

# INSTALACIÓN PURA (BIOS/ UEFI)

## 1. Teclado e internet

### 1.1. Establecer un teclado para toda la instalación:

#### 1.1.1. Elegir entre los principales modelos:

Para el teclado español:

```
loadkeys es
```

Para el teclado latinoamericano:

```
loadkeys la-latin1
```

Para el teclado inglés:

```
loadkeys us
```

### 1.2. Establecer una conexión de red:

1.2.1. En caso de estar conectado directamente a la red (Ethernet), omitir este paso.

1.2.1. Conectarse a una red de forma inalámbrica:

1.2.1.1. Identificar la tarjeta de red: `ip link`

```
2 root@archiso ~ # ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp2s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN mode DEFAULT group default qlen 1000
   link/ether [redacted]
4: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000
   link/ether [redacted]
```

*Figura 1: Tarjetas de red.*

Al ejecutar el comando, éste nos dirá las conexiones que están disponibles así como el tipo de conexión que tiene cada una, en el ejemplo nos centraremos en la que dice “wlan0” debido a que ese es el tipo de red que se usará, esto ya depende de la conexión a internet que tengamos, sin embargo “wlan0” es el más común.

1.2.1.2. Establecer la conexión:

```
iwctl --passphrase $$pass$$ station wlan0 connect “Juan 1”
```

```
root@archiso ~ # iwctl --passphrase CONTRASEÑA station wlan0 connect "Nombre_de_la_red"
```

Contraseña sin comillas

Tipo de red

Nombre de la red entre comillas

Al colocar el nombre de la red, en caso de que tenga espacios, los mismo se respetan; ejemplo: "Juan 2"

Figura 2: Conexión inalámbrica.

1.2.1.3. Comprobación de la conexión a internet: `ping -c 4`

`google.com`

```
root@archiso ~ # ping -c 4 google.com
PING google.com (172.217.15.206) 56(84) bytes of data.
64 bytes from mia09s20-in-f14.1e100.net (172.217.15.206): icmp_seq=1 ttl=112 time=57.9 ms
64 bytes from mia09s20-in-f14.1e100.net (172.217.15.206): icmp_seq=2 ttl=112 time=57.6 ms
64 bytes from mia09s20-in-f14.1e100.net (172.217.15.206): icmp_seq=3 ttl=112 time=61.3 ms
64 bytes from mia09s20-in-f14.1e100.net (172.217.15.206): icmp_seq=4 ttl=112 time=57.6 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3011ms
rtt min/avg/max/mdev = 57.567/58.598/61.303/1.566 ms
root@archiso ~ #
```

Figura 3: Verificación de conexión.

## 2. Particiones de disco

### 2.0.1. Particionado de disco en UEFI:

#### 2.1. Verificación de placa tipo "GPT": `fdisk -l`

```
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
```

Figura 4: Tipo de placa de memoria.

#### 2.2. Verificamos que la placa base sea del tipo UEFI:

`ls /sys/firmware/efi/efivars`

```

root@archiso ~ # ls /sys/firmware/efi/efivars
Boot0000-8be4df61-93ca-11d2-aa0d-00e098032b8c
Boot0001-8be4df61-93ca-11d2-aa0d-00e098032b8c
Boot0002-8be4df61-93ca-11d2-aa0d-00e098032b8c
Boot0003-8be4df61-93ca-11d2-aa0d-00e098032b8c
BootCurrent-8be4df61-93ca-11d2-aa0d-00e098032b8c
BootOptionSupport-8be4df61-93ca-11d2-aa0d-00e098032b8c
BootOrder-8be4df61-93ca-11d2-aa0d-00e098032b8c
ConInDev-8be4df61-93ca-11d2-aa0d-00e098032b8c
ConIn-8be4df61-93ca-11d2-aa0d-00e098032b8c
ConOutDev-8be4df61-93ca-11d2-aa0d-00e098032b8c
ConOut-8be4df61-93ca-11d2-aa0d-00e098032b8c
ConsoleOutMode-793d9786-44dc-4709-b57f-85b8e8fdbfd2
ErrOut-8be4df61-93ca-11d2-aa0d-00e098032b8c
KEKDefault-8be4df61-93ca-11d2-aa0d-00e098032b8c
KEK-8be4df61-93ca-11d2-aa0d-00e098032b8c
LangCodes-8be4df61-93ca-11d2-aa0d-00e098032b8c

```

Figura 5: Información de la placa base: UEFI

Y debe aparecer un texto como el que se presenta en al Fig. 4

## 2.4. Listar los discos físicos existentes:

```

root@archiso ~ # parted -l | egrep "Model|dev/sd|msdos|gpt"
Error: /dev/sda: unrecognised disk label
Warning: Unable to open /dev/sr0 read-write (Read-only file system). /dev/sr0
has been opened read-only.
Error: Can't have a partition outside the disk!
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda 26.8GB

```

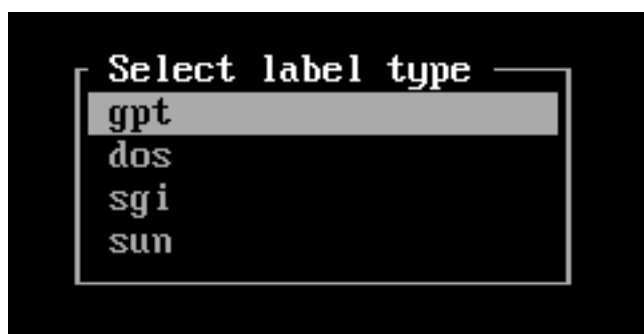
Aquí aparece los discos físicos de la pc, elegimos donde vamos a particionar el disco.

Figura 6: Lista discos físicos

## 2.3. Tabla de particiones con la herramienta cfdisk: cfdisk /dev/sda

isk /dev/sda

### 2.3.1. Seleccionamos: GPT.



### 2.3.2. Observamos la tabla creada:

Disk: /dev/sda					
Size: 25 GiB, 26843545600 bytes, 52428800 sectors					
Label: gpt, identifier: 93A75094-3B7E-1C44-BE94-EB2D614B4640					
Device	Start	End	Sectors	Size	Type
>> Free space	2048	52428766	52426719	25G	

**Para realizar un correcto particionado, ya debemos saber las especificaciones esenciales del pc, como lo es:**

- **Memoria ram.**- En el caso de tener una memoria ram de 8 Gigas, se recomienda asignar la mitad, es decir 4 Gigas, ésta nos servirá como memoria de intercambio, algo casi indispensable en el funcionamiento del sistema.

- **Almacenamiento.**- Aquí debemos identificar todas las particiones que están en el sistema, debido a que en la mayoría de casos, se suele usar dos sistemas a la vez, en caso de tener windows, es menéster conocer en qué particiones se encuentra ubicado, esto con la finalidad de evitar dañar los otros sistemas.

*Figura 7: Aclaraciones antes del particionado.*

## 2.4. Creación de particiones para el sistema.

### 2.4.1. EFI (arranque)

#### 2.4.1.1. Damos a "New":



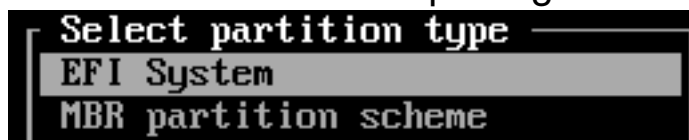
#### 2.4.1.2. Escribimos: 300M (Donde "M" simboliza Mebytes)



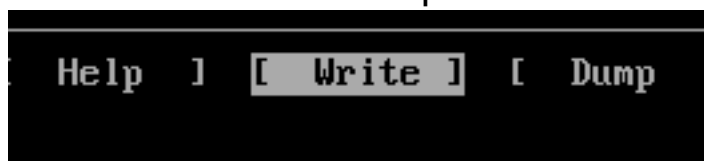
#### 2.4.1.3. Seleccionamos el Tipo: (Type)



#### 2.4.1.4. Buscamos al que diga "EFI System":



#### 2.4.1.5. Escribimos la partición "Write":



#### 2.4.1.6. Tecleamos "yes" para guardar el cambio:

```
Partition UUID: 3591A5AD-C501-0845-B8C3-B794BF8F4177
Partition type: EFI System (C12A7328-F81F-11D2-BA4B-00A0C93EC93B)
```

```
Are you sure you want to write the partition table to disk? yes_
```

Type "yes" or "no", or press ESC to leave this dialog.

Seleccionamos el espacio libre, "Free space" para continuar con las demás particiones.

## 2.4.2. Raíz (/)

2.4.2.1. Damos a "New":

```
[ New ] [ Quit ]
```

2.4.2.2. Tecleamos la memoria que tengamos disponible, dejando espacio para la partición de intercambio:

20.7G (Donde "G" simboliza Gigabytes)

```
Partition size: 20.7G_
```

2.4.2.3. Escribimos directamente la partición con "Write":

```
Help ] [ Write ] [ Dump ]
```

2.4.2.4. Tecleamos "yes" para guardar el cambio:

```
Partition UUID: 3E3BE1D8-76F9-6E43-81ED-3A1AAA621AB9
Partition type: Linux filesystem (0FC63DAF-8483-4772-8E79-3D69D8477DE4)
```

```
Are you sure you want to write the partition table to disk? yes
```

Type "yes" or "no", or press ESC to leave this dialog.

Seleccionamos el espacio libre, "Free space" para continuar última partición, en caso de necesitar una partición "home", repetir 2.4.2.

## 2.4.3. Swap (intercambio)

2.4.3.1. Damos a "New":

```
[ New ] [ Quit ]
```

2.4.3.2. Seleccionamos toda la memoria que quede:

3.7G (Donde "G" simboliza Gigabytes)

```
Partition size: 3.7G_
```

2.4.3.3. Seleccionamos el Tipo: (Type)

```
0FACF215
483-4772-8E79-3D69D8477DE4)

[ Quit ] [ Type ] [ Help ]
```

2.4.3.4. Buscamos al que diga “Linux Swap”:

```
HP-UX data
HP-UX service
Linux swap
Linux filesystem
Linux server data
```

2.4.3.5. Escribimos la partición “Write”:

```
Help ] [ Write ] [ Dump ]
```

2.4.3.6. Tecleamos “yes” para guardar el cambio:

```
4-5602-ED49-A137-36400FACF215
wap (0657FD6D-A4AB-43C4-84E5-0933C84B4F4F)

write the partition table to disk? yes
```

**2.4.4. Tomamos una foto o anotamos los nombres de cada partición:**

```

Disk: /dev/sda
Size: 25 GiB, 26843545600 bytes, 52428800 sectors
Label: gpt, identifier: 93A75094-3B7E-1C44-BE94-EB2D614B4640

Device            Start      End          Sectors      Size Type
/dev/sda1          2048      616447       614400      300M EFI System
/dev/sda2          616448    44656639    44040192     21G Linux filesystem
>> /dev/sda3       44656640    52428766    7772127      3.7G Linux swap
```

*Figura 8: Particiones de disco.*

**2.4.5. Salir con Quit:**

```
-A4AB-43C4-84E5-0933C84B4F4F

esize ] [ Quit ] [ Type ]
```

## 2.5. Particionado de disco en BIOS:

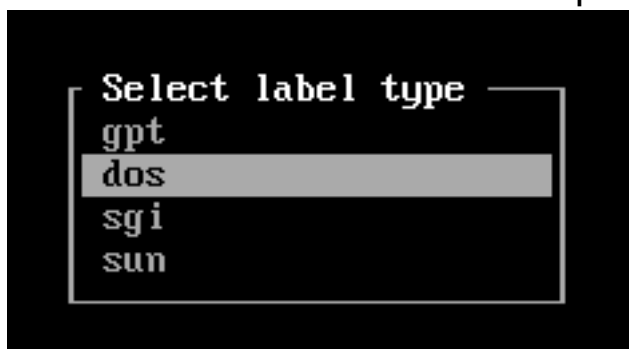
2.5.1. Listamos los discos/particiones existentes:

```
root@archiso ~ # fdisk -l
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors
```

2.5.2. Con “cfdisk” vamos a crear la tabla de particiones del disco.

```
root@archiso ~ # cfdisk_
```

2.5.3. Seleccionamos “dos” para el tipo de particionado.



2.5.4. Vemos la tabla de particiones:

Disk: /dev/sda						
Size: 30 GiB, 32212254720 bytes, 62914560 sectors						
Label: dos, identifier: 0x7bba22fd						
Device	Boot	Start	End	Sectors	Size	Id Type
>> Free space		2048	62914559	62912512	30G	

Y revisamos la información de la figura 1.

2.5.5. Creamos una partición de 300M para el "Boot": Del tipo “primaria”.

```
Partition size: 300M
```

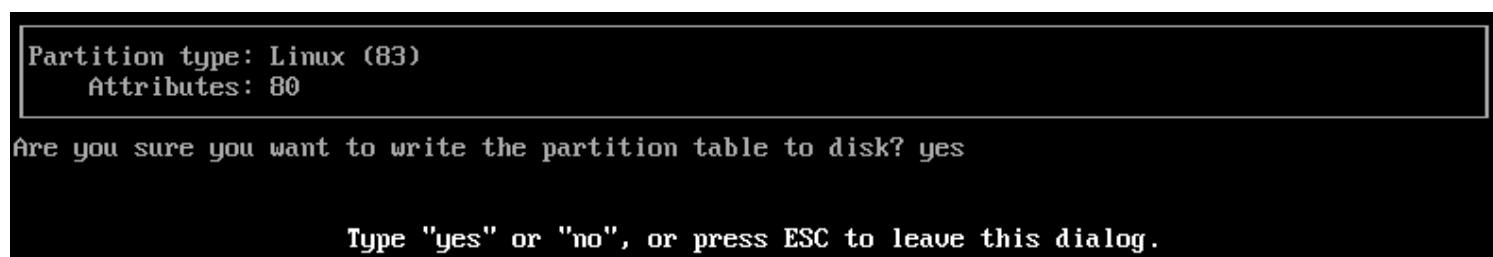
2.5.6. Elegimos el tipo de la partición: “Bootable”.



2.5.7. Seleccionando la partición, la “escribimos”:



2.5.8. Ponemos “yes” para efectuar el cambio:



**2.5.9. Creamos la partición “raíz” con 25,7G en el ejemplo: Del tipo primaria y escribimos directamente.**

```
Partition type: Linux (83)
[Bootable] [ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ] [ Write ] [ Dump ]
```

**2.5.10. Creamos la partición “swap” con 4G en el ejemplo: Del tipo primaria y damos en “type”:**

```
Partition type: Linux (83)
[Bootable] [ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ] [ Write ] [ Dump ]
```

**2.5.11. Seleccionamos el tipo: “82 Linux swap” y escribimos:**

```
65 Novell Netware 386
70 DiskSecure Multi-Boot
75 PC/IX
80 Old Minix
81 Minix / old Linux
82 Linux swap / Solaris
83 Linux
84 OS/2 hidden or Intel hibernation
85 Linux extended
86 NTFS volume set
87 NTFS volume set
```

**2.5.12. Tomamos una foto o anotamos las diferentes particiones:**

Device	Boot	Start	End	Sectors	Size	Id	Type
>> /dev/sda1	*	2048	616447	614400	300M	83	Linux
/dev/sda2		616448	55142399	54525952	26G	83	Linux
/dev/sda3		55142400	62914559	7772160	3.7G	82	Linux swap / Solaris

**2.5.13. Salimos con “Quit”:**

```
Partition type: Linux swap / Solaris (82)
[Bootable] [ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ] [ Write ] [ Dump ]
```

## ***3. Formato y montaje de particiones***

### **3.1. Formato de particiones UEFI:**

#### **3.1.1. Formato de la partición de arranque:**



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

```
root@archiso ~ # mkfs.fat -F 32 /dev/sda1  
mkfs.fat 4.2 (2021-01-31)
```

### 3.1.2. Formato de la partición root:

```
mkfs.ext4 /dev/sda2
```

```
root@archiso ~ # mkfs.ext4 /dev/sda2  
mke2fs 1.46.4 (18-Aug-2021)  
Creating filesystem with 5505024 4k blocks and 1376256 inodes  
Filesystem UUID: b38bd1ef-1250-4cdd-95f6-6d67fdc4c6cd  
Superblock backups stored on blocks:  
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,  
    4096000  
  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (32768 blocks): done  
Writing superblocks and filesystem accounting information: done
```

En caso de haber creado una partición home, repetir 3.2. con la partición home.

### 3.1.3. Formato de la partición de intercambio.

```
mkswap /dev/sda3
```

```
root@archiso ~ # mkswap /dev/sda3  
Setting up swspace version 1, size = 3.7 GiB (3979321344 bytes)  
no label, UUID=6a5b4fd8-95b5-406d-beda-ccc31b048115
```

### 3.1.4. Activar la partición swap recién formateada.

```
swapon /dev/sda3
```

```
root@archiso ~ # swapon /dev/sda3  
root@archiso ~ #
```

## 3.2. Montaje de particiones.

### 3.2.1. Montaje de la partición raíz:

```
mount /dev/sda2 /mnt
```

```
root@archiso ~ # mount /dev/sda2 /mnt  
root@archiso ~ #
```

### 3.2.2. Montaje de la partición efi:

```
mkdir -p /mnt/efi
```

```
root@archiso ~ # mkdir -p /mnt/efi  
root@archiso ~ #
```

### 3.2.3. Montaje de arranque

```
mount /dev/sda1 /mnt/efi
```

```
root@archiso ~ # mount /dev/sda1 /mnt/efi  
root@archiso ~ #
```

## 3.3. verificación de directorios:

```
ls /mnt/
```

```
root@archiso ~ # ls /mnt
efi  lost+found
```

## 3.2. Formato de particiones BIOS:

### 3.2.1. Formateo de las particiones.

#### 3.2.1.1. Formateo del arranque:

```
root@archiso ~ # mkfs.ext2 /dev/sda1
mke2fs 1.46.4 (18-Aug-2021)
Creating filesystem with 307200 1k blocks and 76912 inodes
Filesystem UUID: 1ad0c4ea-4ce2-43de-8aaf-302fac886c58
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

#### 3.2.1.2. Formateo de la raíz:

```
root@archiso ~ # mkfs.ext4 /dev/sda2
mke2fs 1.46.4 (18-Aug-2021)
Creating filesystem with 6815744 4k blocks and 1703936 inodes
Filesystem UUID: fb41600a-e7bf-4624-b900-179637030591
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

#### 3.2.1.3. Formateo del intercambio:

```
root@archiso ~ # mkswap /dev/sda3
Setting up swapspace version 1, size = 3.7 GiB (3979341824 bytes)
no label, UUID=5d8c512d-47ff-453b-9052-252d4b786446
```

#### 3.2.1.4. Formato a la particion de intercambio (swap):

```
root@archiso ~ # swapon
root@archiso ~ #
```

### 3.2.2. Montaje de las particiones:

#### 3.2.2.1 Montaje de la raíz:

```
mount /dev/sda2 /mnt/
```

#### 3.2.2.2. Creamos el punto de montaje para el boot:

```
mkdir -p /mnt/boot/
```

3.2.2.3. Creamos el punto de montaje para Bios en boot: (este paso es diferente en UEFI)

```
mount /dev/sda1 /mnt/boot
```

## 4. Paquetes base

### 4.1. Comando completo con los paquetes necesarios:

```
pacstrap /mnt base base-devel linux linux-firmware nano vim  
networkmanager network-manager-applet wireless_tools wpa_supplicant  
efibootmgr os-prober mtools dosfstools linux-headers grub dhcpcd  
netctl dialog gvfs xdg-user-dirs xf86-input-synaptics iw wireless-  
regdb xf86-input-libinput bluez bluez-utils pulseaudio-bluetooth
```

### 4.2. Generamos el archivo fstab: `genfstab -p /mnt >> /mnt/etc/fstab`

```
root@archiso ~ # genfstab -p /mnt >> /mnt/etc/fstab  
root@archiso ~ # _
```

### 4.3. Revisamos el archivo creado: `cat /mnt/etc/fstab`

```
root@archiso ~ # cat /mnt/etc/fstab  
# Static information about the filesystems.  
# See fstab(5) for details.  
  
# <file system> <dir> <type> <options> <dump> <pass>  
# UUID=b38bd1ef-1250-4cdd-95f6-6d67fdc4c6cd  
/dev/sda2 / ext4 rw,relatime 0 1  
  
# UUID=B824-3E7a  
/dev/sda1 /efi vfat rw,relatime,fnask=0022,dnask=0022,codepage=437,ioccharset=ascii,shortname  
=mixed,utf8,errors=remount-ro 0 2  
  
# UUID=6a5b4fd8-95b5-406d-beda-ccc31b048115  
/dev/sda3 none swap defaults 0 0  
  
root@archiso ~ #
```

## 5. Configuración básica

### 5.1. Nombre y propietario del sistema:

En este punto ya podemos ingresar al sistema como tal:

```
root@archiso ~ # arch-chroot /mnt  
[root@archiso /]#
```

Con el siguiente comando vamos a asignar el nombre del equipo "Host":

```
[root@archiso /]# echo Purito > /etc/hostname  
[root@archiso /]#
```

## 5.2. Zona horaria:

```
ln -sf /usr/share/zoneinfo/Región/Ciudad /etc/localtime
```

Tanto la región como la ciudad podemos averiguar con: **timedatectl list-timezones**

Un ejemplo sería el siguiente:

```
ln -sf /usr/share/zoneinfo/America/Argentina/Buenos_Aires /etc/  
localtime
```

O también:

```
ln -sf /usr/share/zoneinfo/America/Chicago /etc/localtime
```

## 5.3. Idioma del sistema:

editamos el archivo con:

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

Y quitamos “#” al idioma queramos para el sistema.

```
GNU nano 6.0 /etc/locale.gen  
#es_CL.UTF-8 UTF-8  
#es_CL ISO-8859-1  
#es_CO.UTF-8 UTF-8  
#es_CO ISO-8859-1  
#es_CR.UTF-8 UTF-8  
#es_CR ISO-8859-1  
#es_CU UTF-8  
#es_DO.UTF-8 UTF-8  
#es_DO ISO-8859-1  
#es_EC.UTF-8 UTF-8  
#es_EC ISO-8859-1  
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
```

Luego, guardamos los cambios con “Ctrl + X” y “Enter”.

## 5.4. Ahora agregamos las preferencias de la localización:

```
[root@archiso /]# echo LANG=es_EC.UTF-8 > /etc/locale.conf
[root@archiso /]# _
```

Para actualizar el idioma ya modificado, usamos:

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

## 5.5. Distribución de teclado:

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

## 5.6. Hora

Como ya hemos configurado anteriormente la zona horaria, actualizar la hora es fácil, basta con colocar:

```
[root@archiso /]# hwclock -w
[root@archiso /]#
```

Verificamos con:

```
[root@archiso /]# date
Mon Dec 20 19:05:45 -05 2021
```

Caso contrario instalamos el paquete:

`pacman -S ntp`

y ejecutamos:

`ntpd -qq`

```
[root@archiso /]# ntpd -qq
25 Dec 09:11:03 ntpd[511]: Jul 1 17:02:17 UTC 2020 (1): Starting
25 Dec 09:11:03 ntpd[511]: Command line: ntpd -qq
25 Dec 09:11:03 ntpd[511]: -----
25 Dec 09:11:03 ntpd[511]: ntp-4 is maintained by Network Time Foundation,
25 Dec 09:11:03 ntpd[511]: Inc. (NTF), a non-profit 501(c)(3) public-benefit
25 Dec 09:11:03 ntpd[511]: corporation. Support and training for ntp-4 are
25 Dec 09:11:03 ntpd[511]: available at https://www.ntptime.org/support
25 Dec 09:11:03 ntpd[511]: -----
25 Dec 09:11:03 ntpd[511]: proto: precision = 0.116 usec (-23)
25 Dec 09:11:03 ntpd[511]: basedate set to 2020-06-19
25 Dec 09:11:03 ntpd[511]: gps base set to 2020-06-21 (week 2111)
25 Dec 09:11:03 ntpd[511]: Listen and drop on 0 v6wildcard [::]:123
25 Dec 09:11:03 ntpd[511]: Listen and drop on 1 v4wildcard 0.0.0.0:123
25 Dec 09:11:03 ntpd[511]: Listen normally on 2 [redacted]
25 Dec 09:11:03 ntpd[511]: Listen normally on 3 [redacted]
25 Dec 09:11:03 ntpd[511]: Listen normally on 4 lo [::1]:123
25 Dec 09:11:03 ntpd[511]: Listen normally on 5 [redacted]
25 Dec 09:11:03 ntpd[511]: Listening on routing socket on fd #22 for interface updates
25 Dec 15:36:19 ntpd[511]: ntpd: time set +[redacted] s
ntpd: time set +[redacted] s
```

y actualizamos:

`hwclock -w`

```
[root@archiso /]# hwclock -w
[root@archiso /]# date
Sat Dec 25 15:36:31 -05 2021
```

## 6. GRUB

### 6.1. Instalación grub:

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

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

### 6.2. Cambiar el tiempo de inicio y borramos quiet en el Grub:

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

```
GNU nano 6.0 /etc/default/grub
# GRUB boot loader configuration

GRUB_DEFAULT=0
GRUB_TIMEOUT=3
GRUB_DISTRIBUTOR="Arch"
GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3"
GRUB_CMDLINE_LINUX=""

# Preload both GPT and MBR modules so that they are not missed
GRUB_PRELOAD_MODULES="part_gpt part_msdos"
```

### 6.3. Actualizar los parámetros modificados.

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

## 7. Usuarios

### 7.1 Clave de usuario (root):

Este ya se creó por defecto solo falta asignar una clave.

```
[root@archiso /]# passwd
New password:
Retype new password:
passwd: password updated successfully
[root@archiso /]#
```

## 7.2. Usuario normal:

```
useradd -m -g users -G wheel -s /bin/bash usuario
```

```
[root@archiso /]# useradd -m -g users -G wheel -s /bin/bash erick
[root@archiso /]#
```

```
passwd usuario
```

```
[root@archiso /]# passwd erick
New password:
Retype new password:
passwd: password updated successfully
```

## 7.3. Asignar permisos administrativos al usuario (para ejecutar sudo):

```
nano /etc/sudoers
```

En la parte final colocamos: `usuario ALL=(ALL) ALL`

```
## Read drop-in files from /etc/sudoers.d
@includedir /etc/sudoers.d
erick ALL=(ALL) ALL

^G Help      ^O Write Out  ^W Where Is   ^K Cut
^X Exit      ^R Read File  ^_ Replace    ^U Pas
```

- Ctrl + X,
- escribimos “y” en caso de estar en inglés el sistema, o “s” si ya está en español.
- Enter.

## 7.4. Activamos los servicios de internet y bluetooth.

```
systemctl enable dhcpd NetworkManager
```

```
[root@archiso /]# systemctl enable dhcpd NetworkManager
Created symlink /etc/systemd/system/multi-user.target.wants/dhcpd.service → /usr/lib/systemd/system/dhcpd.service.
Created symlink /etc/systemd/system/multi-user.target.wants/NetworkManager.service → /usr/lib/systemd/system/NetworkManager.service.
Created symlink /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service → /usr/lib/systemd/system/NetworkManager-dispatcher.service.
Created symlink /etc/systemd/system/network-online.target.wants/NetworkManager-wait-online.service → /usr/lib/systemd/system/NetworkManager-wait-online.service.
```

```
systemctl enable bluetooth
```

```
[root@archiso /]# systemctl enable bluetooth
Created symlink /etc/systemd/system/dbus-org.bluez.service → /usr/lib/systemd/system/bluetooth.service.
Created symlink /etc/systemd/system/bluetooth.target.wants/bluetooth.service → /usr/lib/systemd/system/bluetooth.service.
```

## 7.5. Salimos:

```
[root@archiso /]# exit
exit
arch-chroot /mnt 11.75s user 25.11s system 0% cpu 1:02:51.88 total
root@archiso ~ #
```

## 7.6. Desmontamos las particiones.

```
root@archiso ~ # umount -R /mnt
root@archiso ~ #
```

## 7.7. Reiniciamos.

# 8. Conexión a internet (Postinstalación)

## 8.1. Verificamos que no contamos con una conexión a internet.

```
[root@Purito arch]# ping www.google.com
ping: www.google.com: Temporary failure in name resolution
[root@Purito arch]#
```

Esto se presenta tanto para usuarios con red inalámbrica o los que estén conectados directo a la red.

Por lo que, haremos 2 distinciones, 1 para los usuarios que estén conectados mediante Ethernet, es decir, de forma directa:

### 8.1.1. Red Cableada:

#### 8.1.1.1. Habilitar el servicio de “Network Manager”.

```
[root@Purito arch]# systemctl start NetworkManager.service
[root@Purito arch]# _
```

#### 8.1.1.2. Activar el servicio de “Network Manager”.

```
[root@Purito arch]# systemctl enable NetworkManager.service
Created symlink /etc/systemd/system/multi-user.target.wants/NetworkManager.service → /usr/lib/systemd/system/NetworkManager.service.
Created symlink /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service → /usr/lib/systemd/system/NetworkManager-dispatcher.service.
Created symlink /etc/systemd/system/network-online.target.wants/NetworkManager-wait-online.service → /usr/lib/systemd/system/NetworkManager-wait-online.service.
[root@Purito arch]# _
```

#### 8.1.1.3. Verificar la conexión a internet.



```
[root@Purito arch]# ping www.google.com
PING www.google.com ( [REDACTED] 56(84) bytes of data.
64 bytes from gru14s35-in-f4.1e100.net ([REDACTED]): icmp_seq=1 ttl=128 time=160 ms
64 bytes from gru14s35-in-f4.1e100.net ([REDACTED]): icmp_seq=2 ttl=128 time=158 ms
64 bytes from gru14s35-in-f4.1e100.net ([REDACTED]): icmp_seq=3 ttl=128 time=162 ms
^C
--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = [REDACTED] ms
```

## 8.1.2. Red inalámbrica:

8.1.2.1. Vemos las conexiones disponibles:

```
[root@Purito arch]# ip link
```

8.1.2.1.1. Otra alternativa también es posible:

```
nmcli dev wifi list
```

```
root@py nichel# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp2s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN mode DEFAULT group default qlen 1000
    link/ether [REDACTED] brd ff:ff:ff:ff:ff:ff
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DORMANT group default qlen 1000
    link/ether [REDACTED] brd ff:ff:ff:ff:ff:ff permaddr 04:0e:00:04:0e:00
root@py nichel# nmcli dev wifi list
IN-USE  BSSID          SSID      MODE  CHAN  RATE      SIGNAL  BARS  SECURITY
[REDACTED] [REDACTED] MIRIAM 2    Infra  1      195 Mbit/s  87      ****  WPA1 WPA2
[REDACTED] [REDACTED] MIRIAM          Infra  11     130 Mbit/s  50      **    WPA1 WPA2
[REDACTED] [REDACTED] ELENITA   Infra  6      270 Mbit/s  30      =     WPA1 WPA2
[REDACTED] [REDACTED] DARLIN MISO Infra  1      135 Mbit/s  17      =     WPA2
root@py nichel# nmcli dev wifi connect [REDACTED] password [REDACTED]
Device 'wlan0' successfully activated with '[REDACTED]'.
```

8.1.2.2. Levantamos la conexión que vayamos a conectarnos, en el caso se usa “wlan0”: como se presenta en el apartado “2.2”.

```
[root@Purito arch]# ip link set wlan0 up
```

8.1.2.3. Conexión a internet:

**Colocamos el nombre de la red entre comillas si tiene espacios, caso contrario no es necesario.**

```
[root@Purito arch]# nmcli dev wifi connect "Nombre_de_la_red" password contraseña
```

**Se escribe la contraseña tal cual sea sin comillas.**

```
nmcli dev wifi connect "NETLIFE-uiomvcostaa1" password
1102571369
```

8.1.2.4. Verificación de conexión:

```
[root@Purito arch]# ping www.google.com
PING www.google.com ( [REDACTED] 56(84) bytes of data.
64 bytes from gru14s35-in-f4.1e100.net ( [REDACTED] ): icmp_seq=1 ttl=128 time=160 ms
64 bytes from gru14s35-in-f4.1e100.net ( [REDACTED] ): icmp_seq=2 ttl=128 time=158 ms
64 bytes from gru14s35-in-f4.1e100.net ( [REDACTED] ): icmp_seq=3 ttl=128 time=162 ms
^C
--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = [REDACTED] ms
```

## 8.2. Mejorar los servidores de descarga.

Debido a que de aquí para adelante ya es la instalación del entorno o gestor para nuestro sistema, podemos ayudarnos mucho si los servidores de descarga son los más rápidos y por ende, el tiempo en que estos procesos duran, será más corto.

Por ello con ayuda de “Reflector”, vamos a optimizar las descargas:

<https://ayudalinux.com/optimiza-los-mirrors-de-arch-linux-con-reflector/>

<https://muylinux.xyz/obtener-la-ultima-lista-de-replicas-usando-reflector-en-arch-linux/>

### 8.2.1. Colocamos los comandos:

#### 8.2.1.1. Instalar reflector:

```
sudo pacman -S reflector rsync
```

```
[root@lagarto ~]# pacman -S reflector rsync
```

#### 8.2.1.2. Hacer un respaldo de la configuración por defecto de la mirrorlist:

```
sudo cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.bak
```

```
[root@lagarto ~]# cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.bak
```

#### 8.2.1.3. Colocar los 10 mirror más rápidos al principio:

```
sudo reflector --latest 20 --protocol https --sort rate --save /etc/pacman.d/mirrorlist
```

```
[root@lagarto ~]# reflector --latest 20 --protocol https --sort rate --save /etc/pacman.d/mirrorlist
```

#### 8.2.1.4. Actualizar la configuración y el sistema:

```
sudo pacman -Syuu --noconfirm
```

```
[root@lagarto ~]# pacman -Syuu --noconfirm
:: Sincronizando las bases de datos de los paquetes...
core               137,5 KiB   46,1 KiB/s   00:03 [#####] 100%
extra              1568,3 KiB  58,7 KiB/s   00:27 [#####] 100%
community          5,9 MiB    54,9 KiB/s   01:50 [#####] 100%
:: Iniciando actualización completa del sistema...
...el sistema ya está actualizado.
[root@lagarto ~]#
```

Otra opción que podemos usar en caso de que la anterior no nos convenció:

```
sudo reflector --verbose --country 'Country' -l 5 --sort  
rate --save /etc/pacman.d/mirrorlist
```

Donde “Country” es el país al que pertenecemos.

```
[root@lagarto ~]# reflector --verbose --country 'United States' -l 5 --sort rate --save /etc/pacman.d/mirrorlist
```

#### 8.2.1.5. Respaldo del mejor mirrorlist:

```
#####  
#####  
##### Arch Linux mirrorlist generated by Zeus  
#####  
#####  
#####  
#####
```

```
# With: reflector @/etc/xdg/reflectord/reflectord.conf  
# When: 2021-12-26 12:55:03 UTC  
# From: https://archlinux.org/mirrors/status/json/  
# Retrieved: 2021-12-26 12:53:14 UTC  
# Last Check: 2021-12-26 17:43:40 UTC
```

```
Server = https://mirror.osbeck.com/archlinux/$repo/os/$arch  
Server = https://arch.mirror.constant.com/$repo/os/$arch  
Server = https://mirror.luzea.de/archlinux/$repo/os/$arch  
Server = https://mirror.pseudoform.org/$repo/os/$arch  
Server = https://mirror.cspacehostings.com/archlinux/$repo/os/$arch  
Server = https://archlinux.thaller.ws/$repo/os/$arch  
Server = https://mirror.telepoint.bg/archlinux/$repo/os/$arch  
Server = https://mirror.f4st.host/archlinux/$repo/os/$arch  
Server = https://phinau.de/arch/$repo/os/$arch  
Server = https://archmirror.it/repos/$repo/os/$arch  
Server = https://mirror.cyberbits.asia/archlinux/$repo/os/$arch  
Server = https://archlinux-br.com.br/archlinux/$repo/os/$arch  
Server = https://ftp.halifax.rwth-aachen.de/archlinux/$repo/os/$arch  
Server = https://mirrors.neusoft.edu.cn/archlinux/$repo/os/$arch  
Server = https://mirror.pkgbuild.com/$repo/os/$arch  
Server = https://archlinux.uk.mirror.allworldit.com/archlinux/$repo/os/$arch  
Server = https://archlinux.za.mirror.allworldit.com/archlinux/$repo/os/$arch  
Server = https://mirror.theash.xyz/arch/$repo/os/$arch  
Server = https://archlinux.mailtunnel.eu/$repo/os/$arch  
Server = https://mirror.lty.me/archlinux/$repo/os/$arch
```

## 9. Drivers (pantalla, audio y video)

**9.1 Listamos el tipo de tarjeta gráfica con la que cuenta nuestro pc:**

```
lspci | grep VGA
```

En el caso del ejemplo se obtiene resultado como:

**00:02.0 VGA compatible controller: Intel Corporation HD Graphics 620 (rev 02)**

9.1.1. Driver de gráficos para AMD:

```
pacman -S xf86-video-nouveau xf86-video-ati xf86-video-amdgpu vulkan-radeon amd-ucode
```

9.1.2. Driver de gráficos para Intel:

```
pacman -S xf86-video-intel xf86-video-vesa xf86-video-nouveau vulkan-intel intel-ucode
```

### 9.2. Drivers de audio y video:

```
pacman -Sy pulseaudio pavucontrol gstreamer gst-plugins-bad gst-plugins-base gst-plugins-base-libs gst-plugins-good gst-plugins-ugly xine-lib libdvdcss libdvdread dvd+rw-tools vlc lame alsa-utils alsa-plugins alsa-oss ffmpeg gstreamer --needed --noconfirm
```

### 9.3. Drivers para laptops:

9.3.1. Touchpad:

```
sudo pacman -S xf86-input-synaptics xf86-input-libinput
```

9.3.2. Control por teclas de brillo:

```
sudo pacman -S brightnessctl
```

9.3.3. Control por teclas de sonido:

```
sudo pacman -S xdg-user-dirs pulseaudio pulseaudio-alsa xbindkeys --needed --noconfirm
```

## 10. Xorg

### 10.1. Instalar complementos de xorg.

Xorg es una serie de herramientas que son indispensables para un correcto funcionamiento tanto para gestores de ventanas como para entornos de escritorio, a continuación se muestra en una línea lo mencionado:

```
pacman -Sy xorg xorg-server mesa mesa-demos xorg-apps xorg-xinit xorg-twm xterm xorg-xclock --noconfirm
```

# 11. *Fuentes de letra*

```
pacman -S ttf-dejavu ttf-liberation noto-fonts gnu-free-fonts  
ttf-hack ttf-inconsolata gnome-font-viewer --needed
```