



Instalación de

Arch Linux

Guía de instalación paso a paso

Félix J. Marcelo Wirnitzer

Índice

1	Descargar la ISO de Arch Linux	4
1.1	Verificación de integridad y autenticidad	4
2	Establecemos el teclado en español	5
3	Comprobamos qué tipo de arranque tiene la placa	6
4	¿Tenemos conexión a Internet?	7
5	Particionado del disco	8
5.1	Flujo de trabajo en cfdisk	8
5.2	Esquemas de particionado	9
5.3	Formatear y montar las particiones	10
6	Instalación del sistema base	12
6.1	Optimizar los mirrors	12
6.2	Actualización del catálogo de paquetes	12
6.3	Instalación	12
6.4	Paquetes instalados	13
7	Generación del fichero fstab	14
7.1	Generación de fstab	14
7.2	Entrada a la raíz	14
8	Zona horaria	15
8.1	Zona horaria	15
9	Localización	16
9.1	Localización	16
10	Configuración de la red	17
10.1	Nombre del equipo (<i>hostname</i>)	17
10.2	Fichero <i>/etc/hosts</i>	17
10.3	Gestión de red	17
11	Usuarios	18
11.1	Establecemos la contraseña de <i>root</i>	18
11.2	Nuevo usuario <i>non-root</i>	18
11.3	Privilegios <i>sudo</i> para viriato	18
12	Instalación del paquete de microcode (<i>µcode</i>)	20
13	Cargador de arranque	21
13.1	GRUB	21
13.2	Arranque BIOS	21
13.3	Arranque EFI	21
13.4	Guardamos la configuración	21

14 Repositorio AUR	22
14.1 Introducción	22
15 Instalación de yay	22
15.1 Proceso de instalación	22
16 Entornos gráficos	23
16.1 Identificar la tarjeta gráfica	23
16.2 Drivers de vídeo	23
16.3 Entornos de escritorio	24
16.4 Compositores Wayland (caso avanzado)	24
16.5 Reinicio y verificación	25
17 Últimas consideraciones	25
17.1 Fin	25

1. Descargar la ISO de Arch Linux

1. Comprobamos si la ISO tiene más de un mes. Si es así, hay que descargar la última. Siempre se publica una nueva cada primero de mes.
2. Hay que descargarla desde la [página oficial de descargas de Arch Linux](#).¹
3. Tostamos la ISO en un USB con Ventoy o con la orden dd:

Ejemplo con dd (sustituir /dev/sdX por el dispositivo real)

```
dd if=archlinux-*.iso of=/dev/sdX bs=4M status=progress && sync
```

1.1. Verificación de integridad y autenticidad

Antes de arrancar la ISO conviene verificar que la descarga es íntegra y auténtica. Si la ISO está comprometida, todo lo que viene después tampoco es de fiar.

Verificar el hash SHA256

```
1 sha256sum archlinux-*.iso
```

Compara la salida con el hash publicado en archlinux.org/download. Deben coincidir exactamente.

Verificar la firma GPG

```
1 gpg --keyserver-options auto-key-retrieve \  
2 --verify archlinux-*.iso.sig
```

Resultado esperado: gpg: Buena firma de "Pierre Schmitz <pierre@archlinux.org>". El aviso sobre clave no certificada es normal.

Si el hash o la firma no coinciden

No uses esa ISO. Vuelve a descargarla desde un mirror diferente y repite la verificación. Una ISO comprometida puede instalar software malicioso sin que lo notes.

¹<https://archlinux.org/download/>

2. Establecemos el teclado en español

Establecemos el teclado en español

loadkeys es

3. Comprobamos qué tipo de arranque tiene la placa

Comprobamos qué tipo de arranque tiene la placa

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

```
ls /sys/firmware/efi/efivars
AMITCPPIVAR-8a2093b-fefa-43c1-8662-c6526047265e
AMITSESetup-c811fa30-42c6-4579-a9bb-60e94eddff34
AsusRomLayout-2e055e9-2b5e-4f1e-bbeb-6e32c5ef44b8
BiosEventLog-4034591c-48ea-4cdc-864f-e7cb61cfd0f2
BiosSettingMappingTable-b57086d5-c2e5-4654-9e3a-0b55830fbb32
Boot0001-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
Boot0002-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
Boot0003-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
Boot0004-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
BootCurrent-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
BootFromUSB-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
BootOptionSetup-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
BootOrder-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
CMBSPartition-890dc97-5105-472c-a743-b1621e142b41
ConIn-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
ConInDev-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
ConOut-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
ConOutDev-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
CpuSetupVolatileData-b08f97ff-e6e8-4193-a997-5e9e9b0adb32
CpuSmm-90d93e09-4e91-423d-6c77-c32ff10e3c31
CurrentPolicy-77fa9abd-0359-4d32-bd60-28f4e78f784b
db-d719b2cb-3d3a-4596-a3bc-dad00e67656f
dbDefault-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
dbx-d719b2cb-3d3a-4596-a3bc-dad00e67656f
dbxDefault-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
DefaultBootOrder-45f3516-0d0e-4d04-850a-0379a3b16f53
DeploymentModNv-97e8955f-c761-4f48-b664-9ffa9cb2a2d6
EfiTime-9d0da369-540b-46f8-05a0-2b5f2c301e15
Ep-73dad563-8f27-42af-918f-8651eb0a93ef
EPCBIOS-c60aa77f-e8de-4956-8ba1-fe26298f5e87
ErrOut-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
FirstBootOption-b540530-6978-4d07-91cb-72076764d262
FirstBootFirmware-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
FPDT_Volatile-01368881-c4n1-ab1d-b631-d57a8ec8db6b
FPLayoutOrder-4db88a62-6721-47a0-9082-280b00323594
FTMActiveFlag-4034591c-48ea-4cdc-864f-e7cb61cfd0f2
HddSmartInfo-a6f44860-92e8-4fda-bd45-78368994b6ae
HiIDB-1b838190-4625-4ead-ab99-cd5e6af18f00
HM_Change_Warning-0025a1bf-fdc6-420a-8fc6-6cd9e4736a3b
IntUCode-eda41d22-7729-5b91-93ee-ba619921c6fa
KEK-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
KEKDefault-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
Kernel_ATRISStatus-77fa9abd-0359-4d32-bd60-28f4e78f784b
Kernel_EnterRevokeSISStatus-77fa9abd-0359-4d32-bd60-28f4e78f784b
Kernel_SISStatus-77fa9abd-0359-4d32-bd60-28f4e78f784b
Kernel_SkuSISStatus-77fa9abd-0359-4d32-bd60-28f4e78f784b
Kernel_WinSISStatus-77fa9abd-0359-4d32-bd60-28f4e78f784b
LastBoot-b540530-6978-4d07-91cb-72076764d262
LoaderDevicePartUUID-4a87b002-0a4c-41cf-b6c7-440b29bb8c4f
LoaderInfo-4a67b002-0a4c-41cf-b6c7-440b29bb8c4f
MaximumTableSize-4b3082a3-80c6-4d7e-9c0d-583917265df1
MemoryOverwriteRequestControl-e20939be-32d4-41be-a150-897f85d49829
MemoryOverwriteRequestControlLock-bb983cfc-151d-40e1-a07b-4a17be16829c
MSetUpStorage-542122d-0b34-49d2-adde-65a829eb4c74
MonotonicCounter-01368881-c4n1-ab1d-b631-d57a8ec8db6b
MyFav-4034591c-48ea-4cdc-864f-e7cb61cfd0f2
NBGopPlatformData-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
NetworkStackVar-d1405d16-7afc-4695-bb12-4159d36950a2
NVRAM_Verify-15a9d661-e4f8-49a9-80db-353b13d76490
NV_ST00_LD1-560bf58a-1e0d-4d7e-953f-2980d261e031
NV_ST00_LD2-560bf58a-1e0d-4d7e-953f-2980d261e031
NV_ST00_LD3-560bf58a-1e0d-4d7e-953f-2980d261e031
OCMR_CPU_Av_VARIABLE_NAME-a3a3b074-7636-4122-ba1e-f52c584494e3
OfflineUniqueIDRandomSeedCRC-eaac226f-c9a3-477a-a826-ddc716cdc0e3
OfflineUniqueIDRandomSeed-eaac226f-c9a3-477a-a826-ddc716cdc0e3
OsIndications-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
OsIndicationsSupported-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
PasswordMode-2b2a7572-f2aa-4f6e-a133-4d4c7117cee8
PCI_COMMON-ac9af304-21e2-4852-9975-77f4801d67a5
PciSataModVar-5e9a565f-cdc0-413b-ad13-1fe8713fddcd
PK-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
PKDefault-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
PlatformLang-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
PlatformLangCodes-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
PreVgaInfo-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
QFanConfig-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
QFan-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
RstOptaneConfig-4da4f952-251e-4d06-8975-65036403a8c7
RstSataV-193dfefa-a445-4302-99d0-af3aad1a04c6
SecureBoot-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
SecureBootSetup-7b59104a-c00d-4158-87ff-f4d6396a915
SetupAPMFeatures-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
SetupCpuFeatures-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
SetupUpdateComVar-91c16078-bfd6-4308-9790-570914c01a65
Setup-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
SetupPMMFeatures-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
SetupPMMOneof-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
SetupMode-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
SIBoardItemControl-0024fd50-3bc1-436c-bddd-2c92161c285
SignatureSupport-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
SmartDisplay-3fbae528-ed54-46fb-acad-9da534d4d1789
SmbiosEntryPointTable-4b3082a3-80c6-4d7e-9c0d-583917265df1
SmbiosScratchBuffer-4b3082a3-80c6-4d7e-9c0d-583917265df1
SmbiosV3EntryPointTable-4b3082a3-80c6-4d7e-9c0d-583917265df1
SOFTWAREGUARDSTATUS-9cb2e73f-7325-40f4-a044-659bb344c3cd
Stdefaultx-4599266f-1a11-89b0-891f-850745cfff024
Timeout-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
TPMPERBIOSFLAGS-7d3dcee-cbce-4ea7-8709-6e552f1edbbe
TpmServFlags-7d3dcee-cbce-4ea7-8709-6e552f1edbbe
UnlockIDCopy-eaac226f-c9a3-477a-a826-ddc716cdc0e3
UsbSupport-ec87d643-eba4-4bb5-a1e5-3f3e36b20da9
VARSORE_OCR_SETTINGS_NAME-a3a3b074-7636-4122-ba1e-f52c584494e3
VendorKeys-8be4d6f1-93ca-11d2-aa0d-00e098032b8c
WriteOnceStatus-4b3082a3-80c6-4d7e-9c0d-583917265df1
```

Si vemos esta salida entonces es un arranque **UEFI**

Instalaremos el paquete **grub** y, adicionalmente, **efibootmgr**.

Si no la vemos entonces es un arranque **BIOS**

Instalaremos únicamente el paquete **grub**.

4. ¿Tenemos conexión a Internet?

Comprobamos la conexión a Internet

```
ping -c 4 google.es
PING google.es (172.217.17.3) 56(84) bytes de datos.
64 bytes desde mad07s09-in-f3.1e100.net (172.217.17.3): icmp_seq=1 ttl=119 tiempo=63.4 ms
64 bytes desde mad07s09-in-f3.1e100.net (172.217.17.3): icmp_seq=2 ttl=119 tiempo=77.9 ms
64 bytes desde mad07s09-in-f3.1e100.net (172.217.17.3): icmp_seq=3 ttl=119 tiempo=34.3 ms
64 bytes desde mad07s09-in-f3.1e100.net (172.217.17.3): icmp_seq=4 ttl=119 tiempo=33.3 ms

--- google.es estadísticas ping ---
4 paquetes transmitidos, 4 recibidos, 0%\% packet loss, time 3004ms
rtt min/avg/max/mdev = 33.302/52.228/77.929/19.152 ms
```

Nota

En caso de que sea un portátil o un ordenador con conexión inalámbrica, se deberá usar la aplicación `iwctl`.

Pasos a seguir

Para conectarse a la red WiFi, debemos saber el nombre de nuestro dispositivo primero; para ello ejecutamos `ip link` y buscamos la interfaz inalámbrica. Para los comandos a continuación usaremos `wlan0` como nombre de dispositivo y supongamos además que nuestra SSID es `redCasa` con clave `1234567890`.

```
iwctl --passphrase 1234567890 station wlan0 connect redCasa
```

Si todo va bien debería funcionar la orden `ping`

5. Particionado del disco

En este paso dividiremos el disco en particiones. Usaremos `cdisk`, una herramienta interactiva más intuitiva que `fdisk` para quienes se acercan por primera vez.

Identificar el disco antes de empezar

```
1 lsblk
```

El nombre del dispositivo depende del tipo de unidad:

- **SATA** (disco mecánico o SSD): `/dev/sda` — particiones: `/dev/sda1`, `/dev/sda2`...
- **NVMe** (unidad M.2): `/dev/nvme0n1` — particiones añaden `p:` `/dev/nvme0n1p1`, `/dev/nvme0n1p2`...

Abre `cdisk` con el nombre que corresponda a tu disco:

Abrir `cdisk`

```
1 cdisk /dev/sda      # SATA
2 cdisk /dev/nvme0n1 # NVMe
```

5.1. Flujo de trabajo en `cdisk`

Al abrir `cdisk` en un disco vacío se pedirá el **tipo de tabla de particiones**:

- **gpt** — Para sistemas **UEFI** (hardware moderno, recomendado).
- **dos** — Para sistemas **BIOS legacy**.

Una vez dentro, el flujo para crear cada partición es el siguiente:

1. Seleccionar `Free space` con las teclas `↑↓`.
 2. Elegir **[New]** y pulsar `Enter`.
 3. Introducir el **tamaño** (ej. `512M`, `4G`, `100G`).
 4. Elegir **[Type]** y asignar el tipo correcto:
 - Partición EFI → *EFI System*
 - Raíz o home → *Linux filesystem*
 - Swap → *Linux swap*
-

5. Repetir para cada partición del esquema.
6. Elegir [**Write**], escribir **yes** y luego [**Quit**].

⚠ Los cambios no se graban hasta Write

Puedes rediseñar el esquema libremente. Solo se aplican al disco físico al confirmar con **Write**.

5.2. Esquemas de particionado

Esquema MBR (BIOS legacy) — SATA

<code>/dev/sda1</code>	<code>/</code>	Linux fs	20--50 GiB	<code>mkfs.ext4 /dev/sda1</code>
<code>/dev/sda2</code>	<code>swap</code>	Linux swap	2--4 GiB	<code>mkswap /dev/sda2</code>
<code>/dev/sda3</code>	<code>/home</code>	Linux fs	Resto del disco	<code>mkfs.ext4 /dev/sda3</code>

Esquema MBR (BIOS legacy) — NVMe

<code>/dev/nvme0n1p1</code>	<code>/</code>	Linux fs	20--50 GiB	<code>mkfs.ext4 /dev/nvme0n1p1</code>
<code>/dev/nvme0n1p2</code>	<code>swap</code>	Linux swap	2--4 GiB	<code>mkswap /dev/nvme0n1p2</code>
<code>/dev/nvme0n1p3</code>	<code>/home</code>	Linux fs	Resto del disco	<code>mkfs.ext4 /dev/nvme0n1p3</code>

Esquema GPT con BIOS — SATA

<code>/dev/sda1</code>	<code>BIOS boot</code>	BIOS boot	1 MiB	(sin formato)
<code>/dev/sda2</code>	<code>/</code>	Linux fs	20--50 GiB	<code>mkfs.ext4 /dev/sda2</code>
<code>/dev/sda3</code>	<code>swap</code>	Linux swap	2--4 GiB	<code>mkswap /dev/sda3</code>
<code>/dev/sda4</code>	<code>/home</code>	Linux fs	Resto del disco	<code>mkfs.ext4 /dev/sda4</code>

Esquema GPT con BIOS — NVMe

<code>/dev/nvme0n1p1</code>	<code>BIOS boot</code>	BIOS boot	1 MiB	(sin formato)
<code>/dev/nvme0n1p2</code>	<code>/</code>	Linux fs	20--50 GiB	<code>mkfs.ext4 /dev/nvme0n1p2</code>
<code>/dev/nvme0n1p3</code>	<code>swap</code>	Linux swap	2--4 GiB	<code>mkswap /dev/nvme0n1p3</code>
<code>/dev/nvme0n1p4</code>	<code>/home</code>	Linux fs	Resto del disco	<code>mkfs.ext4 /dev/nvme0n1p4</code>

Esquema GPT con EFI (UEFI) — SATA

<code>/dev/sda1</code>	<code>/boot (EFI)</code>	EFI System	512 MiB	<code>mkfs.fat -F32 /dev/sda1</code>
<code>/dev/sda2</code>	<code>/</code>	Linux fs	20--50 GiB	<code>mkfs.ext4 /dev/sda2</code>
<code>/dev/sda3</code>	<code>swap</code>	Linux swap	2--4 GiB	<code>mkswap /dev/sda3</code>
<code>/dev/sda4</code>	<code>/home</code>	Linux fs	Resto del disco	<code>mkfs.ext4 /dev/sda4</code>

Esquema GPT con EFI (UEFI) — NVMe

<code>/dev/nvme0n1p1</code>	<code>/boot (EFI)</code>	EFI System	512 MiB	<code>mkfs.fat -F32 /dev/nvme0n1p1</code>
<code>/dev/nvme0n1p2</code>	<code>/</code>	Linux fs	20--50 GiB	<code>mkfs.ext4 /dev/nvme0n1p2</code>
<code>/dev/nvme0n1p3</code>	<code>swap</code>	Linux swap	2--4 GiB	<code>mkswap /dev/nvme0n1p3</code>
<code>/dev/nvme0n1p4</code>	<code>/home</code>	Linux fs	Resto del disco	<code>mkfs.ext4 /dev/nvme0n1p4</code>

⚠ A tener en cuenta

Es recomendable que la partición BIOS boot o EFI sea la primera del disco. Algunos firmwares fallan si no es así.

5.3. Formatear y montar las particiones

Formateamos las particiones (GPT+EFI — SATA)

```
1 mkfs.fat -F32 /dev/sda1 # EFI
2 mkfs.ext4 /dev/sda2 # raiz
3 mkswap /dev/sda3 # swap
4 mkfs.ext4 /dev/sda4 # home
```

Formateamos las particiones (GPT+EFI — NVMe)

```
1 mkfs.fat -F32 /dev/nvme0n1p1 # EFI
2 mkfs.ext4 /dev/nvme0n1p2 # raiz
3 mkswap /dev/nvme0n1p3 # swap
4 mkfs.ext4 /dev/nvme0n1p4 # home
```

Montamos en /mnt — SATA

```
1 mount /dev/sda2 /mnt
2 mount --mkdir /dev/sda1 /mnt/boot
3 mount --mkdir /dev/sda4 /mnt/home
4 swapon /dev/sda3
```

Montamos en /mnt — NVMe

```
1 mount /dev/nvme0n1p2 /mnt
2 mount --mkdir /dev/nvme0n1p1 /mnt/boot
3 mount --mkdir /dev/nvme0n1p4 /mnt/home
4 swapon /dev/nvme0n1p3
```

Verifica con `lsblk` que las particiones aparecen montadas en sus puntos correspondientes.

6. Instalación del sistema base

6.1. Optimizar los mirrors

Antes de instalar, optimizamos la lista de mirrors para usar los más rápidos y cercanos. Esto hace que pacstrap sea mucho más rápido, especialmente en Canarias o conexiones lentas.

Seleccionamos los mirrors más rápidos de España

```
1 reflector --country Spain --sort rate --save /etc/pacman.d/mirrorlist
```

reflector consulta y ordena los mirrors por velocidad y guarda el resultado en /etc/pacman.d/mirrorlist, que es el fichero que usa pacman (y pacstrap) para las descargas. Si estás en Canarias, añade más países: --country Spain,France,Germany

6.2. Actualización del catálogo de paquetes

Primero actualizamos el catálogo de paquetes

```
1 pacman -Sy
```

6.3. Instalación

Instalación del sistema base

```
1 pacstrap -K /mnt base base-devel linux linux-firmware sudo ntfs-3g networkmanager curl  
aria2 zsh neovim grub sddm plasma openssh git
```

Si el arranque es **UEFI**, instala también **efibootmgr**: `pacman -S efibootmgr`

6.4. Paquetes instalados

! Explicación de los paquetes instalados

base:

Conjunto mínimo de paquetes para definir una instalación básica de Arch Linux.

base-devel:

Conjunto de paquetes necesarios para crear software a partir del código fuente.

linux:

El propio *kernel*.

linux-firmware:

Controladores para hardware común.

sudo:

Quieres ejecutar comandos como root, ¿verdad?

ntfs-3g:

Controlador del sistema de archivos NTFS y utilidades necesarias para trabajar con unidades NTFS.

networkmanager:

Facilita detección y configuración para que los sistemas se conecten automáticamente a las redes.

curl: Herramienta de línea de comandos y una biblioteca para transferir datos mediante URL.

aria2:

aria2 es una ligera utilidad de descarga multiprotocolo y multifuente.

zsh: *Shell* potente que funciona como shell interactivo y como intérprete de lenguajes de scripting.

neovim:

Editor de texto plano de gran potencia.

grub:

Gestor multiarranque (válido tanto para BIOS como para UEFI).

efibootmgr:

Herramienta para gestionar las entradas de arranque EFI (solo necesaria en instalaciones UEFI).

sddm:

Gestor de sesión gráfico (*Simple Desktop Display Manager*).

plasma:

Entorno de escritorio KDE Plasma.

openssh:

Protocolo de red para iniciar una sesión en una máquina remota.

git: Programa para control de versiones.

7. Generación del fichero fstab

7.1. Generación de fstab

Generación de fstab

```
1 genfstab -U /mnt >> /mnt/etc/fstab
```

Comprobamos que se ha generado correctamente

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

7.2. Entrada a la raíz

Y entramos en la partición /mnt

```
1 arch-chroot /mnt
```

8. Zona horaria

8.1. Zona horaria

Establecemos la zona horaria que es **Atlántico/Canarias**

```
1 ln -sf /usr/share/zoneinfo/Atlantic/Canary /etc/localtime
```

A continuación ejecutamos

```
1 hwclock --systohc
```

9. Localización

9.1. Localización

Editamos con `nano /etc/locale.gen`

Aquí configuramos el idioma

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

Descomentamos la línea que contenga `es_ES.UTF-8 UTF-8`

```
#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
```

Y tras guardar:

```
1 locale-gen
```

Para establecer el idioma por defecto:

```
1 echo 'LANG=es_ES.UTF-8' > /etc/locale.conf
2 echo 'KEYMAP=es' > /etc/vconsole.conf
```

10. Configuración de la red

10.1. Nombre del equipo (*hostname*)

Le damos nombre a nuestro ordenador

```
1 echo Lusitania > /etc/hostname
```

10.2. Fichero `/etc/hosts`

Añadimos las siguientes líneas en `/etc/hosts`

```
1 127.0.0.1 localhost
2 ::1      localhost
3 127.0.1.1 Lusitania
```


10.3. Gestión de red

Arch Linux ofrece dos opciones principales. Elige según el caso de uso:

 NetworkManager — recomendado para escritorio

```
1 systemctl enable NetworkManager
```

Se integra con GNOME, KDE y otros escritorios. Gestiona Wi-Fi, VPN y conexiones móviles de forma automática. Ya está incluido en el `pacstrap` del paso anterior.

 systemd-networkd — recomendado para servidor o instalación mínima

```
1 systemctl enable systemd-networkd
2 systemctl enable systemd-resolved
```

Opción más ligera. Requiere crear un fichero de configuración por interfaz en `/etc/systemd/network/`. Usa `ip link` para obtener el nombre real de la interfaz.

11. Usuarios

11.1. Establecemos la contraseña de *root*

Establecemos la contraseña del usuario *root*

```
1 passwd
```

11.2. Nuevo usuario *non-root*

Seguiremos los siguientes pasos para la creación de nuestro usuario **viriato** con su contraseña **iberia**.

Creamos el usuario y le asignamos la contraseña.

```
1 useradd -m viriato
2 passwd viriato
```

Con el parámetro **-m** indicamos que debe crear el directorio `/home/viriato`.

11.3. Privilegios *sudo* para *viriato*

Queremos darle a *viriato* privilegios de tipo *sudo*.

⚠ Edición segura de *sudoers* con *visudo*

Nunca edites `/etc/sudoers` directamente. Un error de sintaxis dejaría el sistema sin acceso a *sudo*. Usa *visudo*, que valida la sintaxis antes de guardar:

```
1 EDITOR=nano visudo
```

Configuración para entorno de clase o prácticas

```
1 viriato ALL=(ALL:ALL) NOPASSWD:ALL
```

`NOPASSWD:ALL` elimina la petición de contraseña para *sudo*. Cómodo en clase, pero **no recomendado en producción**.

Configuración recomendada para entornos reales

```
1 viriato ALL=(ALL:ALL) ALL
```

Configuración estándar: sudo pide contraseña en cada sesión. Es la opción correcta para cualquier sistema en producción o accesible desde Internet.

12. Instalación del paquete de microcode (μ code)

En Arch Linux, las actualizaciones de microcódigo están disponibles a través de paquetes oficiales que todo usuario debería instalar en sus sistemas.

Los fabricantes de procesadores como Intel y AMD lanzan a menudo actualizaciones de estabilidad y seguridad para el procesador. Estas actualizaciones son cruciales para la estabilidad del sistema.

En el caso de Intel

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

En el caso de AMD

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

13. Cargador de arranque

13.1. GRUB

Configuramos GRUB, el cargador de arranque. Debemos tener en cuenta si la placa tiene BIOS o UEFI como vimos al principio.

13.2. Arranque BIOS

Si se trata de un arranque con disco mecánico o una unidad SSD

```
1 grub-install --target=i386-pc /dev/sda
```

Si se trata de un arranque con una unidad M.2

```
1 grub-install --target=i386-pc /dev/nvme0n1
```

13.3. Arranque EFI

Si se trata de un arranque UEFI

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

13.4. Guardamos la configuración

Finalmente guardamos la configuración:

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

Y si no muestra un mensaje de error, podemos reiniciar comprobando que podemos arrancar Arch Linux.

14. Repositorio AUR

14.1. Introducción

El **Arch User Repository (AUR)** es un repositorio comunitario para usuarios de Arch.

Contiene descripciones de paquetes (PKGBUILDS) que permiten compilar un paquete desde el código fuente con `makepkg` y luego instalarlo mediante `pacman`.

El **AUR** se creó para organizar y compartir nuevos paquetes de la comunidad y para ayudar a acelerar la inclusión de paquetes populares en el repositorio extra.

La aplicación que usaremos para instalar paquetes de los repositorios de **AUR** será `yay`.

15. Instalación de `yay`

15.1. Proceso de instalación

Precaución:

`yay` no puede ser ejecutado por **root**. Los siguientes pasos deben realizarse como el usuario **viriato**. Cambia de usuario antes de continuar:

```
1 su - viriato
```

Pasos a seguir:

Descargamos los ficheros desde el repositorio **AUR**:

```
git clone https://aur.archlinux.org/yay.git
```

Ahora vamos al directorio y compilamos:

```
cd yay
makepkg -si
```

Para instalar software solo hay que escribir `yay` en lugar de `pacman` sin necesidad de usar la orden `sudo`.

Desde este momento podemos instalar todos los paquetes con `yay`.

16. Entornos gráficos

Antes de instalar un escritorio hay que asegurarse de que los drivers de vídeo están correctamente instalados. Es uno de los puntos donde más alumnos se atascan.

16.1. Identificar la tarjeta gráfica

Identificamos el hardware gráfico

```
1 lspci | grep -i vga
```

16.2. Drivers de vídeo

Intel o AMD — Mesa (recomendado)

```
1 yay -S mesa
```

Proporciona soporte OpenGL y Vulkan para Intel y AMD. Opción por defecto y recomendada para Wayland.

NVIDIA — driver propietario

```
1 yay -S nvidia nvidia-utils
```

Para RTX 2000 o más nuevas puede usarse también `nvidia-open`. Para kernels LTS, usa `nvidia-lts`.

Hardware antiguo o genérico

```
1 yay -S xf86-video-vesa
```

16.3. Entornos de escritorio

Instalar GNOME

```
1 yay -S gnome gdm
2 systemctl enable gdm
```

Entorno completo con excelente soporte Wayland. gdm es su display manager nativo.

Instalar KDE Plasma

```
1 yay -S plasma sddm
2 systemctl enable sddm
```

Altamente configurable. sddm es el display manager recomendado por KDE.

Instalar XFCE (opción ligera)

```
1 yay -S xfce4 xfce4-goodies lightdm lightdm-gtk-greeter
2 systemctl enable lightdm
```

Ideal para hardware con pocos recursos. Consume sensiblemente menos RAM que GNOME o KDE.

16.4. Compositores Wayland (caso avanzado)

Hyprland — tiling compositor Wayland

```
1 yay -S hyprland waybar wofi kitty
```

Se inicia desde la TTY con Hyprland. Sin necesidad de display manager.

Sway — equivalente Wayland de i3

```
1 yay -S sway swaybar dmenu foot
```

Se inicia con sway. Más estable y conservador que Hyprland.

16.5. Reinicio y verificación

Salimos del chroot y reiniciamos

```
1 exit
2 umount -R /mnt
3 reboot
```

Al arrancar deberías ver la pantalla de login del display manager. Si el sistema arranca en TTY, comprueba: `systemctl status gdm` (o `sddm`, `lightdm`).

17. Últimas consideraciones

17.1. Fin

Con esto podemos dar por terminada la instalación de Arch Linux.

En caso de dudas siempre se puede consultar su wiki en español

[https://wiki.archlinux.org/title/Main_page_\(Español\)](https://wiki.archlinux.org/title/Main_page_(Español))
