

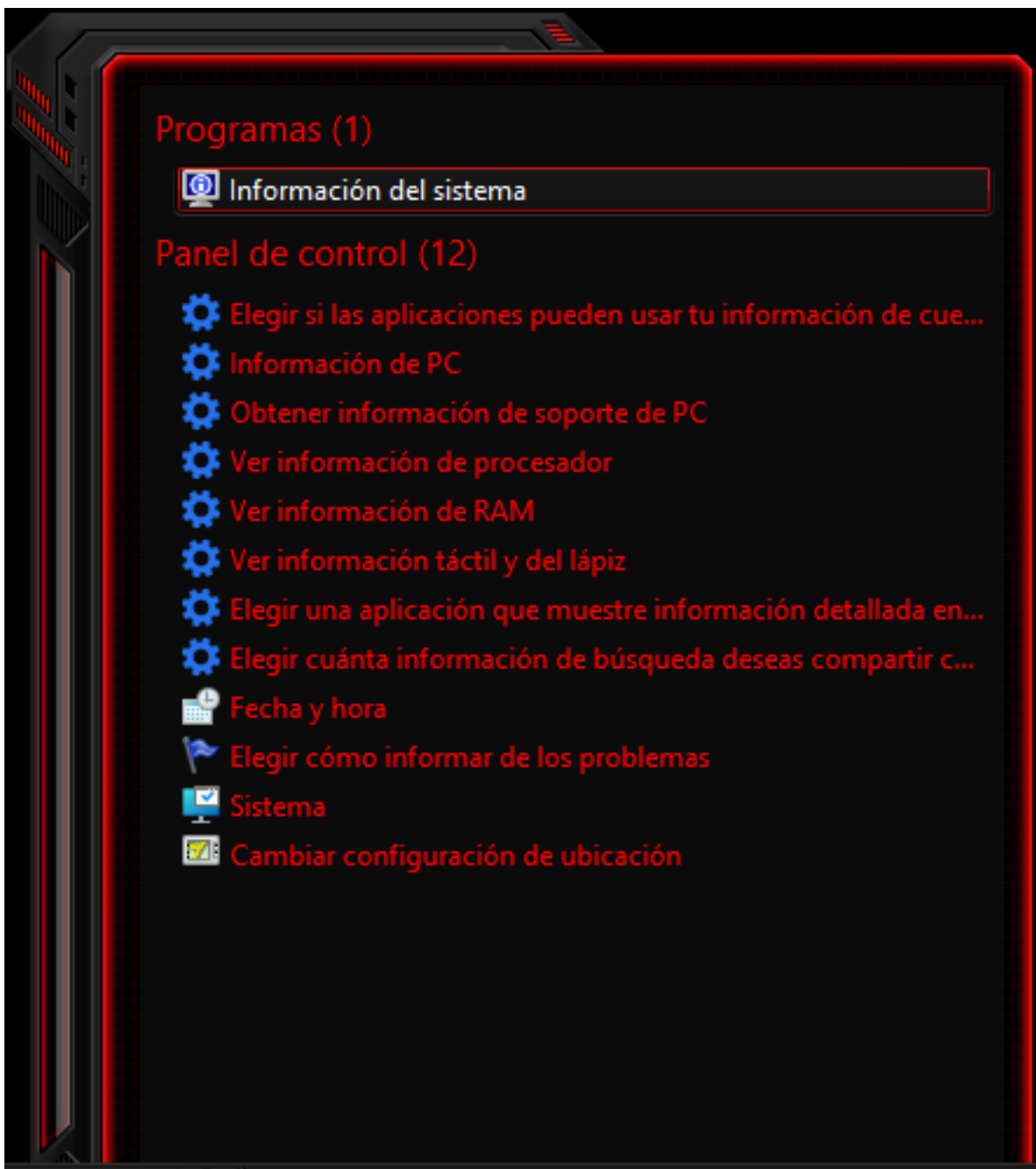
INSTALACIÓN (PURA) (BIOS/UEFI)

1. Preinstalación (Datos del sistema)

Antes que nada para realizar una correcta instalación del sistema, es necesario conocer cuáles son las características específicas con las que cuenta, debido a que, sino se toma en cuenta al momento de realizar la instalación, se presentarán errores al hacer las particiones del sistema y también al reiniciar el equipo, éste no ingresará al sistema.

1.1. Averiguar si nuestra placa base utiliza BIOS o UEFI

1.1.1. Lo podemos averiguar de una manera muy sencilla, al usar el buscador de windows y escribir: "Información del sistema":



Y nos aparecerá un resultado similar:

Elemento	Valor
Nombre del SO	Microsoft Windows 8.1 Pro
Versión	
Descripción adicional del SO	No disponible
Fabricante del SO	Microsoft Corporation
Nombre del sistema	ERICK
Fabricante del sistema	Dell Inc.
Modelo del sistema	Inspiron 14-3467
Tipo de sistema	PC basado en x64
SKU del sistema	
Procesador	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 2601 Mhz, 2 procesadores princi...
Versión y fecha de BIOS	Dell Inc. 2.9.0, 17/01/2019
Versión de SMBIOS	3.1
Versión de controladora integr...	255.255
Modo de BIOS	UEFI
Fabricante de la placa base	Dell Inc.
Modelo de placa base	No disponible
Nombre de la placa base	Placa base
Rol de plataforma	Móvil
Estado de arranque seguro	Desactivada
Configuración de PCR7	Enlace no posible
Directorio de Windows	C:\Windows
Directorio del sistema	C:\Windows\system32
Dispositivo de arranque	
Configuración regional	

Por lo que, concluimos que el tipo de placa base que se usa es “UEFI”.

1.2. Ahora debemos averiguar. ¿Qué tipo de de partición tengo en mi sistema?

1.2.1. Para averiguarlo, abrimos cmd como administrador y colocamos: “diskpart”:

```

Administrador: C:\Windows\System32\cmd.exe - diskpart
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>diskpart

Microsoft DiskPart versión 6.3.9600

Copyright (C) 1999-2013 Microsoft Corporation.
En el equipo: ERICK

DISKPART>
  
```

1.2.2. Luego el comando “list disk”:

```
Administrador: C:\Windows\System32\cmd.exe - dis...

DISKPART> list disk

  Núm Disco  Estado      Tamaño  Disp  Din  Gpt
-----
Disco 0      En línea      931 GB  0 B   *

DISKPART>
```

Si nos fijamos, en la parte de GPT hay un “*”, el que simboliza que el tipo de particionado de disco que cuenta el sistema es “GPT”, en caso de no encontrarse ese “*”, el tipo de particionado de disco será “MBR”.

1.3. Datos esenciales del computador:

1.3.1. Memoria RAM:

Edición de Windows
Windows 8.1 Pro
© 2013 Microsoft Corporation. Todos los derechos reservados.
Obtener más características con una nueva edición de Windows

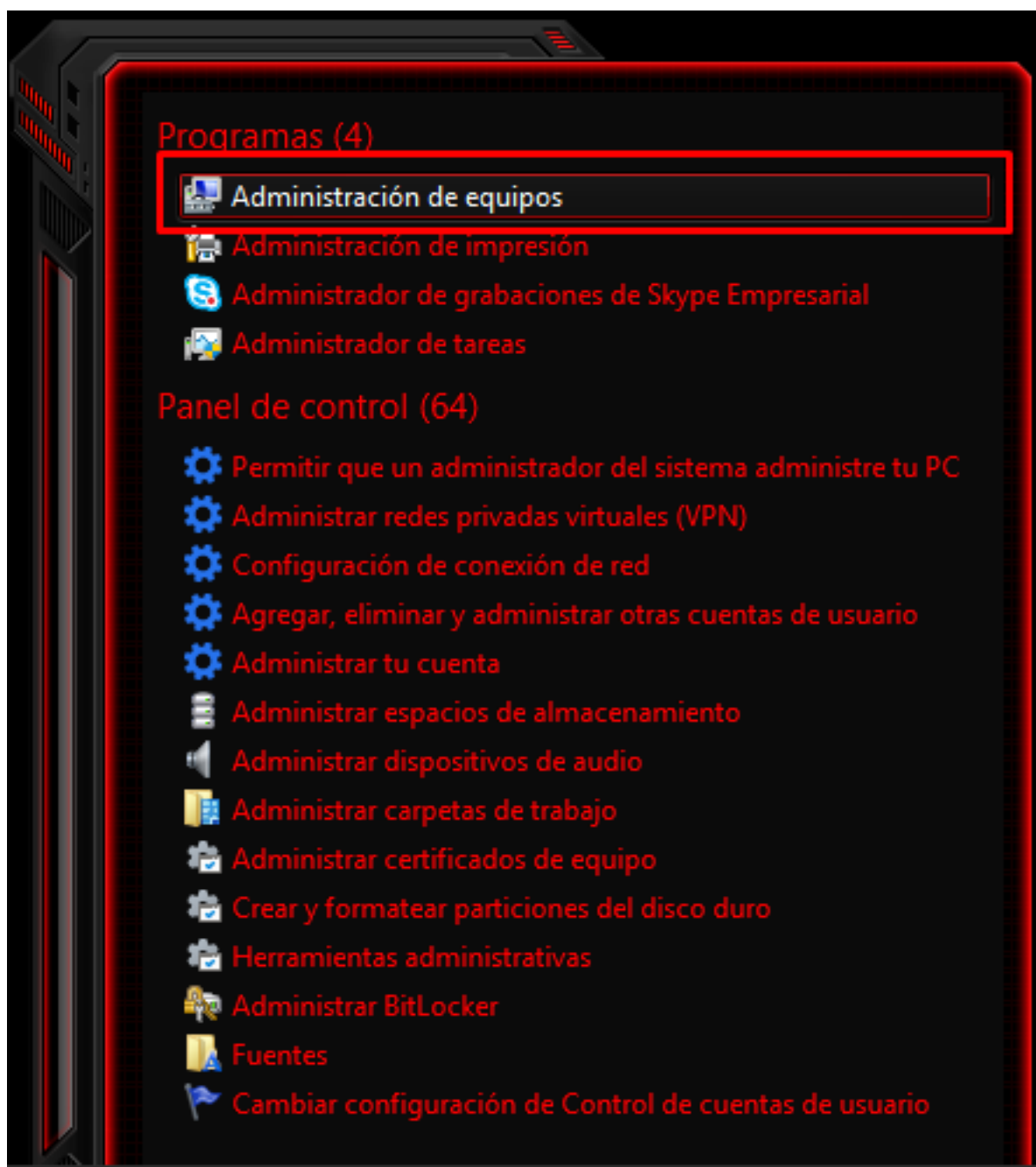
**Windows® 8**

Sistema
Fabricante: Tester Machine
Modelo: Fast OS 8 V2 + AST v3.0
Procesador: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.60 GHz
Memoria instalada (RAM): **8,00 GB (7,87 GB utilizable)**
Tipo de sistema: Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil: La entrada táctil o manuscrita no está disponible para esta pantalla



1.3.2. Almacenamiento y particiones:

Para ubicarlo ingresamos a:



Luego:

Administración de equipos

Archivo Acción Ver Ayuda

Administración del equipo (local)

Herramientas del sistema

Programador de tareas

Visor de eventos

Carpetas compartidas

Usuarios y grupos locales

Rendimiento

Administrador de dispositivos

Almacenamiento

Administración de discos

Servicios y Aplicaciones

Volumen	Distribución	Tipo	Sistema de archivos	Estado
	Simple	Básico		Correcto (Partición primaria)
	Simple	Básico		Correcto (Partición primaria)
	Simple	Básico		Correcto (Partición primaria)
	Simple	Básico		Correcto (Partición de recuperación)
	Simple	Básico		Correcto (Partición de sistema EFI)
(C:)	Simple	Básico	NTFS	Correcto (Arranque, Archivo de paginación, Partición de recuperación)
Otro Respaldo (E:)	Simple	Básico	NTFS	Correcto (Partición primaria)
RESPALDO (D:)	Simple	Básico	NTFS	Correcto (Partición primaria)

Disco 0

Básico

931,39 GB

En pantalla

301 I	4,00 GB	427,20 GB	RESPALDO (Otro Respal	300 I	100	(C:)	
Corr	Correcto	Correcto (Parti	200,00 GB NT	147,61 GB NT	Corr	Co	151,90 GB NT	
		Correcto (Par	Correcto (Par	Correcto (Pa			Correcto (Arr	

CD-ROM 0

DVD (F:)

No hay medios

No asignado

Partición primaria

Y podemos observar todas las particiones que existen en nuestro ordenador.

Si queremos dejar alguna partición lista para trabajar, solo eliminamos el espacio, este porceso también se puede hacer desde la instalación de Arch.

2. Teclado e internet (Instalación)

6/29

```
Arch Linux 5.15.5-arch1-1 (tty1)
```

```
archiso login: root (automatic login)
```

```
To install Arch Linux follow the installation guide:  
https://wiki.archlinux.org/title/Installation\_guide
```

```
For Wi-Fi, authenticate to the wireless network using the iwctl utility.  
For mobile broadband (WWAN) modems, connect with the mmcli utility.  
Ethernet, WLAN and WWAN interfaces using DHCP should work automatically.
```

```
After connecting to the internet, the installation guide can be accessed  
via the convenience script Installation_guide.
```

```
root@archiso ~ # _
```

Ahora comenzamos con la instalación propiamente dicha:

2.1. Elegimos la distribución de teclado que vamos a ocupar para el uso de toda la instalación y el propio sistema, siendo así:

```
After connecting to the internet, the installation guide can be accessed  
via the convenience script Installation_guide.
```

```
root@archiso ~ # loadkeys es  
root@archiso ~ # _
```

Se debe usar el teclado con el que el usuario esté familiarizado, en el ejemplo se presenta con la distribución “es” que quiere decir: “Español”, por otra parte, se puede usar otra distribución más familiarizada como “latam” que se refiere al teclado “latinoamericano”.

Sin embargo, si usted usa otra distribución de teclado específica, lo puede encontrar con el comando:

```
root@archiso ~ # localectl list-keymaps
```

Y le aparecerá una lista completa de todas las distribuciones de teclado existentes y para filtrar una búsqueda específica, usamos:

```
root@archiso ~ # localectl list-keymaps | grep -i teclado_que_necesitamos
```

2.2. Identificar la red de internet. [En caso de estar conectado directamente por cable (Ethernet), se puede saltar este paso]

Para conseguirlo, ingresamos el comando: “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]

```

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.

2.3. Conexión directa a internet.

Para conectarnos directamente a internet, vamos a realizarlo con una sola línea de comando:

```

root@archiso ~ # iwctl --passphrase [redacted] station [redacted] connect [redacted]

```

Colocamos la contraseña tal cual sea, sin comillas.

El tipo de red que decidimos conectarnos

Colocamos el nombre de la red, ENTRE COMILLAS

Cabe mencionar que si la red tienes espacios, éstos se respetan, así como las mayúsculas es decir: "Juanito 2"; "Red 1"; "Pepe"

2.4 Verificación de acceso a internet:

Para verificar el acceso a internet, basta con escribir el comando “ping” seguido de una dirección.

Ejemplo:

```

root@archiso ~ # ping www.google.com
PING www.google.com (142.250.217.228) 56(84) bytes of data.
64 bytes from mia07s62-in-f4.1e100.net (142.250.217.228): icmp_seq=1 ttl=128 time=62.7 ms
64 bytes from mia07s62-in-f4.1e100.net (142.250.217.228): icmp_seq=2 ttl=128 time=62.4 ms
64 bytes from mia07s62-in-f4.1e100.net (142.250.217.228): icmp_seq=3 ttl=128 time=62.1 ms
64 bytes from mia07s62-in-f4.1e100.net (142.250.217.228): icmp_seq=4 ttl=128 time=62.0 ms
64 bytes from mia07s62-in-f4.1e100.net (142.250.217.228): icmp_seq=5 ttl=128 time=62.3 ms
^C
--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = [redacted] ms

```

3. Verificación de imagen

Para continuar con la instalación, es necesario que sepamos que la imagen de

archlinux no ha sido modificada por terceros ya que si ese es el caso la instalación es riesgosa y nuestros datos personales se pueden ver vulnerables.

3.1. Verificamos que la imagen .iso que descargamos no ha sido manipulada por terceros y es original:

```
gpg --keyserver-options auto-key-retrieve --verify archlinux-version-x86_64.iso.sig
```

version: día en que descargamos la .iso

Ejemplo: archlinux-2021.12.01-x86_64.iso.

```
root@archiso ~ # gpg --keyserver-options auto-key-retrieve --verify archlinux-2021.12.01-x86_64.iso.sig
gpg: directory '/root/.gnupg' created
gpg: keybox '/root/.gnupg/pubring.kbx' created
gpg: can't open 'archlinux-2021.12.01-x86_64.iso.sig': No such file or directory
gpg: verify signatures failed: No such file or directory
```

Otra alternativa también es:

```
pacman-key -v archlinux-version-x86_64.iso.sig
```

version: día en que descargamos la .iso

Ejemplo: archlinux-2021.12.01-x86_64.iso.

```
root@archiso ~ # pacman-key -v archlinux-2021.12.01-x86_64.iso.sig
==> Checking archlinux-2021.12.01-x86_64.iso.sig... (embedded)
grep: archlinux-2021.12.01-x86_64.iso.sig: No such file or directory
gpg: can't open 'archlinux-2021.12.01-x86_64.iso.sig': No such file or directory
gpg: verify signatures failed: No such file or directory
==> ERROR: The signature identified by archlinux-2021.12.01-x86_64.iso.sig could not be verified.
```

4. Particiones de disco (UEFI y BIOS)

En el caso de que hallamos llegado hasta este punto y no sepamos qué tipo de placa base cuenta nuestra pc, podemos averiguarlo con el comando:

```
ls /sys/firmware/efi/efivars
```

Ejecutado el comando, si nos aparece una lista de alrededor de 10 líneas, significa que nuestro pc tiene una placa base en UEFI:

```

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
Lang-8be4df61-93ca-11d2-aa0d-00e098032b8c
LoaderEntries-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f
LoaderEntrySelected-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f
LoaderFeatures-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f
LoaderFirmwareInfo-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f
LoaderFirmwareType-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f
LoaderImageIdentifier-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f
LoaderInfo-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f
LoaderTimeExecUSec-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f
LoaderTimeInitUSec-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f
LoaderTimeMenuUSec-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f
MTC-eb704011-1402-11d3-8e77-00a0c969723b
MemoryOverwriteRequestControlLock-bb983ccf-151d-40e1-a07b-4a17be168292
MemoryOverwriteRequestControl-e20939be-32d4-41be-a150-897f85d49829
MemoryTypeInfoInformation-4c19049f-4137-4dd3-9c10-8b97a83ffdfa
OsIndicationsSupported-8be4df61-93ca-11d2-aa0d-00e098032b8c
PKDefault-8be4df61-93ca-11d2-aa0d-00e098032b8c
PK-8be4df61-93ca-11d2-aa0d-00e098032b8c
PlatformLangCodes-8be4df61-93ca-11d2-aa0d-00e098032b8c

```

caso contrario:

```

root@archiso ~ # ls /sys/firmware/efi/efivars
ls: cannot access '/sys/firmware/efi/efivars': No such file or directory
2 root@archiso ~ # _

```

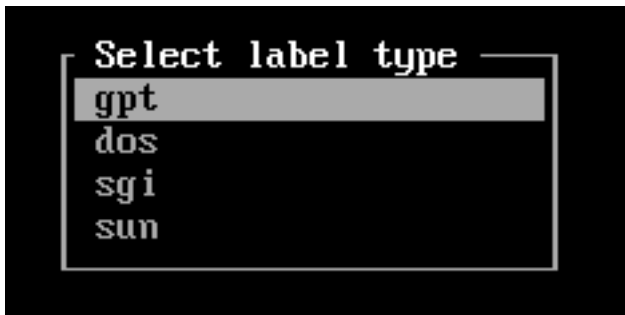
Nos saldrá ese error y la placa base será del tipo BIOS.

Por ello, ésta parte se dividirá en 2 partes, una para el particionado del tipo UEFI y otra para el tipo BIOS:

4.1. Particionado del tipo UEFI

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

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



4.1.2. Seleccionamos “gpt” para el tipo del particionado.

```

                                Disk: /dev/sda
                        Size: 30 GiB, 32212254720 bytes, 62914560 sectors
                    Label: gpt, identifier: C41059F5-16F7-3648-B44D-24D7134BE48A

```

Device	Start	End	Sectors	Size	Type
>> Free space	2048	62914526	62912479	30G	

Para realizar un correcto partionado, 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 partiones 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 particones se encuentra ubicado, esto con la finalidad de evitar dañar los otros sistemas.

```

[ New ] [ Quit ] [ Help ] [ Write ] [ Dump ]

```

Create new partition from free space

Figura 1: Tabla de particiones e información.

4.1.3. Tabla de particiones:

4.1.3.1 Creamos una partición de: 300M (dónde M significa Megabytes)

Disk: /dev/sda					
Size: 20 GiB, 21474836480 bytes, 41943040 sectors					
Label: gpt, identifier: C510805A-F6B2-7641-AC06-14A776FF0553					
Device	Start	End	Sectors	Size	Type
>> /dev/sda1	2048	616447	614400	300M	Linux filesystem
Free space	616448	41943006	41326559	19.7G	

4.1.3.2. Cambiamos el tipo a EFI System:

4.1.3.2.1. Seleccionamos "Type" marcado la partición creada:

```
Partition UUID: 016FFE88-65EA-E24C-B8F4-029328BFABB8  
Partition type: Linux filesystem (0FC63DAF-8483-4772-8E79-3D69D8477DE4)
```

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

4.1.3.2.2. Seleccionamos EFI System:

```
Select partition type  
EFI System  
MBR partition scheme  
Intel Fast Flash  
BIOS boot  
Sony boot partition  
Lenovo boot partition  
PowerPC PReP boot  
ONIE boot  
ONIE config  
Microsoft reserved
```

4.1.3.2.3. Verificamos que se seleccionó:

Disk: /dev/sda
Size: 20 GiB, 21474836480 bytes, 41943040 sectors
Label: gpt, identifier: C510805A-F6B2-7641-AC06-14A776FF0553

Device	Start	End	Sectors	Size	Type
>> /dev/sda1	2048	616447	614400	300M	EFI System
Free space	616448	41943006	41326559	19.7G	

4.1.3.2.4. Escribimos para guardar el cambio con "Write":

```
Partition UUID: 016FFE88-65EA-E24C-B8F4-029328BFABB8  
Partition type: EFI System (C12A7328-F81F-11D2-BA4B-00A0C93EC93B)
```

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

Write partition table to disk (this might destroy data)

4.1.3.2.5. Damos en "yes" y salimos con "Enter":

```
Partition UUID: 016FFE88-65EA-E24C-B8F4-029328BFABB8  
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.

4.1.3.2.6. Vemos el cuadro de diálogo:

```
Partition UUID: 016FFE88-65EA-E24C-B8F4-029328BFABB8  
Partition type: EFI System (C12A7328-F81F-11D2-BA4B-00A0C93EC93B)
```

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

The partition table has been altered.

4.1.3.3. Creamos la partición raíz: Donde "G" significa "Gigabyte".

Partition size: 15.7G

May be followed by M for MiB, G for GiB, T for TiB, or S for sectors.

4.1.3.4. No cambiamos el tipo (Type) y solo escribimos (Write):

Disk: /dev/sda
Size: 20 GiB, 21474836480 bytes, 41943040 sectors
Label: gpt, identifier: C510805A-F6B2-7641-AC06-14A776FF0553

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	616447	614400	300M	EFI System
>> /dev/sda2	616448	34170879	33554432	16G	Linux filesystem
Free space	34170880	41943006	7772127	3.7G	

4.1.3.5. Creamos la partición de intercambio: Swap.

Partition size: 3.7G

May be followed by M for MiB, G for GiB, T for TiB, or S for sectors.

4.1.3.6. Cambiamos el tipo (Type) y ponemos: Linux Swap:



Si el usuario cuenta con muchas particiones y tiene miedo de perderlas u olvidar en cuál instalar cada partición, se recomienda tomar nota de cada partición en una hoja aparte.

Disk: /dev/sda
Size: 20 GiB, 21474836480 bytes, 41943040 sectors
Label: gpt, identifier: C510805A-F6B2-7641-AC06-14A776FF0553

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	616447	614400	300M	EFI System
/dev/sda2	616448	34170879	33554432	16G	Linux filesystem
>> /dev/sda3	34170880	41943006	7772127	3.7G	Linux swap

4.1.3.7. Escribimos y salimos (Quit):

Partition UUID: E088BC62-E6E0-A746-B8EA-44389381B759
Partition type: Linux swap (0657FD6D-A4AB-43C4-84E5-0933C84B4F4F)

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

Quit program without writing changes

4.1.4. Formateo de las particiones:

4.1.4.1. Formateo del arranque:

```
root@archiso ~ # mkfs.vfat -F32 /dev/sda1  
mkfs.fat 4.2 (2021-01-31)
```

4.1.4.2. Formateo de la raíz:

```
root@archiso ~ # mkfs.ext4 /dev/sda2
mke2fs 1.46.4 (18-Aug-2021)
Creating filesystem with 4194304 4k blocks and 1048576 inodes
Filesystem UUID: 85a08294-7c36-4232-8651-914a24184933
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
```

4.1.4.3. Formateo del intercambio:

```
root@archiso ~ # mkswap /dev/sda3
Setting up swspace version 1, size = 3.7 GiB (3979321344 bytes)
no label, UUID=6416137e-b150-4fe2-a98d-0a8c448f8b07
```

4.1.4.4. Formato a la particion de intercambio (swap):

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

4.1.5. Montaje de las particiones:

4.1.5.1 Montaje de la raíz:

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

4.1.5.2. Creamos el punto de montaje para el boot:

```
root@archiso ~ # mkdir /mnt/boot
```

4.1.5.3. Creamos el punto de montaje para EFI:

```
root@archiso ~ # mkdir /mnt/boot/efi
```

4.1.5.4. Montaje del boot:

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

4.2. Particionado del tipo BIOS

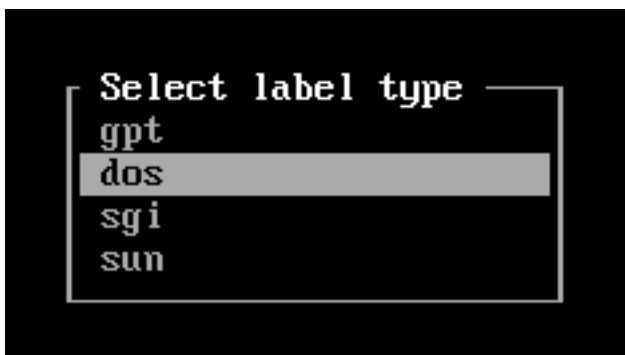
4.2.1. Listamos los discos/particiones existentes:

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

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

```
root@archiso ~ # cfdisk _
```

4.2.3. Seleccionamos “dos” para el tipo del particionado.

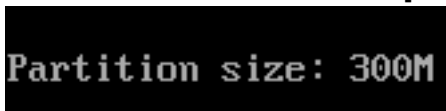


4.2.4. Vemos la tabla de particiones:

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.

4.2.5. Creamos una partición de 300M para el "Boot": Del tipo "primaria".



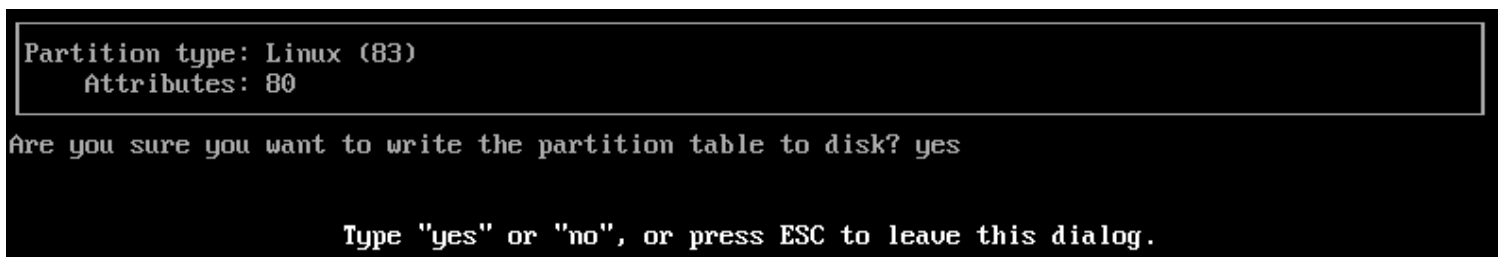
4.2.6. Elegimos el tipo de la partición: "Bootable".



4.2.7. Seleccionando la partición, la "escribimos":



4.2.8. Ponemos "yes" para efectuar el cambio:



4.2.9. 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 ]
```

4.2.10. 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 ]
```

4.2.11. 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
```

4.2.12. 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

4.2.13. Salimos con “Quit”:

```
Partition type: Linux swap / Solaris (82)
```

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

4.2.14. Formateo de las particiones.

4.2.14.1. 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
```

4.2.14.2. 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
```

4.2.14.3. 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
```

4.2.14.4. Formato a la particion de intercambio (swap):

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

4.2.15. Montaje de las particiones:

4.2.15.1 Montaje de la raíz:

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

4.2.15.2. Creamos el punto de montaje para el boot:

```
root@archiso ~ # mkdir /mnt/boot
```

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

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

5. Instalación de paquetes básicos.

Después de crear las particiones del sistema, podemos comenzar a instalar los paquetes básicos para que pueda funcionar el sistema tanto para UEFI y BIOS en adelante, esto lo podemos lograr con el siguiente comando:

5.1. Comando de instalación de paquetes base.

```
root@archiso ~ # pacstrap /mnt base base-devel linux linux-firmware nano vim networkmanager network-manager-applet wireless_tools  
s wpa_supplicant os-prober mtools dosfstools linux-headers grub dhcpcd efibootmgr netctl dialog gvfs xdg-user-dirs xf86-input-sy  
naptics
```

Este proceso depende exclusivamente de la conexión a internet, puede variar de 5 a 10 min.

5.2. Generar un fichero de particiones (fstab).

Este fichero nos sirve para guardar las particiones que hemos realizado anteriormente, este proceso es el mismo ya sea para BIOS o UEFI.

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

Podemos verificar que el archivos se generó correctamente con:

```
root@archiso ~ # cat /mnt/etc/fstab  
# Static information about the filesystems.  
# See fstab(5) for details.  
  
# <file system> <dir> <type> <options> <dump> <pass>  
# UUID=fb41600a-e7bf-4624-b900-179637030591  
/dev/sda2 / ext4 rw,relatime 0 1  
  
# UUID=1ad0c4ea-4ce2-43de-8aaf-302fac886c58  
/dev/sda1 /boot ext2 rw,relatime 0 2
```

6. Configuración básica

6.1. 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 /]#
```

6.2. Zona horaria:

In -sf /usr/share/zoneinfo/*Región/Ciudad* /etc/localtime

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

Un ejemplo sería:

In -sf /usr/share/zoneinfo/America/Argentina/Buenos_Aires /etc/localtime

O también:

In -sf /usr/share/zoneinfo/America/Chicago /etc/localtime

6.3. 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”.

6.4. 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.
```

6.5. Distribución de teclado:

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

6.6. 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]: [REDACTED] 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.nwtime.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
```

7. GRUB

Instalamos el grub, sin embargo este proceso varía dependiendo si es BIOS o UEFI.

7.1. Grub en UEFI

```
[root@archiso /]# grub-install --efi-directory=/boot/efi --bootloader-id='Pro Arch' --target=x86_64-efi
Installing for x86_64-efi platform.
Installation finished. No error reported.
[root@archiso /]#
```

grub-install --target=x86_64-efi --efi-directory=/boot/efi/ --bootloader-id=grub --boot-directory=/boot/efi/EFI --recheck --debug

En el caso de que tengamos algún otro arranque de un viejo sistema u otra distribución de linux obsoleta y querramos eliminarla de paso, agregamos al final: --removable.

```
[root@archiso /]# grub-install --efi-directory=/boot/efi --bootloader-id='Pro Arch' --target=x86_64-efi --removable
```

Se recomienda usar la primera opción en caso de tener windows y mantener un “dual-boot”.

7.2. Grub en BIOS:

```
[root@archiso /]# grub-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
[root@archiso /]# _
```

7.3. Configuración del GRUB: BIOS

```
[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
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
[root@archiso /]# _
```

7.4. Configuración del GRUB: UEFI

```
[root@archiso /]# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/umluz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot: initramfs-linux-fallback.img
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
[root@archiso /]#
```

8. Usuarios

No debemos reiniciar en ningún momento del proceso.

8.1 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 /]#
```

8.2. Usuario normal: (Luego se asignará los permisos de administrador “sudo”)

```
[root@archiso /]# groupadd pepe
[root@archiso /]# useradd -m -g pepe -s /bin/bash pepe
[root@archiso /]# passwd pepe
New password:
Retype new password:
passwd: password updated successfully
[root@archiso /]#
```

Y Listo!!, hasta este punto ya tenemos instalado el sistema.

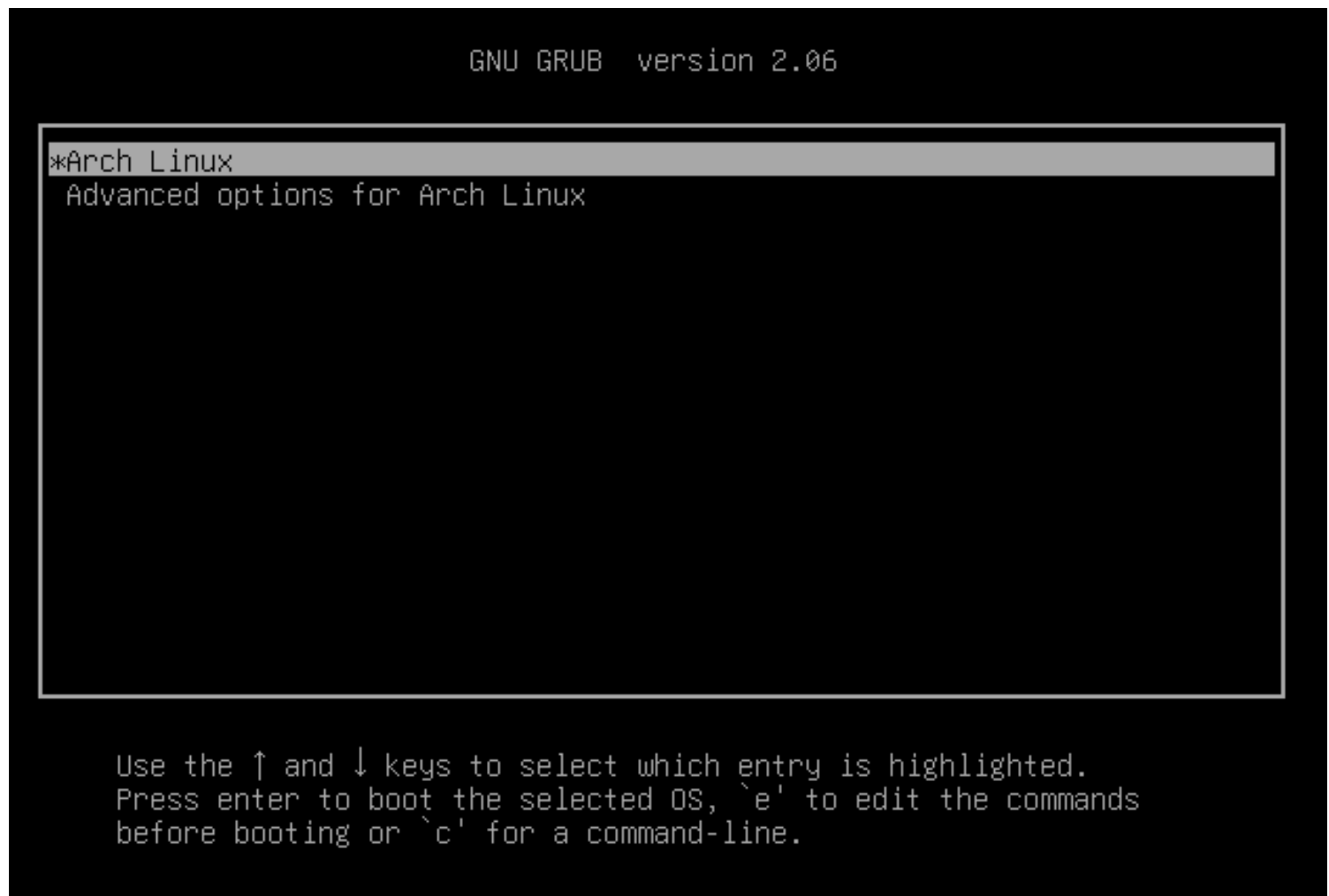
8.3. Salimos:

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

8.4. Desmontamos las particiones.

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

8.5. Reiniciamos.



En el momento que aparezca el grub sin errores y nos indica la distribución, quiere decir que hemos logrado una instalación correcta.

8.6. Dar administrativos al usuario creado.

8.6.1. Iniciamos sesión como usuario “root”:

```
lagarto login: root
Password:
[root@lagarto ~]# _
```

8.6.2. Editamos la configuración de “sudoers”, con el comando:
nano /etc/sudoers

```

GNU nano 6.0 /etc/sudoers
## sudoers file.
##
## This file MUST be edited with the 'visudo' command as root.
## Failure to use 'visudo' may result in syntax or file permission errors
## that prevent sudo from running.
##
## See the sudoers man page for the details on how to write a sudoers file.
##
##
## Host alias specification
##
## Groups of machines. These may include host names (optionally with wildcards),
## IP addresses, network numbers or netgroups.
# Host_Alias    WEBSERVERS = www1, www2, www3

```

8.6.3. Nos ubicamos al final del archivo y colocamos: (sin las comillas)

<https://askubuntu.com/questions/334318/sudoers-file-enable-nopasswd-for-user-all-commands>

"usuario" ALL=(ALL) NOPASSWD:ALL

```

## Uncomment to allow any user to run sudo if they know the password
## of the user they are running the command as (root by default).
# Defaults targetpw # Ask for the password of the target user
# ALL ALL=(ALL) ALL # WARNING: only use this together with 'Defaults targetpw'

## Read drop-in files from /etc/sudoers.d
@includedir /etc/sudoers.d
pepe ALL=(ALL) NOPASSWD:ALL

```

→ Colocamos nuestro usuario.

Damos "Ctrl + X"; "y"; "Enter" para guardar los cambios.

La configuración anterior (8.6.3) es para que cuando ejecutemos "sudo" no nos pida la contraseña.

8.6.4. En el caso de que cada vez que escribamos "sudo" SI nos pida la contraseña, es decir ejecutemos algún comando como administrador, en la última línea en vez de "NOPASSWD:ALL" colocamos "ALL":

```

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

```

9. Internet (Postinstalación)

9.1. 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:

9.1.1. Red Cableada:

9.1.1.1. Habilitar el servicio de “Network Manager”.

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

9.1.1.2. 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]# _
```

9.1.1.3. 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
```

9.1.2. Red inalámbrica:

9.1.2.1. Vemos las conexiones disponibles:

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

9.1.2.2. 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
```

9.1.2.3. 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
```

9.1.2.4. Verificación de conexión:

```
[root@Purito arch]# ping www.google.com
PING www.google.com ( ) 56(84) bytes of data.
64 bytes from gru14s35-in-f4.1e100.net ( ): icmp_seq=1 ttl=128 time=160 ms
64 bytes from gru14s35-in-f4.1e100.net ( ): icmp_seq=2 ttl=128 time=158 ms
64 bytes from gru14s35-in-f4.1e100.net ( ): 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 =  ms
```

9.2. 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/>

9.2.1. Colocamos los comandos:

9.2.1.1. Instalar reflector:

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

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

9.2.1.2. 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
```

9.2.1.3. 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
```

9.2.1.4. 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
```

9.2.1.5. Respaldo Mirrorlist:

```
#####
#####
##### Arch Linux mirrorlist generated by Reflector
#####
#####
#####
```

With: reflector @/etc/xdg/reflector/reflector.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://mirrors.kamey.tk/archlinux/\\$repo/os/\\$arch](https://mirrors.kamey.tk/archlinux/$repo/os/$arch)

Server = [https://mirrors.neusoft.edu.cn/archlinux/\\$repo/os/\\$arch](https://mirrors.neusoft.edu.cn/archlinux/$repo/os/$arch)

Server = [https://mirror.pseudoform.org/\\$repo/os/\\$arch](https://mirror.pseudoform.org/$repo/os/$arch)

Server = [https://archlinux.uk.mirror.allworldit.com/archlinux/\\$repo/os/\\$arch](https://archlinux.uk.mirror.allworldit.com/archlinux/$repo/os/$arch)

Server = [https://mirror.telepoint.bg/archlinux/\\$repo/os/\\$arch](https://mirror.telepoint.bg/archlinux/$repo/os/$arch)

Server = [https://mirror.cyberbits.eu/archlinux/\\$repo/os/\\$arch](https://mirror.cyberbits.eu/archlinux/$repo/os/$arch)

Server = [https://phinau.de/arch/\\$repo/os/\\$arch](https://phinau.de/arch/$repo/os/$arch)

Server = [https://archmirror.it/repos/\\$repo/os/\\$arch](https://archmirror.it/repos/$repo/os/$arch)

Server = [https://mirror.cyberbits.asia/archlinux/\\$repo/os/\\$arch](https://mirror.cyberbits.asia/archlinux/$repo/os/$arch)

Server = [https://archlinux.thaller.ws/\\$repo/os/\\$arch](https://archlinux.thaller.ws/$repo/os/$arch)

Server = [https://mirror.pkgbuild.com/\\$repo/os/\\$arch](https://mirror.pkgbuild.com/$repo/os/$arch)

Server = [https://archlinux.za.mirror.allworldit.com/archlinux/\\$repo/os/\\$arch](https://archlinux.za.mirror.allworldit.com/archlinux/$repo/os/$arch)

Server = [https://europe.mirror.pkgbuild.com/\\$repo/os/\\$arch](https://europe.mirror.pkgbuild.com/$repo/os/$arch)

Server = [https://america.mirror.pkgbuild.com/\\$repo/os/\\$arch](https://america.mirror.pkgbuild.com/$repo/os/$arch)
Server = [https://asia.mirror.pkgbuild.com/\\$repo/os/\\$arch](https://asia.mirror.pkgbuild.com/$repo/os/$arch)
Server = [https://archlinux.mailtunnel.eu/\\$repo/os/\\$arch](https://archlinux.mailtunnel.eu/$repo/os/$arch)
Server = [https://mirror.f4st.host/archlinux/\\$repo/os/\\$arch](https://mirror.f4st.host/archlinux/$repo/os/$arch)
Server = [https://mirror.lty.me/archlinux/\\$repo/os/\\$arch](https://mirror.lty.me/archlinux/$repo/os/$arch)
Server = [https://mirror.chaoticum.net/arch/\\$repo/os/\\$arch](https://mirror.chaoticum.net/arch/$repo/os/$arch)
Server = [https://archlinux-br.com.br/archlinux/\\$repo/os/\\$arch](https://archlinux-br.com.br/archlinux/$repo/os/$arch)

10. Controladores gráficos, audio y video.

10.1. ¿Cómo saber qué tipo de tarjeta gráfica debemos instalar?

Para averiguarlo basta con ingresar el comando:

```
[root@Purito arch]# lspci | grep VGA
```

o también podemos instalar una genérica:

```
[root@Purito arch]# pacman -S xf86-video-vesa xf86-video-nouveau xf86-video-intel intel-ucode --needed --noconfirm _
```

10.2. Controladores de audio y video.

Independientemente de las especificaciones del ordenador, hay ciertos paquetes de video y audio que son recomendables instalar para un correcto funcionamiento, estos se pueden simplificar en una línea de comando:

```
pacman -Sy 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
```

```
[root@Purito arch]# pacman -Sy 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
```

10.3. Actualización de sistema.

Llegado a este punto es necesario que el usuario realice una actualización completa, ya que pueden existir paquetes que lo ameriten, para lograrlo comentamos:

```
pacman -Syyu --noconfirm
```

```
[root@Purito arch]# pacman -Syuu --noconfirm
:: Synchronizing package databases...
core                  137.6 KiB   529 KiB/s  00:00 [#####] 100%
extra                1567.5 KiB   751 KiB/s  00:02 [#####] 100%
community             5.9 MiB  1387 KiB/s  00:04 [#####] 100%
:: Starting full system upgrade...
resolving dependencies...
looking for conflicting packages...

Packages (1) mesa-21.3.2-1

Total Download Size:    17.79 MiB
Total Installed Size:  106.44 MiB
Net Upgrade Size:       0.67 MiB
```

De hecho es un comando recomendable realizar al menos 1 vez al día.