

---

# **Debian nativo en una RPi**

**Angel de la Iglesia**

**25 de abril de 2025**



---

## Contenido:

---

<b>1. Introducción</b>	<b>3</b>
1.1. Procedimiento para instalar <i>Debian</i> «nativo» en una <i>Raspberry Pi</i> . . . . .	5



---

**Ediciones del documento**

20250425

20220430

20210910 (interna)

---



# CAPÍTULO 1

---

## Introducción

---

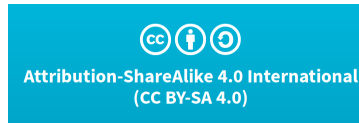
En este documento se describe como instalar el *Sistema Operativo Debian* en una *Raspberry Pi*. Esta tarjeta, que es un ordenador completo, tiene su propio sistema operativo oficial: *Raspberry Pi OS (de 32 o 64-bits)* que se puede obtener de <https://www.raspberrypi.com/software/operating-systems/> y que es un derivado de *Debian estable*, optimizado para esta tarjeta. Yo prefiero trabajar con un sistema *Debian* nativo y en su versión *testing*, si ya la versión estable tiene más de 3 o 6 meses de antigüedad.

A continuación te muestro el procedimiento para realizar una instalación de *Debian* en una tarjeta *microSD* ya sea para la versión *estable* o la *testing*.

Básicamente, la selección de la imagen de partida (ver <https://wiki.debian.org/RaspberryPi> y la respuesta 6 de <https://raspi.debian.net/faq/>) depende de la *familia* a la que pertenece la tarjeta de la *Raspberry Pi*.

- Las *Raspberry Pi Zero*, *Raspberry Pi Zero W* o *Raspberry Pi Zero WH* utilizan un chip que se corresponde con una arquitectura de 32 bits de tipo *armel* lo que supone que no dispone de una unidad de coma (punto en inglés) flotante (*FPU Floating Point Unit*) para las operaciones de cálculo. Los cálculos los hace mediante software y por lo tanto es más lento que si dispusiera de una *FPU*. En la notación que utiliza la wiki de *Debian* es de la **familia 0/1**. No consideraré la *Raspberry Pi Zero* porque no tiene conectividad con la que acceder a *Internet* mientras que las que sí la tienen, *W* y *WH*, solo se diferencian en si tiene (*WH*) o no (*W*) la tira de pines para conexiones (*GPIO*) soldada. A esta familia pertenecen también las *Raspberry Pi A*, *B*, *A+*, *B+* y las mencionadas *Zero*, *Zero W*)
- La *Raspberry Pi 2* tiene arquitectura *armhf* de 32 bits con 1 GB de memoria *RAM*. En la notación que utiliza la wiki de *Debian* es de la **familia 2**
- La *Raspberry Pi Zero 2W* es de 64 bits con *FPU*. Su arquitectura es *armhf*. Tiene, como la *Zero W* 512 MB de memoria *RAM* por lo que si instalas un escritorio (sí, se puede instalar un escritorio y tener un entorno gráfico) no será muy rápido (pero un poco más rápida que si lo pruebas en una *Zero W*). En la notación que utiliza la wiki de *Debian* es de la **familia 3**. A esta familia pertenecen también las *Raspberry Pi 3*, *3A+*, *3B+* y la mencionada *Zero 2 W*).
- Las *Raspberry Pi 4* y *Raspberry Pi 400* son las versiones más potentes, con arquitectura *arm64* de 64 bits y 4 GB de memoria. La *Raspberry Pi 4* dispone de una versión con 8 GB de memoria *RAM*. En la notación que utiliza la wiki de *Debian* es de la **familia 4**.

Este documento está bajo [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](#). Todas las marcas comerciales referidas en este documento pertenecen a sus legítimos propietarios.



La última versión de este documento está en [https://github.com/aig-microC/Debian\\_en\\_Rpi](https://github.com/aig-microC/Debian_en_Rpi).



## 1.1 Procedimiento para instalar *Debian* «nativo» en una *Raspberry Pi*

Lo primero que hay que hacer es descargar la imagen básica desde el enlace <https://raspi.debian.net/>. Hay dos posibilidades: descargar la imagen creada diariamente (<https://raspi.debian.net/daily-images/>) o de la versión probada (<https://raspi.debian.net/tested-images/>). Yo suelo utilizar las imágenes creadas diariamente y si encuentro problemas en la instalación me descargo la imagen testeada.

**Nota:** Aunque la página dice que las imágenes son creadas diariamente si observas la fecha de los ficheros verás que (posiblemente) no son muy recientes. No te preocupes cuando actualices tu sistema se configurará con los últimos paquetes de la distribución.

Una vez descargada hay que escribirla en una tarjeta *microSD*. Para ello utilizaremos, si tenemos instalado el *Raspberry Pi OS* en una *RPi 4* o *400* el programa *rpi-imager* que que está disponible en este sistema operativo o si estamos en un sistema *Debian* se puede compilar e instalar siguiendo las instrucciones que se pueden ver en <https://github.com/raspberrypi/rpi-imager>. También es posible bajar una *AppImage* de <https://github.com/raspberrypi/rpi-imager/releases>. Una vez descargada la tienes que hacer ejecutable con `chmod +x Raspberry_Pi_Imager-x.y.z-x86_64.AppImage` y luego ejecutarla con `./Raspberry_Pi_Imager-x.y.z-x86_64.AppImage`.



Figura 1: *rpi-imager*. Programa para copiar una imagen del SO

Seleccionamos *ELEGIR SO* (*CHOSE OS*) y vamos al final para seleccionar «Usar personalizado, *Seleziona un .img personalizado de su ordenador*» («Use custom, *Select a custom .img from your computer*») y buscamos la imagen que acabamos de descargar. A continuación seleccionamos «*ELEGIR ALMACENAMIENTO*» («*CHOSE STORAGE*») y seleccionamos la memoria *microSD*. Y por último pulsamos «*SIGUIENTE*» y en la siguiente ventana para personalizar los ajustes del *SO* pulsamos **NO**. A continuación aparece una ventana de advertencia de borrado del «**MASS STORAGE DEVICE**» y si seleccionas **SI** borrará toda la memoria que hayas seleccionado (tienes que prestar mucha atención para no equivocarte) e instalará el *Sistema Operativo*.

También lo podemos hacer utilizando la línea de comandos, con el procedimiento que se describe en <https://raspi.debian.net/how-to-image/> y que básicamente consiste en, situándonos en el subdirectorio donde hayamos descargado la imagen, teclear:

```
1 $ xzcat fichero_imagen_descargado.img.xz | sudo dd of=/dev/{indicador de la tarjeta SD} \
  bs=64k oflag=dsync status=progress
```

**Advertencia:** ¡Asegúrate de tener el indicador correcto para la tarjeta SD! (¡corres el riesgo de perder el disco de tu sistema si no identificas bien la tarjeta del nuevo sistema operativo!)

Una manera fácil de ver cual es el dispositivo al que está conectada nuestra tarjeta *microSD* es el siguiente:

- Con la tarjeta *microSD* desconectada del ordenador teclear el comando `lsblk` y observar que elementos hay en nuestro sistema debajo de `/dev/`. Por ejemplo en mi sistema obtengo lo siguiente:

```
1 angel@debianHP:~$ lsblk
2 NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
3 sda          8:0    0   1,8T  0 disk
4 └─sda1       8:1    0   1,8T  0 part /
5 └─sda2       8:2    0    1K  0 part
6 └─sda5       8:5    0   975M  0 part [SWAP]
7 nvme0n1     259:0    0 238,5G  0 disk
8 └─nvme0n1p1 259:1    0   260M  0 part
9 └─nvme0n1p2 259:2    0    16M  0 part
10 └─nvme0n1p3 259:3    0 237,2G  0 part
11 └─nvme0n1p4 259:4    0   980M  0 part
12 angel@debianHP:~$
```

- Y, a continuación, conectamos nuestra tarjeta *microSD*, repetimos el comando y vemos la diferencia,

```
1 angel@debianHP:~$ lsblk
2 NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
3 sda          8:0    0   1,8T  0 disk
4 └─sda1       8:1    0   1,8T  0 part /
5 └─sda2       8:2    0    1K  0 part
6 └─sda5       8:5    0   975M  0 part [SWAP]
7 mmcblk0     179:0    0   29,7G  0 disk
8 └─mmcblk0p1 179:1    0   29,7G  0 part /media/angel/ext4-MICROSD
9 nvme0n1     259:0    0 238,5G  0 disk
10 └─nvme0n1p1 259:1    0   260M  0 part
11 └─nvme0n1p2 259:2    0    16M  0 part
12 └─nvme0n1p3 259:3    0 237,2G  0 part
13 └─nvme0n1p4 259:4    0   980M  0 part
14 angel@debianHP:~$
```

Vemos que el dispositivo que hay que utilizar es `/dev/mmcblk0`. Podéis ver que yo tenía formateada la tarjeta *microSD* como ext4, como indica la etiqueta con la que la creé.

**Nota:** Puede ser conveniente que, si la tarjeta *microSD* no es nueva, la formatees previamente. Lo puedes hacer con un interfaz gráfico con un programa como *GParted*, o desde la línea de comandos tal como se muestra a continuación. Si tienes instalados los paquetes *parted dosfstools* y *mttools* puedes ignorar las dos primeras líneas.

```

1 $ sudo apt update
2 $ sudo apt install parted dosfstools mtools
3 $ sudo umount /media/angel/ext4-MICROSD
4 $ sudo parted /dev/mmcblk0 --script -- mklabel msdos
5 $ sudo parted /dev/mmcblk0 --script -- mkpart primary fat32 1MiB 100%
6 $ sudo mkfs.vfat -F32 /dev/mmcblk0
7 $ sudo mlabel -i /dev/mmcblk0 ::mietiqueta

```

### 1.1.1 Conexión a Internet

Si tu tarjeta tiene conector RJ45 para conectar una línea ethernet puedes conectarte al router con el cable. La conexión a internet será sencilla y no hay que hacer nada. Si no hay conector RJ45 o solo puedes conectar por *wifi* es necesario hacer dos cosas más en la *microSD* antes de ponerla en el zócalo de la tarjeta *Raspberry Pi*.

La tarjeta que acabamos de crear tiene dos particiones: *RASPIFIRM* donde están todos los ficheros para el arranque del Sistema Operativo y *RASPIROOT* que contiene la estructura de ficheros de nuestro Sistema *Debian*.

Para tener conexión a *Internet* debemos editar el fichero */etc/network/interfaces.d/wlan0* que está en la partición *RASPIROOT* de la tarjeta que acabamos de crear. Este fichero está en la partición *EXT4* de la *microSD* y seguramente no te dejará editarla con tu usuario normal. Para poder editarlo deberás hacerlo como usuario *root* o mejor mediante *sudo*:

```
$ sudo pluma /media/angel/RASPIROOT/etc/network/interfaces.d/wlan0
```

En lugar del editor *pluma* puedes utilizar el que más re guste: *vi*, *emacs featherpad*, *gedit*, etc.

El contenido del fichero deberá ser como el siguiente

```

1 # To enable wireless networking, uncomment the following lines and -naturally-
2 # replace with your network's details.
3 #
4 allow-hotplug wlan0
5 iface wlan0 inet dhcp
6 #iface wlan0 inet6 dhcp
7     wpa-ssid "tu_identificador_de_red"
8     wpa-psk "tu_contraseña"

```

En la línea 5 o 6 deberás descomentar la correspondiente a tu router: descomenta la línea 5 si tienes direccionamiento *IPv4* (del estilo 192.168.1.23) o descomenta la línea 6 si tienes direccionamiento *IPv6* (del estilo 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

En las líneas 7 y 8 deberás escribir los datos de tu red.

### 1.1.2 Conexión a la tarjeta *Raspberry Pi* sin monitor

Si no dispones de monitor para conectar a tu tarjeta *Raspberry Pi* puedes conectarte desde tu ordenador anfitrión (el que has utilizado para grabar la imagen) mediante *ssh*.

Para ello debes modificar, antes del primer arranque de la *RPi* el fichero */media/usuario/RASPYFIRM/sysconf.txt* que está en la partición *RASPIFIRM* que está formateada com *msdos* y la podrás editar sin utilizar *sudo* como en el caso del fichero *wlan0*.

Previamente has de generar la clave de autorización para poder acceder mediante *ssh*. Para ello hay que generar unos ficheros (*id\_rsaP* y *id\_rsa.pub*) que se crean en tu subdirectorio raíz bajo */home/usuario/.ssh* mediante el comando:

```
ssh-keygen -t rsa
```

Cuando lo ejecutes te pedirá una *passphrase* que es una contraseña (nos recomienda que sea una frase larga) que deberás recordar para poder utilizar el *ssh*. Si lo dejas en blanco no te pedirá esta contraseña pero el sistema estará más desprotegido.

Una vez ejecutado este comando debemos copiar el contenido del fichero *id\_rsa.pub*, en el fichero */media/usuario/RASPYFIRM/sysconf.txt* descomentando la línea 28 «*root\_authorized\_key=*» y pegando el contenido a partir del signo = y sin dejar ningún espacio en blanco entre el = y lo pegado. El resultado deberá quedar de forma similar a:

```
# root_pw - Set a password for the root user (by default, it allows
# for a passwordless login)
#root_pw=FooBar

# root_authorized_key - Set an authorized key for a root ssh login
root_authorized_key=ssh-rsa_
  AAAAB3NzaC1yc2EAAAADAQABAgQCYuJLQDeYhvitA4rCS9xRgBEf2L+sCsadGScC2H2VJN0//
  G3K0W57IX58t2ptaJunfEakxxZ+fperXLFIDBQaeDLQyQXwI8CA/
  Dj4DRjQMh9MuRRWMBRtMYz0rdIc0bnlHVh5q9SgJJReyfs4sibEV4wZUq3GIe+8lqGlpbGqevDdU/
  TY6swHBs8Ff+N187xHCvd6NJCByYDwddnUSj4WPdsOdUgiGMpmXII5M9zGqztycAuwpeMeW13L2GBCEI8q6pqUWUwIOMNsPN/
  BVFqsnrnXxBFVMS7CSGkFCeUwvQDz9LY2gHbs9x9lJWLT0D9dg2YkCuGDBe/
  H0k7EdsH3FXs079YdUfnA9lyWe81oGlBwIYqo2+bAx91sbc/DjhML9G0UsxsHGC/
  88BFeaQ8jmwoi21x0ZBN0kfAQaR5u1cSZYE5340cy5/FZvF5PWiv7XZQuVS6VeJg2H9n3il/APAuB34XjIyvr/
  hzJrASr4YtNkAtKi2FlIxntUkAYbmK= usuario@debian

# hostname - Set the system hostname.
#hostname=rpi
```

Además debemos conocer las direcciones *IP* de nuestra red. Para ello es necesario tener instalado el programa *nmap*. Si no lo tienes instalado lo puedes instalar con:

```
$ sudo apt install nmap
```

Alimentamos la tarjeta *Raspberry Pi* con la *microSD* en su zócalo y esperamos unos minutos para proseguir. La *Raspberry Pi* puede tardar unos minutos en arrancar completamente.

Para poder conectarnos con la tarjeta *Raspberry Pi* utilizamos el comando *hostname -I* para saber cual es la dirección *IP* de nuestro *PC* y una vez identificado utilizamos *nmap* con la dirección de nuestro *PC* pero en el rango *1-255* para descubrir la dirección de nuestra *Raspberry Pi*.

```
$ hostname -I
192.168.1.105
$ nmap -sP 192.168.1.1-255
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-13 17:17 CEST
Nmap scan report for mi_router (192.168.1.1)
Host is up (0.0014s latency).
Nmap scan report for rpi1-20230908.home (192.168.1.106)
Host is up (0.048s latency).
Nmap scan report for debHP5T.home (192.168.1.105)
Host is up (0.00073s latency).
Nmap done: 255 IP addresses (3 hosts up) scanned in 5.37 seconds
```

En este ejemplo la *Raspberry Pi* está en la dirección *192.168.1.106*.

Para conectar con la *Raspberry Pi* tecleamos lo siguiente (en este paso nos pedirá la *passphrase* si es que la creamos anteriormente):

```

1 >>DebianNativoRPi$ ssh root@192.168.1.106
2 Linux rp1-20230915 6.1.0-11-rpi #1 Debian 6.1.38-4 (2023-08-08) armv6l
3
4 The programs included with the Debian GNU/Linux system are free software;
5 the exact distribution terms for each program are described in the
6 individual files in /usr/share/doc/*/copyright.
7
8 Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
9 permitted by applicable law.
10 root@rp1-20230915:~#

```

**Nota:** Inicialmente el usuario root no tiene *password*

Y ya puedes empezar a trabajar con un monitor remoto.

**Advertencia:** Lo que viene a continuación en este apartado es válido para cuando ya tengas instalado el sistema con un usuario creado. No es aplicable al estado actual de la instalación.

Si ya tuviéramos el usuario con password (cuando tengamos el S.O. instalado y operativo), la respuesta sería similar a:

```

1 $ ssh usuario@192.168.1.106
2 The authenticity of host '192.168.1.106 (192.168.1.106)' can't be established.
3 ED25519 key fingerprint is SHA256:iG2kEuxjKaxgRYdF7gj3den/J0NsNM7fPoe/ZkFbskM.
4 This key is not known by any other names.
5 Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
6 Warning: Permanently added '192.168.1.106' (ED25519) to the list of known hosts.
7 usuario@192.168.1.106's password:
8 Linux rp1-20230908 6.1.0-11-rpi #1 Debian 6.1.38-4 (2023-08-08) armv6l
9
10 The programs included with the Debian GNU/Linux system are free software;
11 the exact distribution terms for each program are described in the
12 individual files in /usr/share/doc/*/copyright.
13
14 Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
15 permitted by applicable law.
16 Last login: Thu Sep 14 09:28:33 2023 from 192.168.1.28

```

Donde *usuario* será en nombre del usuario que hayas creado.

**Nota:** Cuando tengas instalado el sistema completo podrás conectarte mediante *ssh* con un ordenador (como estamos haciendo ahora) pero también lo puedes hacer con una tablet o un teléfono móvil.

### 1.1.3 Configuración del sistema operativo

Una vez grabado el *SO* en la *microSD* la insertamos en nuestra *RPi* y esperamos a que aparezca el *login* de entrada al sistema. Entramos con el usuario *root* y veremos que **no** nos pide contraseña. Los pasos para configurar nuestro sistema en español son los siguientes:

1. Actualización del sistema

```
1 # apt update
2 # apt upgrade
```

**Nota:** La primera vez que tecleas *apt update* el reloj del sistema puede que no esté sincronizado y puede producir un error de repositorio antiguo. La segunda vez que repitas el comando es posible que ya se haya sincronizado y ya lo acepte sin error. Puedes hacer *ping google.es* y ver si tienes o no conexión de internet.

**Advertencia:** Si cuando haces *apt upgrade* aparecen muchos paquetes bloqueados o retenidos dile *n* cuando te pregunta si deseas proceder con la actualización y cambia el comando anterior *apt upgrade* por *apt full-upgrade*.

Si al hacer la actualización (*apt upgrade* o *apt full-upgrade*) hace preguntas contesta la respuesta por defecto (*Intro*).

2. Añadir una *password* para el usuario *root*.

```
1 # passwd
2 New password: 'Tu_password_para_root'
3 Retype new password: 'Tu_password_para_root'
```

3. Instalamos el programa *sudo* que permite a un usuario normal tener los privilegios de *root*.

```
1 # apt install sudo
```

4. Creamos un usuario nuevo. Yo voy a crear, como ejemplo, el usuario *usuario*. Te pedirá el nombre del usuario y su contraseña. El resto de los campos que solicita son opcionales (yo los dejo en blanco pulsando *Intro*) y al final pedirá confirmación.

```
1 # adduser usuario
```

**Nota:** En usuario pon el nombre que quieres para ti en el sistema.

5. Para que *usuario* pertenezca al grupo *sudo* hacemos lo siguiente:

```
1 # usermod -aG sudo usuario
```

Aquí tenemos un problema. Todavía no tenemos instalado el teclado y las *Locales* en español, por lo que el guión «-» no está en la tecla de nuestro teclado. Podemos ver en [https://es.wikipedia.org/wiki/Distribuci%C3%B3n\\_del\\_teclado](https://es.wikipedia.org/wiki/Distribuci%C3%B3n_del_teclado) que el guión en el teclado del Reino Unido o de Estados Unidos está en la tercera tecla, por la derecha, de la fila de números y símbolos del teclado y que en el teclado español se corresponde con la tecla «'» (comilla simple) la que tiene el «?» cuando pulsamos la tecla *Shift* o *Mayúsculas*.

**Nota:** Si estás usando *ssh* no tendrás este problema ya que el programa de *consola* de tu ordenador anfitrión ya estará

configurado para el idioma español.

6. Ahora podemos reiniciar el sistema y entrar como usuario *root* o *usuario* con su contraseña correspondiente.

```
1 # reboot
```

Cuando arranque de nuevo entramos como usuario *usuario* y su contraseña

```
1 login: usuario
2 Password:
```

7. A continuación instalamos las *locales*. Al utilizar *sudo* nos pedirá la password de *usuario* para proceder.

```
1 $ sudo apt install locales
2 $ sudo dpkg-reconfigure locales
```

Y seleccionaremos, con la *barra de espacio*, dos: *en\_US.UTF8 UTF-8* y *es\_ES.UTF8 UTF-8*. Pulsamos *<tabular>* y *ok* y cuando nos pregunte que *locale* queremos que sea nuestra local por defecto seleccionamos *es\_ES.UTF-8*. Con esto el teclado todavía no está configurado en español.

8. Configuración del teclado en español. Para ello hacemos lo siguiente:

```
1 $ sudo apt install keyboard-configuration
```

**Nota:** Todavía no tenemos configurado el teclado en español por lo que deberemos usar nuevamente la tecla «*l*».

En la primera pantalla seleccionamos *Other*, pulsamos *<tabular>*, *ok* e *Intro*. De la lista que aparece seleccionamos *Spanish* y *ok* y, luego, *Spanish - Spanish (Windows)* y *ok*.

9. Ahora volvemos a reiniciar el sistema y entrar como usuario *usuario* con su contraseña correspondiente.

```
1 $ sudo reboot
```

10. A continuación instalamos los paquetes necesarios para la consola:

```
1 $ sudo apt install gpm console-common console-data console-setup
```

**Nota:** El paquete *gpm* es para poder usar el ratón en la consola.

Veremos que ahora la *Configuración de console-data* ya nos aparece en español, aunque el teclado todavía no funciona en español.

Seleccionamos la opción *Elegir el mapa de teclado de la lista completa* y seleccionamos *<Aceptar>* y seleccionamos *pc / qwerty Spanish / Standard / Standard* y *<Aceptar>* ya tendremos el teclado en español. Verás, además, que si mueves el ratón el cursor se moverá por la pantalla.

11. Este paso es opcional. En la consola el tipo de caracteres (*fuentes*) que se han instalado es *Fixed* que tiene una mejor cobertura para los *scripts* internacionales. A mí, particularmente me gusta más los tipos *VGA*. En cualquier caso si deseas configurar los tipos de caracteres de la consola puedes hacer:

```
1 $ sudo dpkg-reconfigure console-setup
```

**Nota:** Ahora ya sí, el guión «-» está en la tecla de nuestro teclado.

Seleccionar *UTF-8* y luego *#Latino1* y *Latino5 - Europa Occidental y lenguas turcas* y ahora el tipo que desees. Yo elijo *VGA* como he comentado anteriormente y un tamaño de *8x16*. Verás que en este momento la consola presenta los tipos *VGA*.

12. A continuación vamos a instalar el sistema básico. Si quieres instalar un sistema de escritorio lo más práctico es utilizar *tasksel*. Si estás instalando un sistema mínimo, sin entorno gráfico, en una *Raspberry Pi Zero*, por ejemplo, salta al punto 13.

```
1 $ sudo tasksel
```

Y seleccionamos mediante la *barra de espacio* *Debian desktop environment* y el escritorio que más te guste, teniendo en cuenta que el escritorio que elijas puede consumir muchos recursos. Yo elijo *LXQT* porque es el *original* del que utiliza *Raspberry Pi OS* y sobre todo porque consume muy pocos recursos. Este paso dura bastante tiempo y es posible que la pantalla se desconfigure. No te preocupes y deja que siga hasta que termine.

13. Cambiamos la hora a nuestra hora local. Para ello tecleamos `timedatectl list-timezones` y buscamos cual es nuestra zona. En mi caso *Europa/Madrid* y hacemos

```
1 $ sudo timedatectl set-timezone Europe/Madrid
2 $ date
```

Y verás que la hora ya está actualizada a tu zona.

14. Ya solo queda reiniciar el sistema para tener un Sistema Operativo instalado en nuestra *RPi*.

```
1 $ sudo reboot
```

15. Entramos en nuestro nuevo sistema con nuestro usuario y contraseña. Normalmente tardará unos minutos, mientras se configura, hasta que se vea el *prompt*, por último, abrimos un terminal, si estamos en un entorno gráfico, y actualizamos y limpiamos nuestro sistema.

```
1 $ sudo apt update && sudo apt upgrade && sudo apt full-upgrade && sudo apt clean && sudo ↵
↵ apt autoremove
```

Ahora ya solo queda configurar el Sistema como más te guste y con las aplicaciones que necesites.

### 1.1.4 Creación de un fichero *SWAP*

Yo no utilizo la memoria *SWAP* para *suspender* la tarjeta procesadora, la uso para ayudar a la memoria *ram* a aumentar su tamaño para los programas que necesitan más memoria. Este apartado está basado en esta [página de debian](#).

La tarjeta que hemos configurado no tiene partición *SWAP*. Podemos crear una partición *SWAP* en nuestra tarjeta de memoria poniendo la tarjeta *SD* en el lector de tarjetas de nuestro ordenador anfitrión para redimensionar (disminuir) la partición actual y crear una nueva partición en el espacio libre creado, por ejemplo con el programa *Discos* (*gnome-disk-utility*) pero aquí te voy a enseñar a como crear un fichero *SWAP* sin necesidad de crear una partición. El procedimiento es el siguiente:

Hay que disponer de la utilidad *fallocate* que está en el paquete *util-linux*. Si no lo tienes instalado lo puedes instalar con `sudo apt install util-linux`.

- 0) verifica que no tienes *SWAP*.



```
sudo swapon --show
```

Si has seguido el procedimiento esto no debería mostrarte *nada*.

- 1) Creamos un fichero que hará las funciones de memoria *SWAP*

```
sudo fallocate -l 1G /fichero_swap
```

El tamaño que pongo en este ejemplo es de 1G porque lo voy a implementar en una *Raspberry Pi Zero* que tiene 512k de memoria *ram*, cuando lo hago para una *4* o *400* un valor recomendable puede ser 4G (depende de si necesitas más o menos memoria *RAM*). El nombre del fichero *fichero\_swap* es arbitrario, tú puedes elegir otro nombre para el fichero.

**Nota:** Puedes ver la memoria usada (*RAM* y *SWAP* que aparece como *Inter:*) mediante el comando `free -h`. Si ves que con tu uso la memoria *SWAP* está siempre sobrada puedes disminuir el fichero, por el contrario si usas programas que consumen mucha memoria y frecuentemente la memoria *SWAP* está próxima a completarse deberás aumentar el tamaño.

- 2) Garantizar que el archivo no sea legible para todo el mundo, por razones de seguridad.

```
sudo chmod 600 /fichero_swap
```

- 3) Hacer que el *fichero\_swap* pueda ser utilizado como *SWAP*

```
sudo mkswap /fichero_swap
```

- 4) Activar el fichero de intercambio

```
sudo swapon /fichero_swap
```

Y habrá que confirmar que se ha creado la nueva memoria de intercambio con `sudo swapon --show`, que en este caso sí deberá producir salida:

```
sudo swapon --show
```

```
NAME          TYPE  SIZE USED PRIO
/fichero_swap file 1024M  0B   -2
```

- 5) Y para finalizar hay que añadir en `/etc/fstab` la entrada para que la memoria *SWAP* se active al arrancar el sistema. Para ello hay que editar el fichero

```
sudo vi /etc/fstab
```

Y añadir al final del mismo las líneas:

```
# Fichero Swap creado en /Fecha/
/fichero_swap none swap sw 0 0
```

Guarda el fichero y ya podrás disponer de la memoria *SWAP* cuando vuelvas a arrancar la tarjeta.