal invertida significa root

/*Ahora toca crear las carpetas para **boot y Home**, luego montar las particiones

```
mkdir /mnt/boot/  
mkdir /mnt/home/
```

```
mount /dev/sda1 /mnt/boot/  
mount /dev/sda3 /mnt/home/
```

/*Ahora montadas las particiones instalamos los programas base y lo más esencial posible
/*Pero hay muchos casos que pacstrap les resulta una descarga muy lenta y eso se debe al no tener los mirros mas rápidos de descarga.
/*Para tener los Mirros más rápidos para tener mejores descargas usaremos **reflector**.

```
path: root/pacstrap.in  
blob: 703df11efba9244289f96c87386c9b26b75d4700 (plain)  
1 #!/bin/bash  
2  
3 #  
4 # Assumptions:  
5 # 1) User has partitioned, formatted, and mounted partitions on /mnt  
6 # 2) Network is functional  
7 # 3) Arguments passed to the script are valid pacman targets  
8 # 4) A valid mirror appears in /etc/pacman.d/mirrorlist  
9 #  
10  
11 shopt -s extglob  
12  
13 m4_include(common)  
14
```

/*Si revisamos **nano /etc/pacman.d/mirrorlist**

/*Vemos que tenemos toda una lista de servers y la función es:

Si falla el primer Server que en este caso es Ecuador, entra el siguiente Server y si falla entra el siguiente y así viene la lista

```
GNU nano 4.5 /etc/pacman.d/mirrorlist  
##  
## Arch Linux repository mirrorlist  
## Filtered by mirror score from mirror status page  
## Generated on 2019-12-01  
##  
## Ecuador  
Server = http://mirror.cedia.org.ec/archlinux/$repo/os/$arch  
## United States  
Server = http://mirrors.rutgers.edu/archlinux/$repo/os/$arch  
## United States  
Server = http://mirrors.xtom.com/archlinux/$repo/os/$arch  
## Taiwan  
Server = http://archlinux.cs.nctu.edu.tw/$repo/os/$arch  
## Germany  
Server = http://artfiles.org/archlinux.org/$repo/os/$arch
```

/*Ahora instalar reflector es:

```
pacman -Sy reflector python3 --noconfirm
```

/*Para ejecutar reflector y tener los mejores Mirrors Servers es: **(es una L minúscula)**

```
reflector --verbose -l 5 --sort rate --save /etc/pacman.d/mirrorlist
```

```

root@archiso ~ #
root@archiso ~ # reflector --verbose -l 5 --sort rate --save /etc/pacman.d/mirrorlist
[2019-12-13 11:18:17] INFO: rating https://ger.mirror.pkgbuild.com/
[2019-12-13 11:18:17] INFO: rating rsync://ger.mirror.pkgbuild.com/packages/
[2019-12-13 11:18:17] INFO: rating http://mirror.orbit-os.com/archlinux/
[2019-12-13 11:18:17] INFO: rating https://mirror.orbit-os.com/archlinux/
[2019-12-13 11:18:17] INFO: rating rsync://mirror.orbit-os.com/archlinux/
[2019-12-13 11:18:21] INFO: Server                                     Rate      Time
[2019-12-13 11:18:21] INFO: http://mirror.orbit-os.com/archlinux/      187.13 KiB/s  0.72 s
[2019-12-13 11:18:21] INFO: https://ger.mirror.pkgbuild.com/    283.29 KiB/s  0.48 s
[2019-12-13 11:18:21] INFO: https://mirror.orbit-os.com/archlinux/ 284.18 KiB/s  0.48 s
[2019-12-13 11:18:21] INFO: rsync://mirror.orbit-os.com/archlinux/  34.68 KiB/s  3.90 s
[2019-12-13 11:18:21] INFO: rsync://ger.mirror.pkgbuild.com/packages/ 31.03 KiB/s  4.35 s
root@archiso ~ #
root@archiso ~ # cat /etc/pacman.d/mirrorlist
#####
##### Arch Linux mirrorlist generated by Reflector #####
#####
# With:      reflector --verbose -l 5 --sort rate --save /etc/pacman.d/mirrorlist
# When:      2019-12-13 11:18:21 UTC
# From:      https://www.archlinux.org/mirrors/status/json/
# Retrieved: 2019-12-13 11:18:17 UTC
# Last Check: 2019-12-13 10:56:04 UTC

Server = https://mirror.orbit-os.com/archlinux/$repo/os/$arch
Server = https://ger.mirror.pkgbuild.com/$repo/os/$arch
Server = http://mirror.orbit-os.com/archlinux/$repo/os/$arch
Server = rsync://mirror.orbit-os.com/archlinux/$repo/os/$arch
Server = rsync://ger.mirror.pkgbuild.com/packages/$repo/os/$arch
root@archiso ~ #

```

/*Instalación base: programas, configuraciones, directorios, etc...

/*Editor de texto en terminal: Nano

```
pacstrap /mnt base base-devel nano
```

/*Instalación para configuración de Ethernet

```
pacstrap /mnt dhcpcd netctl iwd net-tools
```

/*Agregamos estos paquetes si tienen Wi-Fi

```
pacstrap /mnt wireless_tools dialog wpa_supplicant
```

El archivo **/etc/fstab**

/*Es usado para definir cómo las particiones,

/*Estas definiciones se montaran de forma dinámica en el arranque

```
genfstab -p /mnt >> /mnt/etc/fstab
```

/*Luego de generar el archivo fstab para las etiquetas de nuestras particiones

/*Revisamos con:

```
cat /mnt/etc/fstab
```

```

root@archiso ~ #
root@archiso ~ # genfstab -p /mnt >> /mnt/etc/fstab
root@archiso ~ #
root@archiso ~ # cat /mnt/etc/fstab
# Static information about the filesystems.
# See fstab(5) for details.

# <file system> <dir> <type> <options> <dump> <pass>
# UUID=b4a9d86a-b8dc-4618-8f50-ccd8e0c9c6a
/dev/sda2 / ext4 rw,relatime 0 1

# UUID=ac2eff3f-d304-4dd9-ba97-0cf2f599b4e2
/dev/sda1 /boot ext4 rw,relatime 0 2

# UUID=ac2eff3f-d304-4dd9-ba97-0cf2f599b4e2
/dev/sda1 /home ext4 rw,relatime 0 2

# UUID=77fdc9f6-44b4-4d9d-9ee0-e6cf9c838dd9
/dev/sda4 none swap defaults 0 0
root@archiso ~ #

```

```
arch-chroot /mnt
```

/*Entramos a raíz como root

/*Dentro de nuestro sistema vamos a configurar idioma, teclado, hora y usuarios.

```
nano /etc/locale.gen
```

/*Quitamos el # que es comentario en nuestro idioma >> **es_** y nuestro país

/*En mi caso es PERÚ es > PE

/*Debe terminar en **es_[país].UTF-8 UTF-8**

/*Ctrl + W para buscar palabras en nano

/*Ctrl + O para guardar en nano

/*Ctrl + X para cerrar en nano

```
locale-gen
```

/*Genera el idioma seleccionado

```
[root@archiso /]#  
[root@archiso /]# locale-gen  
Generating locales...  
  es_PE.UTF-8... done  
Generation complete.  
[root@archiso /]#
```

```
es_ES.UTF-8 UTF-8  
#es_ES ISO-8859-1  
#es_ES@euro ISO-8859-15  
#es_GT.UTF-8 UTF-8  
#es_GT ISO-8859-1  
#es_HN.UTF-8 UTF-8  
#es_HN ISO-8859-1  
#es_MX.UTF-8 UTF-8  
#es_MX ISO-8859-1  
#es_NI.UTF-8 UTF-8  
#es_NI ISO-8859-1  
#es_PA.UTF-8 UTF-8  
#es_PA ISO-8859-1  
  
es_PE.UTF-8 UTF-8  
  
#es_PE ISO-8859-1  
#es_PR.UTF-8 UTF-8  
  
^G Get Help      ^O Write Out  
^X Exit          ^R Read File
```

/*Vamos a estar usando mucho el comando echo

/*Su uso es el siguiente:

```
echo [texto-a-ingresar] > [archivo-a-guardar]
```

```
echo(Espacio)[texto-a-ingresar](Espacio)>(Espacio)[archivo-a-guardar]
```

```
echo(Espacio)LANG=es_PE.UTF-8(Espacio)>(Espacio)/etc/locale.conf
```

/*Establezca la variable LANG en locale.conf

```
export(Espacio)LANG=es_PE.UTF-8
```

/*Exporte la variable LANG con el local especificado

```
[root@archiso /]#  
[root@archiso /]# locale-gen  
Generating locales...  
  es_PE.UTF-8... done  
Generation complete.  
[root@archiso /]#  
[root@archiso /]# echo LANG=es_PE.UTF-8 > /etc/locale.conf  
[root@archiso /]#  
[root@archiso /]# export LANG=es_PE.UTF-8  
[root@archiso /]#  
[root@archiso /]# _
```

/*Establezca la zona horaria
 /*Primero vamos a revisar cual es nuestra Ciudad
 /*Para América Latina es en América
 /*Para España es Europe
 /*Respete las Mayúsculas y Minúsculas

```
ls /usr/share/zoneinfo/
```

```
[root@archiso /]#
[root@archiso /]# ls /usr/share/zoneinfo/
Africa      Canada  EST      GMT+0    Iran      MET      Poland    ROK      WET
America     CET     EST5EDT  GMT-0    iso3166.tab Mexico    Portugal  Singapore W-SU
Antarctica  Chile   Etc       GMT0     Israel    MST      posix     Turkey   zone1970.tab
Arctic      CST6CDT Europe  Greenwich Jamaica  MST7MDT  posixrules tzdata.zi zone.tab
Asia        Cuba    Factory   Hongkong Japan     NavaJo   PAC       UCT      Zulu
Atlantic    EET     GB        HST      KwaJalein NZ      PST8PDT   Universal
Australia   Egypt   GB-Eire   Iceland  leapseconds NZ-CHAT  right     US
Brazil      Eire    GMT       Indian   Libya     Pacific  ROC       UTC
```

```
ls /usr/share/zoneinfo/America/
```

```
ls /usr/share/zoneinfo/Europe/
```

/*La mejor ayuda es Google, ejemplo para Argentina:

Google zona horaria argentina

America/Argentina/Buenos_Aires: Zona horaria en Argentina ...
<https://www.zeitverschiebung.net/> Sudamérica Argentina ▾
 Cambio de horario: En ésta zona horaria no hay horario de verano. Calculadora de zona horaria. Conversor de diferencias horarias. America/Argentina/ ...
 Mexico City: -2 horas Los Angeles: -4 horas

/*Nos dice que esta en **America/Argentina/Buenos_Aires**.

```
[root@archiso /]#
[root@archiso /]# ls /usr/share/zoneinfo/America/Argentina/
Buenos_Aires ComodRivadavia JuJuy Mendoza Salta San_Luis Ushuaia
Catamarca Cordoba La_Rioja Rio_Gallegos San_Juan Tucuman
[root@archiso /]#
[root@archiso /]# ln -sf /usr/share/zoneinfo/America/Argentina/Buenos_Aires /etc/localtime
```

```
ln(Espacio)-sf(Espacio)/usr/share/zoneinfo/America/Lima(Espacio)/etc/localtime
```

```
[root@archiso /]#
[root@archiso /]# ls /usr/share/zoneinfo/America/
Adak      Catamarca      Glace_Bay      Los_Angeles      Nome      Santo_Domingo
Anchorage Cayenne        Godthab        Louisville      Noronha      Sao_Paulo
Anguilla  Cayman         Goose_Bay      Lower_Princes   North_Dakota Scoresbyund
Antigua   Chicago        Grand_Turk     Maceio          Ojinaga      Shiprock
Araguaina Chihuahua      Grenada        Managua          Panama        Sitka
Argentina Coral_Harbour  Guadeloupe     Manaus          Pangnirtung   St_Barthelemy
Aruba     Cordoba        Guatemala      Marigot          Paramaribo    St_Johns
Asuncion  Costa_Rica     Guayaquil      Martinique       Phoenix       St_Kitts
Atikokan  Creston       Guyana         Matamoros        Port-au-Prince St_Lucia
Atka      Cuabaa        Halifax        Mazatlan         Porto_Acre     St_Thomas
Bahia     Curacao       Havana         Mendoza          Port_of_Spain St_Vincent
Bahia_Banderas Danmarkshavn  Hermosillo     Menominee        Porto_Velho    Swift_Current
Barbados  Dawson        Indiana        Merida           Puerto_Rico     Tegucigalpa
Belen     Dawson_Creek  Indianapolis    Metlakatla      Punta_Arenas   Thule
Belize    Denver        Inuvik         Mexico_City      Rainy_River     Thunder_Bay
Blanc-Sablon Detroit       Iqaluit        Miquelon         Rankin_Inlet    Tijuana
Boa_Vista Dominica      Jamaica        Moncton          Recife          Toronto
Bogota    Edmonton     JuJuy          Monterrey        Regina          Tortola
Boise     Eirunepe     Juneau         Montevideo       Resolute        Vancouver
Buenos_Aires El_Salvador   Kentucky       Montreal         Rio_Branco      Virgin
Cambridge_Bay Ensenada     Knox_IN        Montserrat       Rosario          Whitehorse
Campo_Grande Fortaleza    Kralendijk     Nassau           Santa_Isabel     Winnipeg
Cancun    Fort_Nelson  La_Paz         New_York         Santarem         Yakutat
Caracas   Fort_Wayne   Lima           Nipigon          Santiago         Yellowknife
[root@archiso /]#
[root@archiso /]# ln -sf /usr/share/zoneinfo/America/Lima /etc/localtime
[root@archiso /]#
```

/*Mi caso es de Perú y la Ciudad es Lima, respete las mayúsculas y minúsculas
 /* ln - Genera un enlace simbólico es un acceso al fichero

```
hwclock -w
```

/*El sistema está configurado para leer el reloj interno del equipo, después el reloj del sistema,
/*Estable el RTC a partir de la hora del sistema.

```
echo(Espacio)KEYMAP=es(Espacio)>(Espacio)/etc/vconsole.conf
```

/*Defina la distribución de teclado en vconsole.conf
/*para que permanezca en cada reinicio.

```
echo(Espacio)nombre_de_pc(Espacio)>(Espacio)/etc/hostname
```

/*Nombre del equipo, esto no es USUARIOS!
/*En **nombre_de_pc** le cambian por uno a su gusto, **más adelante crearan un usuario**

```
[root@archiso /]#  
[root@archiso /]# echo KEYMAP=es > /etc/vconsole.conf  
[root@archiso /]#  
[root@archiso /]# echo nombre_de_pc > /etc/hostname  
[root@archiso /]#  
[root@archiso /]# _
```

/*Modificamos el archivo Hosts
/*Si el sistema tiene una dirección IP permanente, reemplazar por 127.0.1.1.
/*Es importante saber que nombre pusieron en **Hostname** porque aquí sera usado

```
nano /etc/hosts
```

```
127.0.0.1[tab] localhost  
::1 [tab] localhost  
127.0.1.1[tab] nombre_de_pc.localdomain(Espacio)nombre_de_pc
```

```
GNU nano 4.6 /etc/hosts  
# Static table lookup for hostnames.  
# See hosts(5) for details.  
  
127.0.0.1    localhost  
::1         localhost  
127.0.1.1    nombre_de_pc.localdomain nombre_de_pc
```

/*Ctrl + O para guardar en nano
/*Ctrl + X para cerrar en nano

/*Otra opción es:

```
echo(Espacio)127.0.1.1(Espacio)nombre_de_pc.localdomain(Espacio)nombre_de_pc(Espacio)>(Espacio)/etc/hosts
```

```
[root@archiso /]#  
[root@archiso /]# echo 127.0.1.1 nombre_de_pc.localdomain nombre_de_pc > /etc/hosts  
[root@archiso /]#  
[root@archiso /]# cat /etc/hosts  
127.0.1.1 nombre_de_pc.localdomain nombre_de_pc  
[root@archiso /]#
```

```
passwd root
```

/*Contraseña para root

```
useradd -m -g users -s /bin/bash nombre_de_usuario
```

```
passwd nombre_de_usuario
```

/*Creamos nuestro usuario, para entrar a nuestro sistema.

/*En **nombre_de_usuario**, le cambiamos a nuestro gusto.

```
[root@archiso /]#
[root@archiso /]# passwd root
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
[root@archiso /]#
[root@archiso /]# useradd -m -g users -s /bin/bash nombre_de_usuario
[root@archiso /]#
[root@archiso /]# passwd nombre_de_usuario
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
[root@archiso /]#
[root@archiso /]#
```

/*Ahora con nuestro usuario si queremos hacer algo como root por lo general usamos **SUDO**

/*Sudo tendrá efecto si nuestro usuario esta en la lista de **Sudoers**.

```
nano /etc/sudoers
```

```
nombre_de_usuario ALL=(ALL) ALL
```

/*Buscamos root ALL=(ALL) ALL y abajo ponemos nuestro usuario

/*para que tenga permisos y mismos privilegios que root al ejecutar sudo.

```
GNU nano 4.6 /etc/sudoers
## sudoreplay and reboot. Use sudoreplay to play back logged sessions.
# Defaults log_output
# Defaults!usr/bin/sudoreplay !log_output
# Defaults!usr/local/bin/sudoreplay !log_output
# Defaults!REBOOT !log_output

##
## Runas alias specification
##

##
## User privilege specification
##
root ALL=(ALL) ALL
nombre_de_usuario ALL=(ALL) ALL_

## Uncomment to allow members of group wheel to execute any command
# %wheel ALL=(ALL) ALL

## Same thing without a password
# %wheel ALL=(ALL) NOPASSWD: ALL

## Uncomment to allow members of group sudo to execute any command
# %sudo ALL=(ALL) ALL
```

/*Ctrl + O para guardar

/*Ctrl + X para cerrar

```
systemctl enable dhcpcd
```

```
pacman -Syu
```

/*Dynamic Host Configuration Protocol (DHCP)
/*El servidor proporciona a los clientes una dirección IP dinámica,
/*La máscara de subred, Gracias a "systemd" podemos activar ese servicio

```
pacman -S reflector
```

/*Reflector es un script que es capaz de generar una lista y usa los repositorios mas rápidos
/*Ordenarlos en base a su velocidad, y sobrescribir el archivo /etc/pacman.d/mirrorlist

```
reflector --verbose -l 5 --sort rate --save /etc/pacman.d/mirrorlist
```

/*Filtrará los primeros 5 mirrors, los ordenará en base a su velocidad de descarga
/* -l es una ELE L minúscula

```
pacman -S networkmanager ifplugd
```

```
systemctl enable NetworkManager
```

/*Detección y configuración automática para conectarse a la red, respetar la mayúsculas.
/*Las funcionalidades de NetworkManager son útiles tanto para redes inalámbricas y cable.
//NetworkManager se ha iniciado, se conectará automáticamente «conexiones del sistema» disponibles que estén configuradas.

```
pacman -S openssh
```

```
systemctl enable sshd
```

/*SSH es un protocolo de red que permite el intercambio de datos y comunicaciones sobre un canal seguro entre dos computadoras
/*Para navegar seguros en internet
/*Gracias a "systemd" podemos activar ese servicio

/*En el caso que uses una laptop, el controlador para el touchpad: (opcional)

```
pacman -S xf86-input-synaptics
```

/*También podemos cambiar la tipografía de la terminal: (opcional)
/*Hacemos un listado de todos los archivos disponibles

```
ls /usr/share/kbd/consolefonts
```

Y para seleccionar y cambiar el estilo de letra

```
setfont tcvn8x16.psf.gz
```

Instalación de Kernel Linux

Oficialmente compatibles

- **Estable** — La versión *vanilla* del kernel y los módulos, con pocas modificaciones aplicadas.
<https://www.kernel.org/> || [linux](#)
- **Hardened** — Un kernel de Linux enfocado en seguridad, aplica parches para mitigar la explotación en el kernel o en el espacio del usuario. También activa mas características de seguridad en comparación con [linux](#), entre otros: *namespaces*, *audit* y [SELinux](#).
<https://github.com/copperhead/linux-hardened> || [linux-hardened](#)
- **Larga duración** — Kernel de Linux y módulos con soporte de larga duración (LTS).
<https://www.kernel.org/> || [linux-lts](#)
- **Kernel ZEN** — Es el resultado de un esfuerzo colaborativo de varios hackers para hacer el mejor kernel para el uso en sistemas de uso diario.
<https://github.com/zen-kernel/zen-kernel> || [linux-zen](#)

Para Linux Estable:

```
pacman -S linux-firmware mkinitcpio linux linux-headers
```

Para Linux Hardened:

```
pacman -S linux-firmware mkinitcpio linux-hardened linux-hardened-headers
```

Para Linux LTS:

```
pacman -S linux-firmware mkinitcpio linux-lts linux-lts-headers
```

Para Linux Zen:

```
pacman -S linux-firmware mkinitcpio linux-zen linux-zen-headers
```

/*Mi recomendación es que si son nuevos es mejor **usar linux la versión estable**

/*Al usar un kernel personalizado como (Hardened o Zen) la instalación de drivers de video o red o wireless cambia por paquetes que terminan en **dkms** y algunas oportunidades fallan

Arch	Repo	Name	Version
x86_64	Community	acpi_call-dkms	1.1.0-264
x86_64	Community-Testing	acpi_call-dkms	1.1.0-265
x86_64	Community	bbswitch-dkms	0.8-269
x86_64	Community-Testing	bbswitch-dkms	0.8-270
x86_64	Community	broadcom-wl-dkms	6.30.223.271-20
x86_64	Community	deepin-anything-dkms	5.0.1-1
x86_64	Community	ndiswrapper-dkms	1.62-1
x86_64	Extra	nvidia-390xx-dkms	390.132-6
x86_64	Testing	nvidia-390xx-dkms	390.132-7
x86_64	Extra	nvidia-dkms	440.36-5
x86_64	Testing	nvidia-dkms	440.44-2
x86_64	Community	vhba-module-dkms	20190831-24
x86_64	Community-Testing	vhba-module-dkms	20190831-25
x86_64	Community	virtualbox-guest-dkms	6.1.0-1
x86_64	Community	virtualbox-host-dkms	6.1.0-1
x86_64	Community	wireguard-dkms	0.0.20191212-1

/*Al usar un kernel LTS sus drivers también son diferentes:

21 matching packages found.

Arch	Repo	Name	Version	Description
x86_64	Community	acpi_call-its	1.1.0-105	A linux kernel module that enables calls to ACPI methods through /proc/acpi/call
x86_64	Community-Testing	acpi_call-its	1.1.0-106	A linux kernel module that enables calls to ACPI methods through /proc/acpi/call
x86_64	Core	linux-its	4.19.88-1	The LTS Linux kernel and modules
x86_64	Testing	linux-its	4.19.89-1	The LTS Linux kernel and modules
x86_64	Core	linux-its-docs	4.19.88-1	Documentation for the LTS Linux kernel
x86_64	Testing	linux-its-docs	4.19.89-1	Documentation for the LTS Linux kernel
x86_64	Core	linux-its-headers	4.19.88-1	Headers and scripts for building modules for the LTS Linux kernel
x86_64	Testing	linux-its-headers	4.19.89-1	Headers and scripts for building modules for the LTS Linux kernel
x86_64	Community	nodejs-its-carbon	8.16.2-2	Evented I/O for V8 javascript (LTS release: Carbon)
x86_64	Community	nodejs-its-dubnium	10.17.0-2	Evented I/O for V8 javascript (LTS release: Dubnium)
x86_64	Community	nodejs-its-erbium	12.13.1-1	Evented I/O for V8 javascript (LTS release: Erbium)
x86_64	Extra	nvidia-390xx-its	1.390.132-5	NVIDIA drivers for linux-its, 390xx legacy branch
x86_64	Testing	nvidia-390xx-its	1.390.132-6	NVIDIA drivers for linux-its, 390xx legacy branch
x86_64	Extra	nvidia-its	1.440.36-4	NVIDIA drivers for linux-its
x86_64	Testing	nvidia-its	1.440.44-2	NVIDIA drivers for linux-its
x86_64	Community	r8168-its	8.047.05-3	A kernel module for Realtek 8168 network cards for linux-its
x86_64	Community-Testing	r8168-its	8.047.05-4	A kernel module for Realtek 8168 network cards for linux-its
x86_64	Community	tp_smapi-its	0.43-24	Modules for ThinkPad's SMAPi functionality
x86_64	Community-Testing	tp_smapi-its	0.43-25	Modules for ThinkPad's SMAPi functionality
x86_64	Community	wireguard-its	0.0.20191212-1	Wireguard module for LTS Kernel
x86_64	Community-Testing	wireguard-its	0.0.20191212-2	Wireguard module for LTS Kernel

/*Sin mencionar que esos kernel no siempre se actualizan inmediato
/*Suelen demorar en tener actualizaciones y hasta ese entonces rompe el sistema con otras dependencias, rara vez sucede, pero es algo que sucede...

mkinitcpio

/*Utilidad de creación de imágenes initramfs para el kernel

linux-firmware

/*El firmware de Linux es un paquete distribuido junto con el kernel de Linux que contiene blobs binarios de firmware necesarios para la funcionalidad parcial o total de ciertos dispositivos

/*Estos blobs binarios generalmente son propietarios porque algunos fabricantes de hardware no liberan el código fuente necesario para construir el firmware en sí.

/*Las tarjetas gráficas modernas de AMD, NVIDIA y Intel Wi-Fi requieren la carga de blobs para que el hardware funcione correctamente.

/*Después de la instalación del Kernel, Automático se crearan las **IMG de Linux** en la carpeta de **/boot/** gracias a **mkinitcpio**

```
(5/5) Updating linux initcpio...
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'default'
-> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux.img
==> Starting build: 5.4.2-arch1-1
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [autodetect]
-> Running build hook: [modconf]
-> Running build hook: [block]
-> Running build hook: [filesystems]
-> Running build hook: [keyboard]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip-compressed initcpio image: /boot/initramfs-linux.img
==> Image generation successful
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'fallback'
-> -k /boot/vmlinuz-linux -c /etc/mkinitcpio.conf -g /boot/initramfs-linux-fallback.img -S autodetect
==> Starting build: 5.4.2-arch1-1
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [modconf]
-> Running build hook: [block]
==> WARNING: Possibly missing firmware for module: wd719x
==> WARNING: Possibly missing firmware for module: aic94xx
-> Running build hook: [filesystems]
-> Running build hook: [keyboard]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating gzip-compressed initcpio image: /boot/initramfs-linux-fallback.img
==> Image generation successful
[root@archiso ~]#
[root@archiso ~]# ls /boot/
initramfs-linux-fallback.img  initramfs-linux.img  lost+found  nombre_de_usuario  vmlinuz-linux
[root@archiso ~]#
```

GRUB - Gestor de Arranque (UEFI)

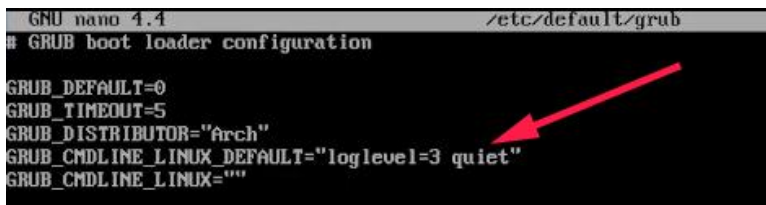
```
pacman -S grub efibootmgr os-prober
```

```
grub-install --target=x86_64-efi --efi-directory=/boot --bootloader-id=Arch
```

```
grub-install --target=x86_64-efi --efi-directory=/boot --removable
```

```
[root@archiso /]#  
[root@archiso /]# grub-install --target=x86_64-efi --efi-directory=/boot --bootloader-id=Arch  
Instalando para plataforma x86_64-efi.  
Instalación terminada. No se notificó ningún error.  
[root@archiso /]#  
[root@archiso /]# grub-install --target=x86_64-efi --efi-directory=/boot --removable  
Instalando para plataforma x86_64-efi.  
Instalación terminada. No se notificó ningún error.  
[root@archiso /]#
```

```
nano /etc/default/grub
```



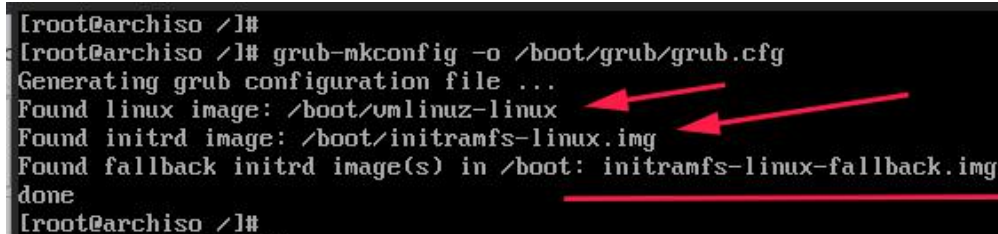
```
GNU nano 4.4 /etc/default/grub  
# GRUB boot loader configuration  
  
GRUB_DEFAULT=0  
GRUB_TIMEOUT=5  
GRUB_DISTRIBUTOR="Arch"  
GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3 quiet"  
GRUB_CMDLINE_LINUX=""
```

- /*Borramos **quiet** ese parámetro para que muestre todo el texto de inicio del sistema
- /*Podemos **cambiar el tiempo de espera de Grub** en **TIMEOUT=5** y poner **otro valor de seg.**



```
[ OK ] Started Rebuild Journal Catalog.  
[ OK ] Started Rebuild Dynamic Linker Cache.  
Starting Update is Completed...  
[ OK ] Started Update is Completed.  
[ OK ] Reached target System Initialization.  
[ OK ] Started Daily rotation of log files.  
[ OK ] Started Daily man-db regeneration.  
[ OK ] Started Daily verification of password and group files.  
[ OK ] Started Daily Cleanup of Temporary Directories.  
[ OK ] Reached target Timers.  
[ OK ] Listening on D-Bus System Message Bus Socket.  
[ OK ] Reached target Sockets.  
[ OK ] Reached target Basic System.  
[ OK ] Started D-Bus System Message Bus.  
Starting Network Manager...  
Starting dhcpcd on all interfaces...  
Starting Login Service...  
Starting Rotate log files...  
Starting Daily man-db regeneration...  
[ OK ] Started Verify integrity of password and group files.  
[ OK ] Started dhcpcd on all interfaces.  
[ OK ] Started Rotate log files.  
[ OK ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Watch.  
[ OK ] Started Login Service.  
[ OK ] Started Network Manager.  
[ OK ] Reached target Network.  
[ OK ] Started OpenSSH Daemon.  
Starting Permit User Sessions...  
[ OK ] Started Permit User Sessions.  
[ OK ] Started Getty on tty1.  
[ OK ] Reached target Login Prompts.  
[ OK ] Reached target Multi-User System.  
[ OK ] Reached target Graphical Interface.  
Starting Hostname Service...  
[ OK ] Started Hostname Service.  
Starting Network Manager Script Dispatcher Service...  
[ OK ] Started Network Manager Script Dispatcher Service.
```

```
grub-mkconfig -o /boot/grub/grub.cfg
```



```
[root@archiso /]#  
[root@archiso /]# grub-mkconfig -o /boot/grub/grub.cfg  
Generating grub configuration file ...  
Found linux image: /boot/vmlinuz-linux  
Found initrd image: /boot/initramfs-linux.img  
Found fallback initrd image(s) in /boot: initramfs-linux-fallback.img  
done  
[root@archiso /]#
```

- /*Como vemos GRUB detecta los archivos **.img** que están en **/boot/**
- /*os-prober detecta los sistemas operativos que usas y que usará GRUB para su menú

/*Antes de reiniciar revisamos si en Ethernet tenemos Gigabit RTL 8111/8168/8411 con:

```
lspci | grep Ethernet
```

```
[cristo@alv:~]$ lspci | grep Ethernet (12-14 00:17)
04:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411
PCI Express Gigabit Ethernet Controller (rev 0c)
[cristo@alv:~]$ (12-14 00:17)
```

/*Si tenemos Gigabit RTL 8111/8168/8411 el kernel Linux usara el módulo r8169

/*Lo que haremos es instalar el driver R8168

/*Y bloquearemos el acceso a R8169

```
nano /etc/modprobe.d/r8169.conf
```

```
GNU nano 4.6 /etc/modprobe.d/r8169.conf
blacklist r8169
```

```
blacklist r8169
```

/*Y luego instalamos el Driver R8168

```
pacman -S r8168
```

/*Entonces el kernel Linux empezara a usar el driver r8168 que es para el Ethernet

/*Salimos con:

```
exit
exit
```

```
[root@archiso /]# exit
exit
arch-chroot /mnt 22.03s user 5.81s system 4% cpu 9:38.60 total
root@archiso ~ # exit
```

/*Salimos del sistema, desmontamos todas las particiones...

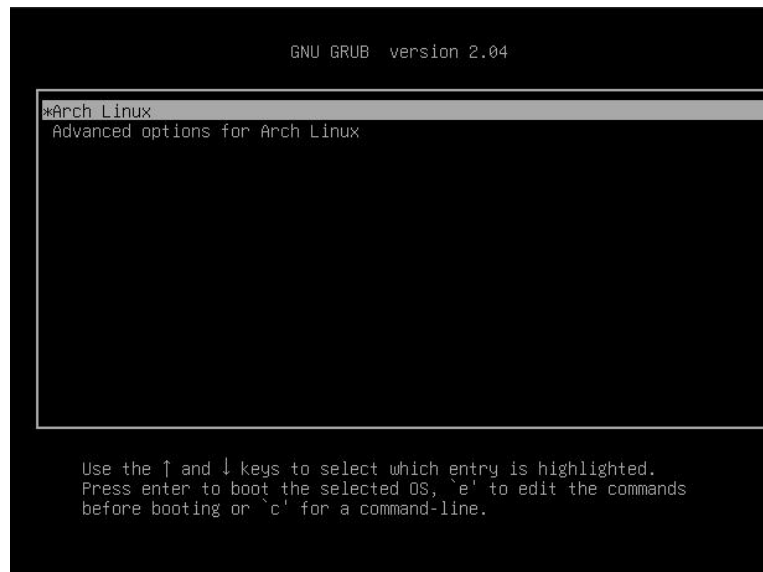
```
umount -R /mnt
```

```
reboot
```

/*Y finalmente reiniciamos, retiramos la USB o el CD cuando este apagada la pc

/*Entramos como usuario ROOT

/* Listo Arch Linux - UEFI */



```
Arch Linux 5.4.2-arch1-1 (tty1)
nombredepc login: root
Password:
[root@nombredepc ~]# uname -a
Linux nombredepc 5.4.2-arch1-1 #1 SMP PREEMPT Thu, 05 Dec 2019 12:29:40 +0000 x86_64 GNU/Linux
[root@nombredepc ~]#
[root@nombredepc ~]# _
```

Después de reiniciar

/*Entramos como ROOT y nos conectamos a internet

/*Con la conexión cableada solo basta tener cable conectado y activado el servicio:

systemctl enable dhcpcd

/*También podemos usar ifplugd o interfaces de red en general:

ip link

```
[cristo@alv:~]$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether fc:aa:14:6c:e7:6a brd ff:ff:ff:ff:ff:ff
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether 18:a6:f7:9f:ab:aa brd ff:ff:ff:ff:ff:ff
```

systemctl enable ifplugd@enp4s0

*O usando Wi-Fi: (Algunos casos es necesario reiniciar)

systemctl enable ifplugd@wlp3s0

/*Si estamos usando Wi-Fi:

```
nmcli dev wifi list
```

```
[root@alv cristo]# nmcli dev wifi list
IN-USE  SSID             MODE   CHAN  RATE        SIGNAL  BARS  SECURITY
        Secta_Alvarado  Infra  11    130 Mbit/s  87      ████  WPA1 WPA2
        VADEAN         Infra  1     54 Mbit/s   49      ███   WPA2
        Valalo         Infra  5     54 Mbit/s   35      ███   WPA2
        MOVISTAR_7DB0  Infra  6     130 Mbit/s  22      ███   WPA1 WPA2
[root@alv cristo]#
```

*Usamos la **comillas simple** si es una red con espacios o con caracteres especiales:

```
nmcli dev wifi connect 'NOMBRE DE RED' password CLAVE
```

/*Verificamos la conexión:

```
nmcli dev status
```

```
[root@alv cristo]# nmcli dev status
DEVICE    TYPE    STATE      CONNECTION
enp4s0    ethernet conectado   enp4s0
wlan0     wifi    conectando (obteniendo configuración IP) Secta_Alvara
p2p-dev-wlan0 wifi-p2p desconectado --
lo        loopback sin gestión --
[root@alv cristo]#
```

/*Y en el ultimo caso que aún así no tengan Wifi activen *ifplug* para que arranque únicamente esa interfaz de internet al iniciar el sistema.

```
systemctl enable ifplugd@wlp3s0
```

/* *ifplug* es un demonio de systemd que configurará automáticamente su dispositivo Ethernet cuando se conecta un cable y desconfigurará automáticamente si se tira del cable. Sucede lo mismo en Wlan.

```
ping -c 3 archlinux.org
```

```
Last login: Sat Dec 14 05:09:33 on tty1
[root@nombredepc ~]#
[root@nombredepc ~]# ping -c 3 archlinux.org
PING archlinux.org (138.201.81.199) 56(84) bytes of data:
64 bytes from apollo.archlinux.org (138.201.81.199): icmp_seq=1 ttl=63 time=243 ms
64 bytes from apollo.archlinux.org (138.201.81.199): icmp_seq=2 ttl=63 time=242 ms
64 bytes from apollo.archlinux.org (138.201.81.199): icmp_seq=3 ttl=63 time=242 ms
--- archlinux.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 242.147/242.443/242.872/0.310 ms
[root@nombredepc ~]#
```

/*Nos aseguramos que tenemos internet

/*Estamos listos para iniciar hacer descargas en nuestro sistema

/*Pero siempre antes de hacer alguna instalación siempre actualizamos el sistema con:

```
pacman -Syu
```

Personalizando PACMAN

/*Configuración:

El archivo general de configuración de pacman se encuentra en

```
nano /etc/pacman.conf
```

/*Quitamos # de comentario

```
# Misc options
#UseSyslog
Color
TotalDownload
CheckSpace
#VerbosePkgLists
```

/*Y agregamos: (es una i mayúscula y luego una L)

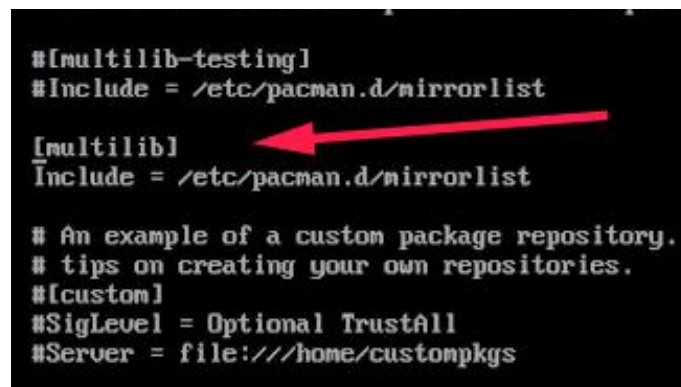
```
ILoveCandy
```



```
GNU nano 4.6 /etc/pacman.conf
# Misc options
#UseSyslog
Color
TotalDownload
CheckSpace
#VerbosePkgLists
ILoveCandy
```

/*Luego quitamos el # comentario en el repositorio de MultiLib

```
[multilib]
Include = /etc/pacman.d/mirrorlist
```



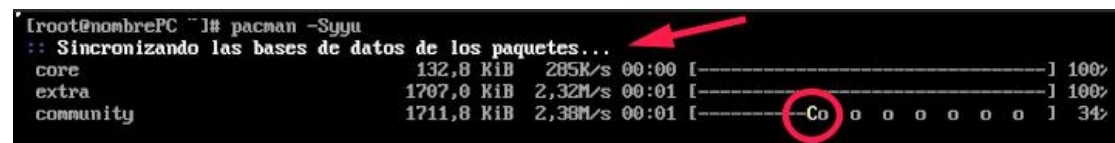
```
#[multilib-testing]
#Include = /etc/pacman.d/mirrorlist

[multilib]
Include = /etc/pacman.d/mirrorlist

# An example of a custom package repository.
# tips on creating your own repositories.
#[custom]
#SigLevel = Optional TrustAll
#Server = file:///home/custompkgs
```

/*Actualizamos el sistema para ver el color y animación de pacman:

```
pacman -Syu
```



```
[root@nombrePC ~]# pacman -Syu
:: Sincronizando las bases de datos de los paquetes...
core               132,8 KiB   285K/s  00:00 [-----] 100%
extra             1707,0 KiB  2,32M/s  00:01 [-----] 100%
community         1711,8 KiB  2,38M/s  00:01 [---Co o o o o o o] 34%
```

Pacman

Pacman (acronimo de Package Manger) es el gestor de paquetes oficial para ArchLinux. Maneja paquetes precompilados que descarga de los repositorios oficiales a través de mirrors (o repositorios espejo).

El objetivo de pacman es hacer posible gestionar fácilmente los paquetes, tanto si son los de los repositorios oficiales de Arch como las compilaciones realizadas por los propios usuarios.

La configuración y el uso de pacman es, como muchas cosas en ArchLinux algo bastante simple.

Guía rápida de uso de pacman - Todos los comandos son como usuario root o con sudo:

*/*Sincroniza la base de datos con los repositorios.*

pacman -Sy

*/*Actualiza el sistema completo.*

pacman -Syu

*/*Instala un paquete.*

pacman -S paquete

*/*Desinstala un paquete.*

pacman -R paquete

*/*Desinstala un paquete junto a las dependencias no utilizadas por otros paquetes.*

pacman -Rs paquete

Para eliminar un paquete, sus dependencias y todos los paquetes que dependen del paquete:

pacman -Rsc paquete

*/*Permite buscar a un paquete oficial específico*

pacman -Ss paquete

*/*Muestra información sobre un paquete no instalado*

pacman -Si paquete

*/*Muestra información sobre un paquete ya instalado*

pacman -Qi paquete

*/*Instala solamente las dependencias del paquete.*

pacman -Se paquete

*/*Muestra todos los archivos pertenecientes al paquete.*

pacman -Ql paquete

*/*Muestra una lista de todos los paquetes instalados en el sistema.*

pacman -Q

*/*Muestra a cual paquete pertenece un archivo en especial.*

pacman -Qo /ruta/del/archivo

*/*Borra todos los paquetes guardados en la cache de pacman ubicado en /var/cache/pacman/pkg*

pacman -Scc

*/*Instala un paquete guardado en una carpeta local.*

pacman -U nombre_del_paquete.pkg.tar.xz

*/*Eliminación de paquetes no utilizados o que no usan por un largo tiempo*

pacman -Rns \$(pacman -Qtdq)

Repositorios

El archivo de configuración maneja su sección donde administra los repositorios de software de donde descargara paquetes, inicialmente serán estos.

-Core:

Cuenta con unos requisitos de calidad muy estrictos, es el repositorio central del sistema, contiene lo básico para tener un sistema corriendo, es el repositorio utilizado para instalar el sistema en el livecd-iso.

-Extra:

Contiene todos los paquetes que no se ajustan a los requisitos para estar en [core]. Por ejemplo: Xorg, gestores de ventanas, navegadores Web, reproductores multimedia, herramientas para trabajar y otros programas.

-Community:

Contiene los paquetes provenientes del repositorio Arch User Repository que obtuvieron los votos suficientes para ser adoptadas como Oficial mantenidos por usuarios oficiales que pasaron varias pruebas de dedicación y buen trabajo, los paquetes aquí listados eran PKGBUILDS de Arch User Repository (AUR)

-Community testing:

Repositorio de pruebas de community, aquí están los paquetes que muy probablemente pasaran a community pero no han sido probados lo suficiente o están inestables, solo paquetes de usuarios de confianza, deshabilitado por default.

-Testing:

Repositorio de prueba de los paquetes oficiales (mantenido por developers oficiales), contiene paquetes que son candidatos para los repositorios [core] o [extra]. Los paquetes aquí están en fase de constante desarrollo y son inestables, esta deshabilitado por default, no se recomienda su uso.

-Multilib:

Contiene el software de 32 bits y las bibliotecas que se pueden utilizar para ejecutar y desarrollar aplicaciones de 32 bits en instalaciones de 64 bits

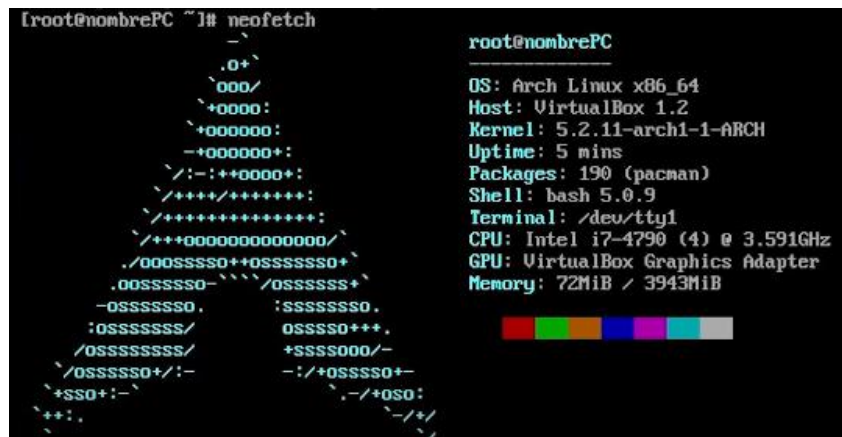
-Multilib-testing:

Repositorio de pruebas de multilib

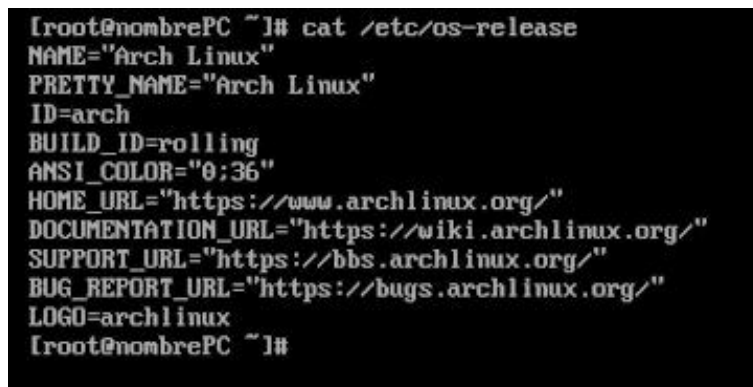
```
pacman -S  git  wget
```

```
/* Descargas con git
/* Descargas con wget
```

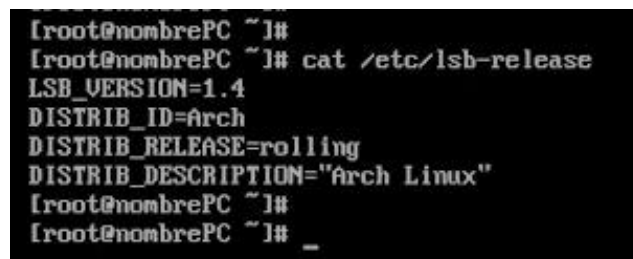
```
pacman -S  neofetch  lsb-release
```



```
/*programa de consulta de versión como nuestro ID de distro
/*os-release - lsb_release
```



```
/*por defecto neofetch instala os-release, pero ciertas aplicaciones
/*solicita información a lsb_release (steam, etc)
```



/*Los fabricantes de procesadores lanzan actualizaciones de
/*estabilidad y seguridad para el microcódigo del procesador

/*El kernel de Linux también puede aplicar estas actualizaciones durante el arranque

```
pacman -S amd-ucode
```

```
pacman -S intel-ucode
```

/*No tenemos los directorios comunes:

/*Escritorio-Documentos-Descargas-Música-Imágenes-Public-Plantillas-Videos

```
[root@nombrePC ~]#  
[root@nombrePC ~]# ls  
[root@nombrePC ~]# ls --all  
.  ..  .bash_history  .cache  .config  .gnupg  
[root@nombrePC ~]# _
```

/*xdg-user-dirs es una herramienta para ayudar a

/*administrar directorios de usuarios

```
pacman -S xdg-user-dirs
```

```
[root@nombrePC ~]# ls  
[root@nombrePC ~]# ls --all  
.  ..  .bash_history  .cache  .config  .gnupg  
[root@nombrePC ~]# pacman -S xdg-user-dirs  
resolviendo dependencias...  
buscando conflictos entre paquetes...  
  
Paquetes (1) xdg-user-dirs-0.17-2  
  
Tamaño total de la descarga:  0,04 MiB  
Tamaño total de la instalación: 0,15 MiB  
  
:: ¿Continuar con la instalación? [S/n] _
```

/*La forma en que funciona es que xdg-user-dirs-update se ejecuta

/*Luego crea versiones localizadas de estos directorios

/*en el directorio de inicio de los usuarios con iconos especiales

```
xdg-user-dirs-update
```

```
[root@nombrePC ~]#  
[root@nombrePC ~]# xdg-user-dirs-update  
[root@nombrePC ~]#  
[root@nombrePC ~]#  
[root@nombrePC ~]# ls  
Descargas Documentos Escritorio Imágenes Música Plantillas Público Vídeos  
[root@nombrePC ~]#  
[root@nombrePC ~]#
```

X.Org Server

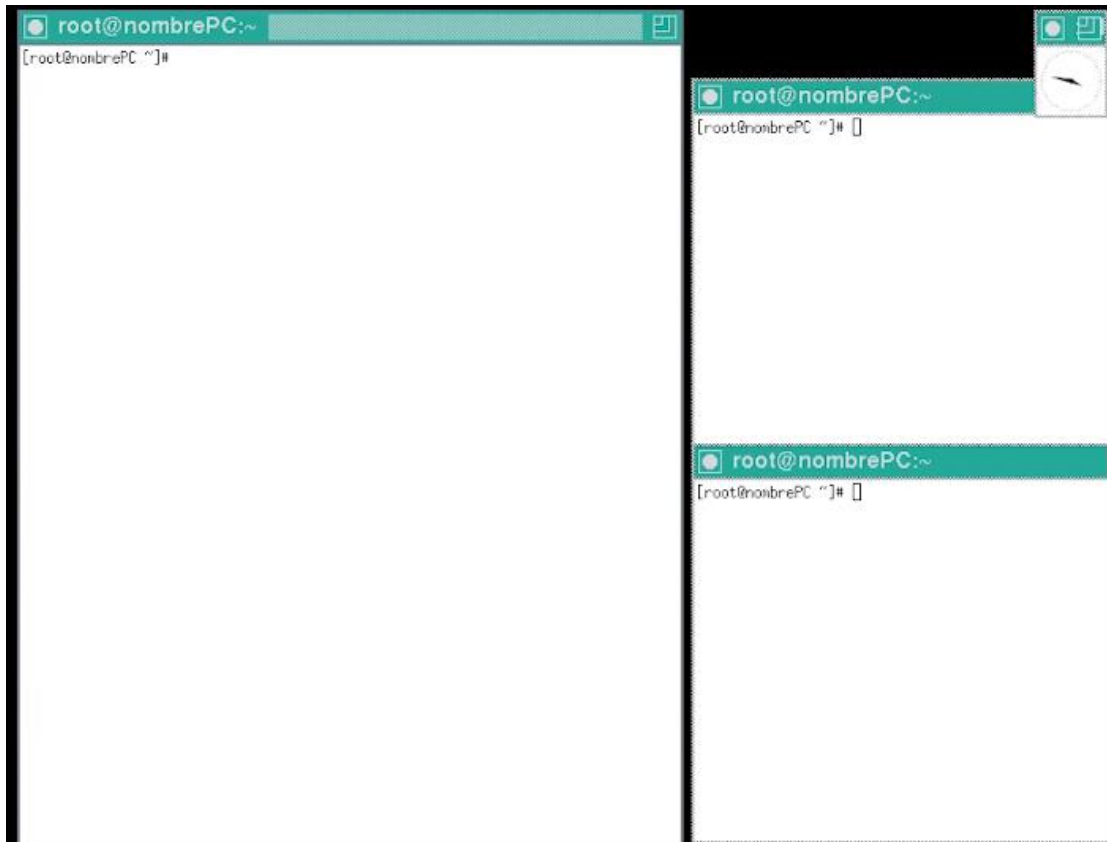
/*X.Org Server proporciona las herramientas estándar
/*para proveer de interfaces gráficas

```
pacman -S xorg xorg-apps xorg-xinit xorg-twm xterm xorg-xclock
```

/*Para ejecutar X.org

```
startx
```

/*Para salir escribir **exit** y **enter**



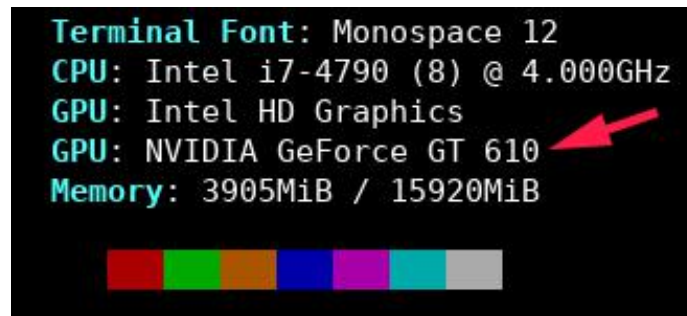
Driver de Vídeo

/*Con el siguiente comando nos dará información de nuestra tarjeta gráfica

```
lspci | grep VGA
```

```
[cristo@alv:~]$ lspci | grep VGA (09-15)
00:02.0 VGA compatible controller: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor Integrated Graphics Controller (rev 06)
01:00.0 VGA compatible controller: NVIDIA Corporation GF119 [GeForce GT 610] (rev a1) (09-15)
[cristo@alv:~]$
```

/*O también en ****neofetch**** en la parte que dice GPU...



```
Terminal Font: Monospace 12
CPU: Intel i7-4790 (8) @ 4.000GHz
GPU: Intel HD Graphics
GPU: NVIDIA GeForce GT 610
Memory: 3905MiB / 15920MiB
```

/*Aquí solo instalaren drivers de código libre, **subí otro pdf a más detalle**

/*La instalación de driver privado de Nvidia, aceleración por hardware y GPGPU

/*Para tarjetas de Intel con soporte a Vulkan (gaming):

```
pacman -S mesa lib32-mesa
```

```
pacman -S xf86-video-intel vulkan-intel
```

/*Para tarjetas de AMD/Radeon con soporte a Vulkan (gaming):

/*Series 200 para arriba

/*(Tarjetas modernas)

```
pacman -S mesa lib32-mesa
```

```
pacman -S xf86-video-amdgpu
```

```
pacman -S vulkan-radeon lib32-vulkan-radeon
```

/*Para tarjetas de AMD/ATI no soporta Vulkan:

/*(Tarjetas Antiguas - Lo he probado en gráficas muy antiguas y he podido ver vídeos en HD sin problemas)

```
pacman -S mesa lib32-mesa
```

```
pacman -S xf86-video-ati
```

/*Para tarjetas de Nvidia Código libre no soporta Vulkan:

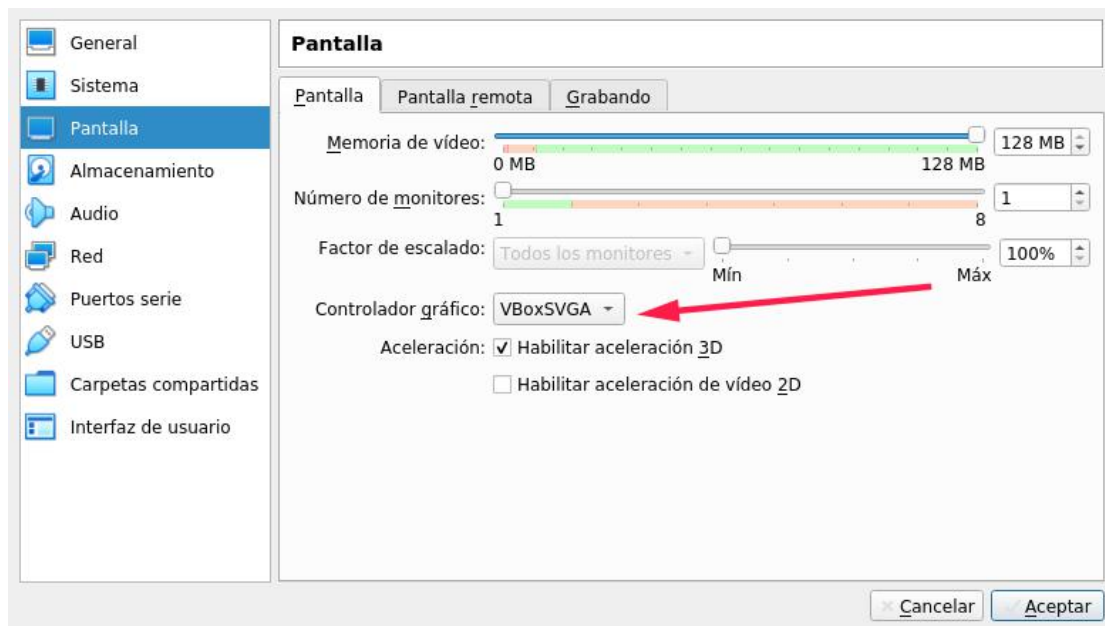
```
pacman -S mesa lib32-mesa
```

```
pacman -S xf86-video-nouveau
```

/*En este caso solo **estamos usando VirtualBox**

/*Igual podemos instalar un módulo de VirtualBox para que se ajuste la pantalla en automático

/*Hay que asegurarse de tener VBoxSVGA



```
pacman -S virtualbox-guest-utils
```

```
[root@nombrePC ~]#  
[root@nombrePC ~]#  
[root@nombrePC ~]# pacman -S virtualbox-guest-utils  
resolviendo dependencias...  
:: Existen 2 proveedores disponibles para VIRTUALBOX-GUEST-MODULES:  
:: Repositorio community  
   1) virtualbox-guest-dkms  2) virtualbox-guest-modules-arch  
Introduzca un número (por omisión=1):
```

/*En ArchLinux cuando tenemos DKMS y Modules-Arch

/***DKMS es cuando tienes otro kernel de linux**

/*(linux-hardened, linux-lts, linux-zen)

/***Modules-Arch es para kernel Linux - seleccionamos 2**

```
systemctl enable vboxservice
```

```
systemctl status vboxservice
```

/*En el caso que no tengan ni Intel, ni AMD, ni Nvidia entonces el driver genérico

```
pacman -S xf86-video-vesa
```

/*Finalmente falta instalar un Display Manager (DM) y un Escritorio (DE)

/*Con programas extras, tengo otros PDF por separados