

# Curso de Raspberry Pi desde cero

En esta guía básica vamos a adentrarnos en la plataforma Raspberry Pi, la cual os sonará a muchos de antes. Nos introduciremos en la plataforma, las características, configuración inicial y algunos ejemplos, siguiendo los puntos a continuación:

## Contenido

1.	Introducción .....	3
1.1.	¿Qué es Raspberry Pi? .....	3
1.2.	¿Para quién es el curso de Raspberry Pi? .....	3
1.3.	¿Qué es la popular Raspberry? .....	4
1.4.	Historia .....	4
1.5.	¿Qué puedo hacer con la Raspberry? .....	4
1.6.	Características .....	4
2.	Instalación del sistema, RPi como PC .....	5
2.1.	Instalación NOOBS .....	5
2.2.	Conexiones y puesta a punto .....	5
2.3.	Raspberry Pi OS, conocer el sistema operativo .....	7
2.3.1.	Configuración mediante el menú propio.....	10
2.3.2.	Apagado .....	13
2.3.3.	Conclusión .....	13
3.	Programación .....	13
3.1.	Scratch .....	13
3.2.	Python .....	14
4.	Consola de comandos y conceptos básicos de Linux.....	15
4.1.	Comandos Bash .....	16
4.2.	Actualización general .....	17
5.	Pines GPIO.....	18
5.1.	Programación GPIO .....	22
5.1.1.	Mediante consola .....	22
5.1.2.	GPIO y WiringPi .....	24
5.2.	Scripts, programas simples .....	26
5.2.1.	Script encendido de LED .....	26
6.	Crear un usuario nuevo .....	27
6.1.	Crear y borrar usuarios.....	27
6.2.	Gestión de sesiones de usuario.....	28
6.3.	Grupos de usuarios .....	29
7.	Configuración IP estática .....	31
7.1.	Configuración desde el menú .....	31
7.2.	Configuración desde la consola.....	34

8.	Conectarse remotamente a la RPi .....	35
8.1.	Conectarse con SSH.....	35
8.1.1.	Linux .....	35
8.1.2.	Windows .....	36
8.2.	Conectarse mediante VNC.....	37
9.	Instalación y comunicación a través de UART .....	39
9.1.	Instalación Raspberry Pi OS Lite .....	40
9.2.	Conexión puerto UART .....	42
10.	Usar una cámara en RPi .....	48
10.1.	Conexión de la cámara.....	49
10.2.	Hacer fotografías .....	49
10.3.	Comandos.....	51
10.4.	Transmisión de imágenes con Motion.....	52
10.5.	Detección de movimiento en RPi .....	54
11.	Señal PWM en Python .....	55
12.	Resumen .....	58

## 1. Introducción

En los últimos años, dos plataformas han ganado una enorme popularidad en el mundo de la electrónica. ¡Gracias a Arduino y Raspberry Pi, cualquiera puede empezar a crear proyectos electrónicos interesantes!

El curso Raspberry Pi desde cero es una serie de artículos que presentar a todo el mundo sobre esta computadora de placa única (SBC) extremadamente popular.

### 1.1. ¿Qué es Raspberry Pi?

¡La Raspberry Pi es el mini-ordenador del tamaño de una tarjeta de crédito más popular del mundo! Gracias al precio relativamente bajo y las enormes posibilidades, rápidamente ganó millones de fanáticos. En el pequeño PCB encontramos, entre otros, un procesador de 4 núcleos, conexiones RAM, USB, Ethernet y HDMI. La última versión (utilizada en este curso) también está equipada con un módulo WiFi y Bluetooth.

### 1.2. ¿Para quién es el curso de Raspberry Pi?

El curso ha sido preparado completamente para principiantes. No se requieren conocimientos de Raspberry Pi, programación o sistemas Linux. Aunque un conocimiento de los conceptos básicos de la electrónica siempre es bienvenido y útil pero no te preocupes, ¡tampoco es necesario!

La parte electrónica se describe y explica de tal manera que pueda ser utilizada por personas sin experiencia en electrónica. ¡Vale la pena hacer estos ejercicios para conocer todas las capacidades de Raspberry Pi!

¡La Raspberry Pi tiene innumerables usos! Para aprovechar las posibilidades que ofrece la popular “frambuesa”, es necesario dar el primer paso y conocer los conceptos básicos para trabajar con este equipo. El curso comienza con los conceptos básicos completos de Raspberry Pi: desde la instalación del sistema (de varias formas), pasando por las configuraciones necesarias y el uso fluido de Linux, hasta los puertos GPIO universales. Pero sin más rollos, vamos a comenzar.



[Referencia B0069](#) – Raspberry Pi 4 Model B 1.5GHz

### 1.3. ¿Qué es la popular Raspberry?

Aunque tiene el tamaño de una tarjeta de crédito, la serie Raspberry Pi (en adelante, para abreviar **RPi**) son unos miniordenadores completamente funcionales, ya que tienen memoria dedicada, tarjeta gráfica y un procesador. La placa puede incluso ejecutar el sistema operativo Linux (teniendo una versión especialmente diseñada) y es fácil de instalar en la mayoría del software Linux, por lo tanto, permite la codificación en varios lenguajes (Python y C++ sobretodo). Una vez que el sistema operativo está instalado, es como trabajar en cualquier máquina Linux.

### 1.4. Historia

Las placas fueron desarrolladas por la empresa Raspberry Pi Foundation en 2012 para fomentar el aprendizaje básico de la informática en las escuelas, junto con los países en desarrollo. Aunque están diseñadas solo para la enseñanza, las placas se han vuelto más populares de lo previsto y se han utilizado en aplicaciones de alta gama como la robótica. A diferencia de otras plataformas, Raspberry Foundation mantiene una filosofía de hardware propio, teniendo el control sobre la creación y fabricación de estas placas.

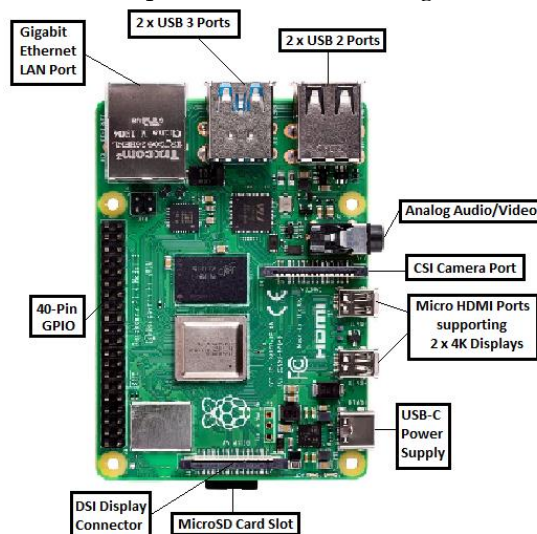
### 1.5. ¿Qué puedo hacer con la Raspberry?

Sí, es cierto que existen infinidad de proyectos que podemos llevar a cabo con una Raspberry, e introducimos en el mundo DIY y Maker, pero la mayoría de los usuarios acaban empleándola para 4 cosas:

- Como media center, o lo que es lo mismo, para convertir una televisión en una smart TV, con software LIBRELEC o OSMC.
- Para emular una videoconsola retro jugando a grandes clásicos con RetroPie instalado.
- Como ordenador con sistema Linux, a través de distribuciones como Ubuntu, Raspbian(Debian) o Pidora (Fedora).
- Domótica, con Windows 10 IOT Core, lo que permite hacer de nuestra casa un espacio un poco más inteligente con proyectos como estaciones meteorológicas o hubs inteligentes.

### 1.6. Características

Los conectores de esta placa son los que se muestran en la siguiente imagen:



## 2. Instalación del sistema, RPi como PC

La Raspberry Pi, como hemos dicho, necesita de un Sistema Operativo para trabajar, por lo que la plataforma en sí dispone de muchos SO diferentes y gratuitos basados en Linux.

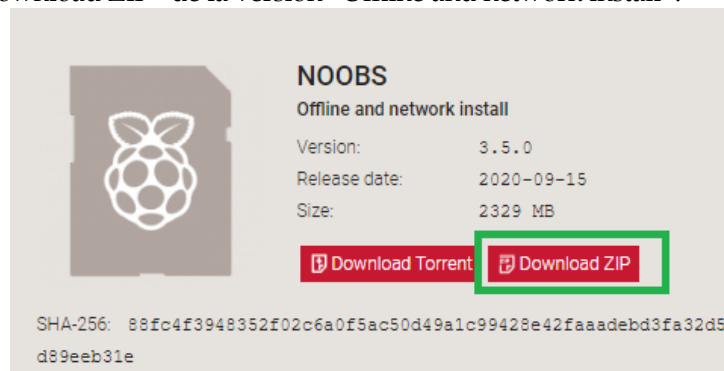
Como en el caso de una PC, el sistema operativo de la Raspberry Pi se puede instalar de varias formas. Hay tres métodos de instalación que vale la pena conocer: "tradicional", a través de UART y a través de la red. Cada uno de estos métodos tiene ventajas y desventajas. Saber cómo instalar el sistema de varias maneras le permite hacer frente a diferentes condiciones (independientemente de su hardware).

Nosotros comenzaremos con lo más simple, la instalación tradicional, cuando conectas al a RPi todos los periféricos: pantalla, teclado y ratón. Esta instalación se basa en NOOBS, el cual explicaremos a continuación. Sin embargo, existe la posibilidad dependiendo de los requerimientos de descargar desde un PC el SO y pasarlo a una tarjeta SD, o descargar una versión más pequeña llamada Lite.

### 2.1. Instalación NOOBS

En esta guía usaremos la Raspberry Pi 4 Model B (1GB RAM) en la cual, primero de todo, instalaremos **NOOBS** (New Out Of the Box Software). NOOBS está diseñado para hacer más lo más fácil posible la instalación y la puesta a punto del sistema operativo de tu Raspberry Pi. Es posible comprar tarjetas microSD con NOOBS pre-instalado, pero estas son más costosas y no vale la pena por algo que es sencillo de instalar. A continuación, vamos a guiar los pasos para instalarlo en una tarjeta microSD formateada.

Para descargar este software iremos a <https://www.raspberrypi.org/downloads/> y clicas en NOOBS y en "Download ZIP" de la versión "Offline and network install".



Mientras se descarga, puedes formatear una tarjeta microSD con capacidad suficiente.

Una vez hecho esto, instalaremos el software. Para ello simplemente tendremos que copiar todos los archivos del comprimido descargado en nuestra tarjeta SD.

Una vez extraigas la tarjeta, ya está lista para insertarla en la Raspberry. Cuando enchufes esta, se cargará NOOBS y te preguntará que elijas tu sistema operativo.

### 2.2. Conexiones y puesta a punto

Para conectar la placa RPi y empezar a usarla, necesitaremos:

- Adaptador de corriente USB Tipo-C: Un adaptador de corriente de 5V y hasta 3A con un conector Tipo-C. Raspberry dispone de adaptadores oficiales que son los recomendables.
- microSD con NOOBS
- Teclado y ratón USB: casi cualquier teclado y ratón cableado o inalámbrico nos servirá, excluyendo los que consuman demasiado como los "gaming" con muchas luces.
- Cable Micro-HDMI: el cual conectaremos a nuestro monitor o TV.

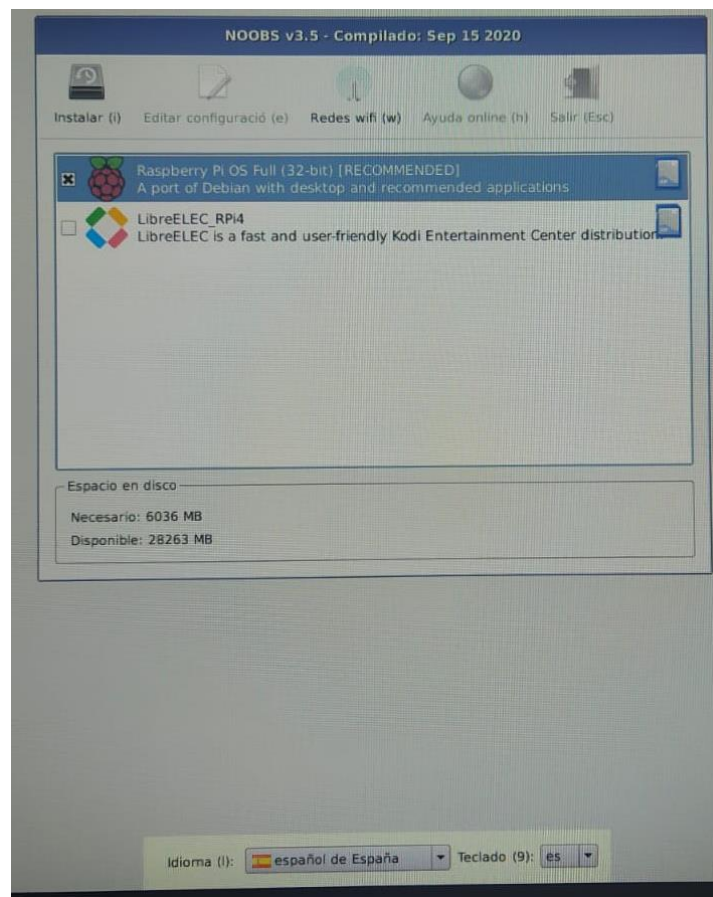
- Caja (opcional): es recomendable encapsular la Raspberry en una caja o funda de las disponibles en el mercado para evitar posibles daños.
- Cable de red Ethernet (opcional): opcionalmente puedes conectar tu Raspberry a la red mediante un cable ethernet con conector RJ-45.

Antes de comenzar a usar la Raspberry Pi, deberás configurar el software, en particular el sistema operativo. Cuando arrancamos nuestra Pi con NOOBS en la tarjeta SD, veremos una pantalla con el logo de Raspberry Pi y después, veremos una pestaña con el menú NOOBS.

Ahora deberás elegir con que sistema operativo se arrancará la placa, de los cuales hay dos incluidos con NOOBS:

- **Raspberry Pi OS Full**, una versión de Debian Linux específicamente pensada para Raspberry que antiguamente se llamaba Raspbian.
- **LibreELEC**, una versión del software Kodi Entertainment Centre.

Si la Raspberry está conectada a internet, desde el apartado “WiFi networks (w)” puedes incluso descargar otras versiones desde el menú. También puedes cambiar el idioma en la pestaña que sale abajo.



Para comenzar la instalación, debes seleccionar el cuadro, para que se marque con una cruz, del sistema elegido, en nuestro caso “**Raspberry Pi OS Full**”.

Ahora podrás clicar sobre el botón “Install (i)” para que comience la instalación, y después de un mensaje de advertencia que indica que cual dato será sobrescrito (excepto NOOBS), comenzará esta, la cual puede tardar entre 10-30 minutos dependiendo de la velocidad de la microSD.

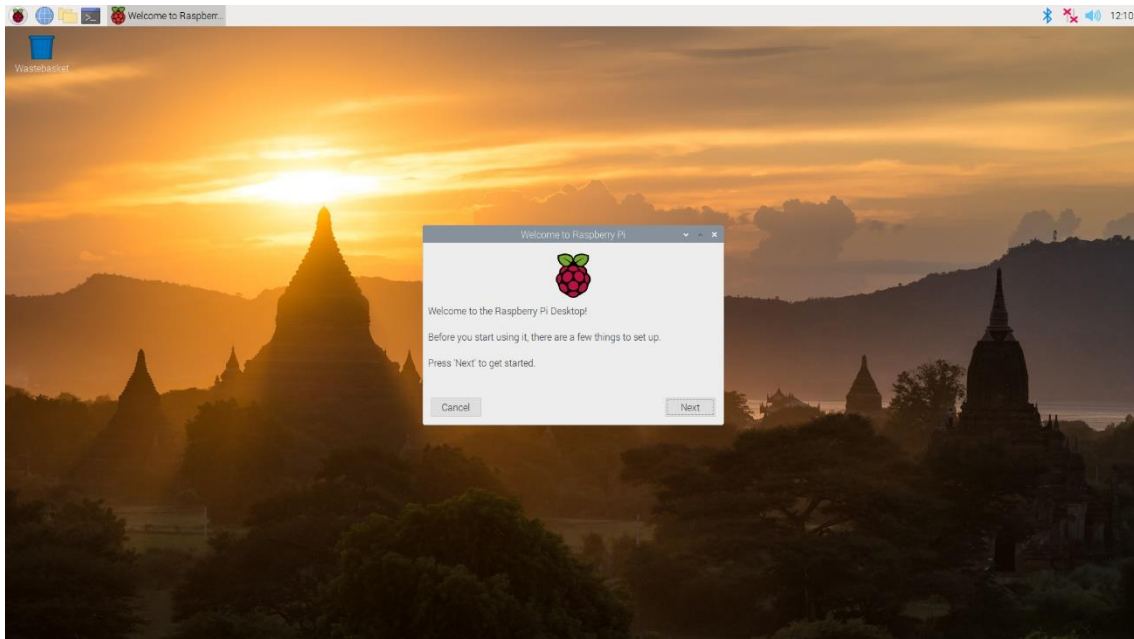
Cuando finalice la instalación, aparecerá un botón de “Ok” para reiniciar la Raspberry. La primera vez tardará 1 o 2 minutos para configurarse adecuadamente, pero el resto será más rápido.



Por último, verás una pantalla dándote la bienvenida al escritorio de tu Raspberry Pi.

### 2.3. Raspberry Pi OS, conocer el sistema operativo

Raspberry Pi OS (antiguo Raspbian) es un sistema operativo con una apariencia intuitiva y muy similar a los sistemas de Windows o macOS, por lo que no es complejo de aprender a usar. Primero de todo, se nos abrirá un menú de configuración inicial para configurar el país, idioma, contraseña de usuario, conexión WiFi y posibles actualizaciones. También nos dirá si la pantalla se ve correctamente o tiene márgenes, y el modo de solucionarlo. Si existieran actualizaciones, lo recomendable es instalarlas y reiniciar la Raspberry Pi.



Escritorio RPi OS

Welcome to Raspberry Pi

### Set Country

Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.

Country: Spain

Language: European Spanish

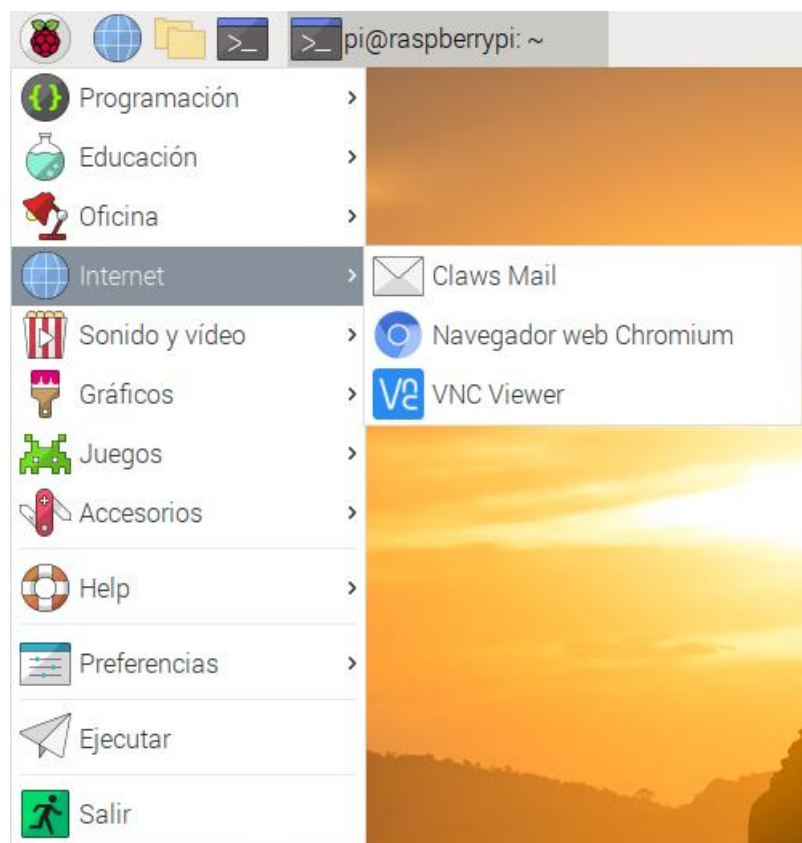
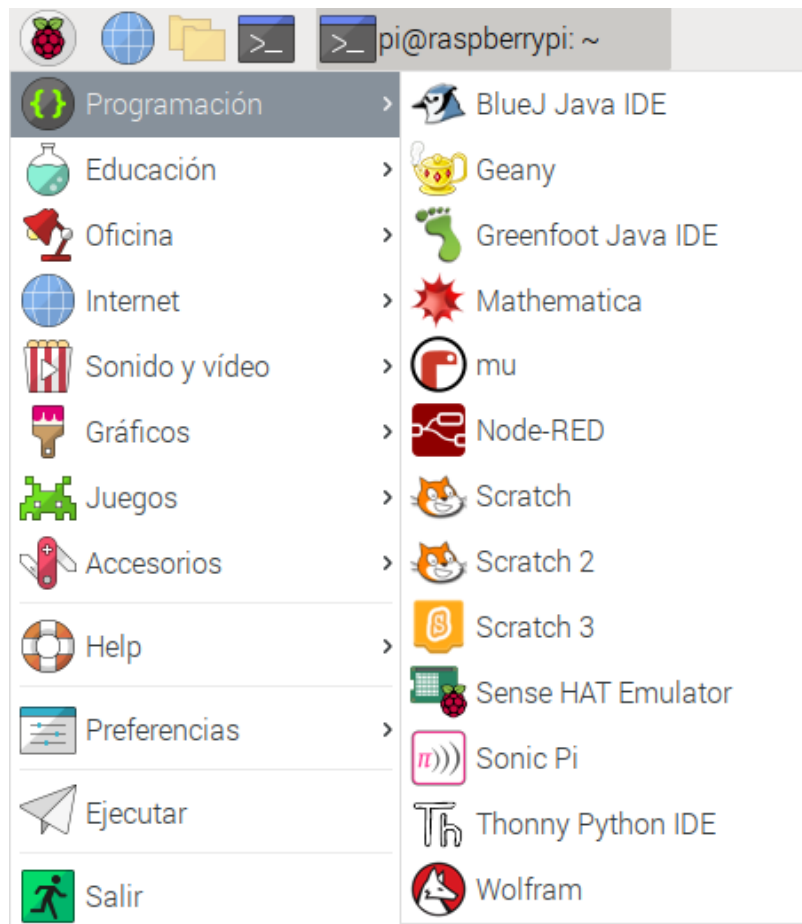
Timezone: Madrid

☐ Use English language ☐ Use US keyboard

Press 'Next' when you have made your selection.

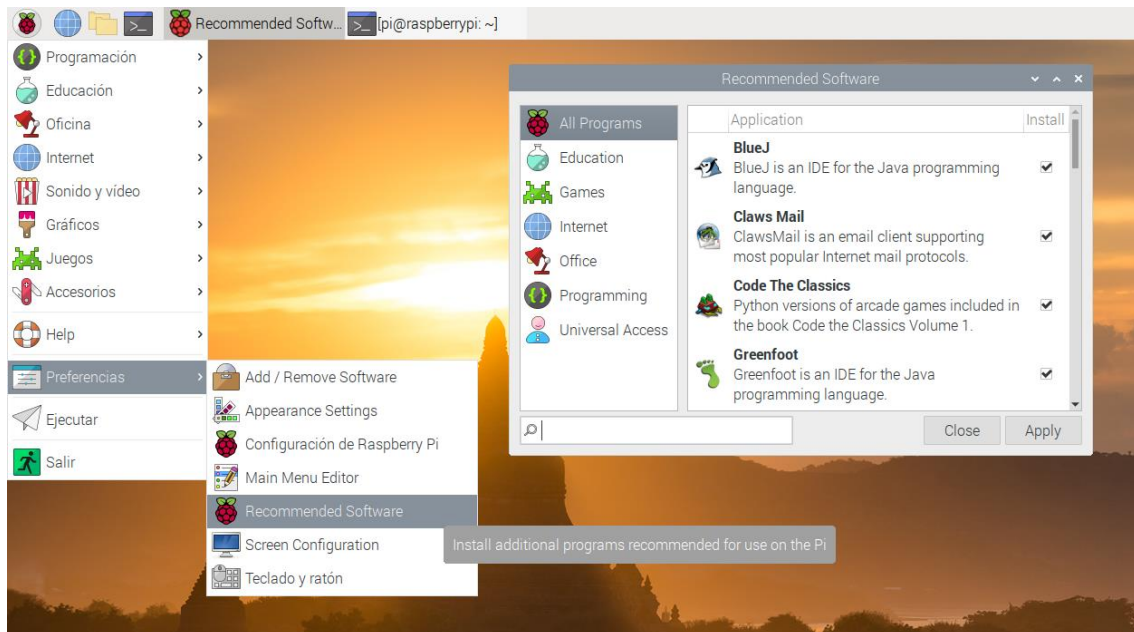
Back Next

Dentro del SO, en el escritorio, podemos explorar libremente sus menús y configuraciones. Explora su navegador Chromium, la suite LibreOffice, aplicaciones de Programación y demás aplicaciones que vienen por defecto.

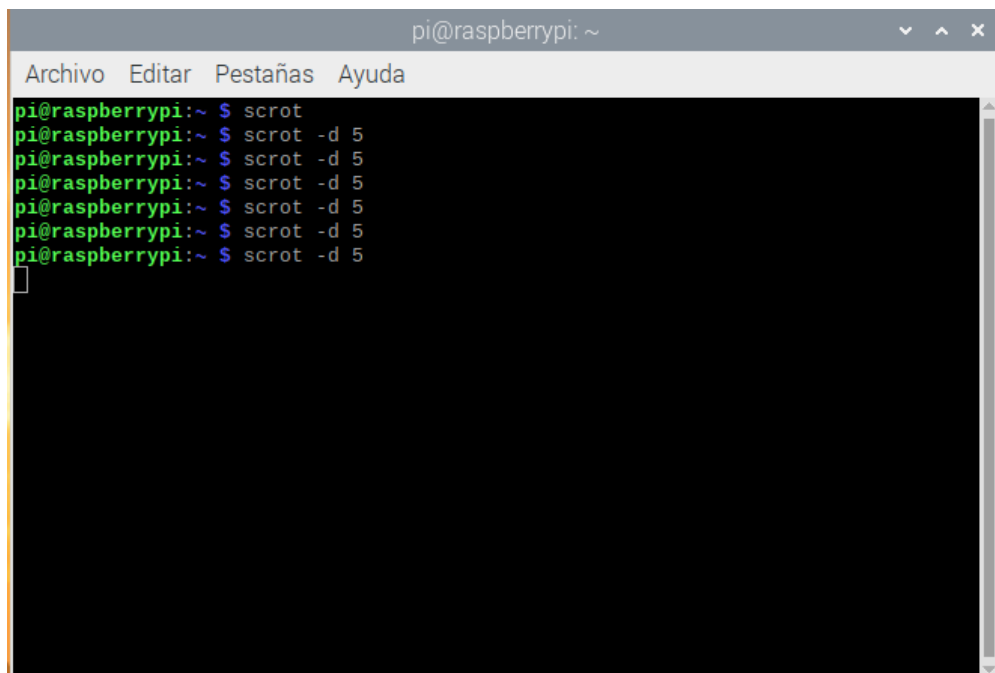




Para instalar otras aplicaciones recomendadas, puedes clicar en el menú raspberry, ir a “preferencias” y clicar en “Software recomendado”. Necesitarás tener internet conectado para que se carguen la lista de paquetes de software disponibles para instalar, desde IDEs para Java hasta el Minecraft. Las aplicaciones con un tick son las que ya están instaladas.



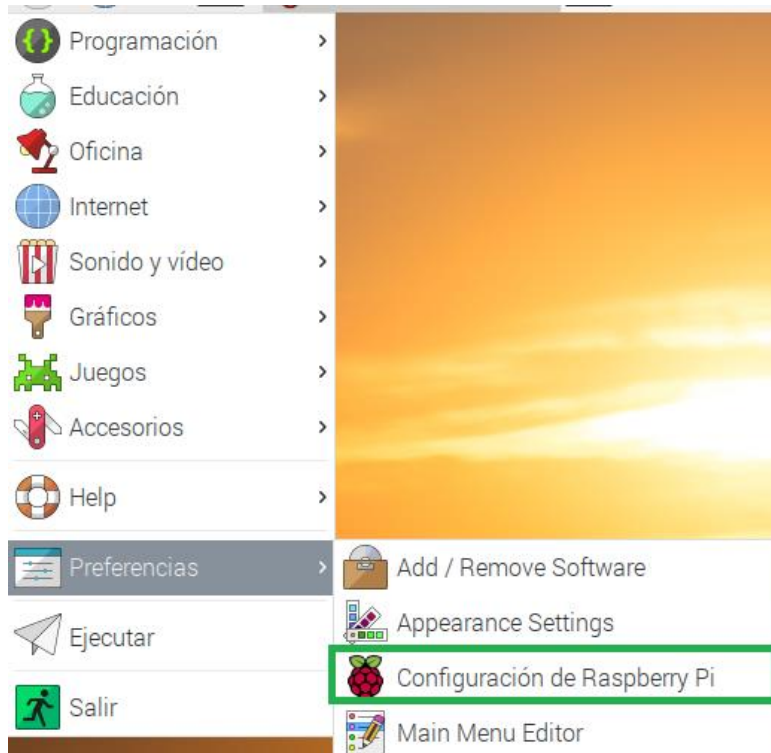
Como puedes ver, está abierta la consola de comandos minimizada, esto es porque la hemos usado para hacer las capturas de pantalla. Para hacer esto se usa el comando “scrot”. Al escribirlo hace una captura inmediatamente, y la guarda en la carpeta por defecto “/home/pi”. Una recomendación es usar un retardo añadiendo después “-d SEGUNDOS”. Por ejemplo, para hacer una captura de pantalla a los 5 segs, usamos:” scrot -d 5” como se puede ver en el ejemplo.



En otro punto, hablaremos de la consola de comandos más en detalle y de los conceptos básicos de Linux.

### 2.3.1. Configuración mediante el menú propio

Ahora hablemos de como cambiar las configuraciones, si te has equivocado al principio, quieres cambiar la contraseña o en un futuro quieres modificar alguna configuración, como en cualquier interfaz gráfica de usuario (GUI), tenemos un menú de configuración.



Accederemos a una ventana que tiene 5 pestañas distintas:

- Sistema: donde podemos cambiar la clave o el modo de inicio entre otras cosas.



- Display: donde podremos modificar las opciones sobre la pantalla.

Configuración de Raspberry Pi

Sistema	Display	Interfaces	Rendimiento	Localización
Overscan:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		
Pixel Doubling:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		
Composite Video:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		
Screen Blanking:	<input checked="" type="radio"/> Activo	<input type="radio"/> Desactivado		

Cancelar Aceptar

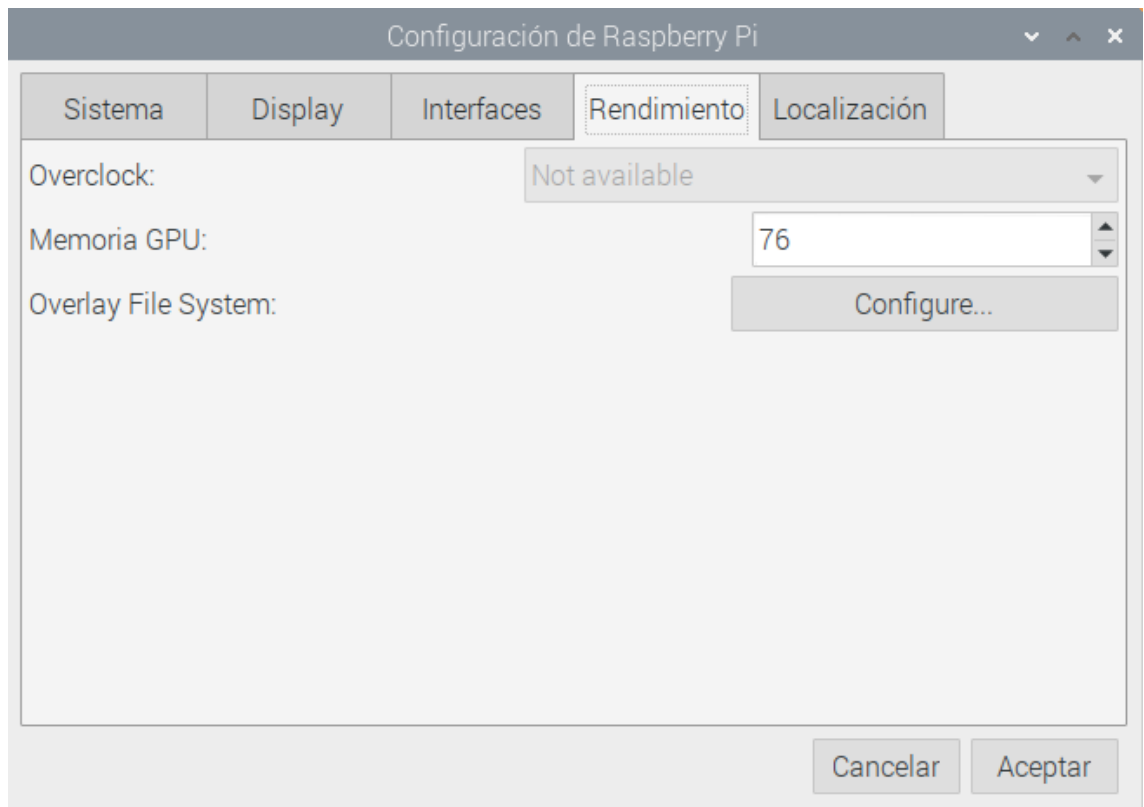
- Interfaces: donde podemos activar o desactivar las interfaces que vayamos a usar. Esto es importante recordarlo, ya que si no está activo no podremos usarlo.

Configuración de Raspberry Pi

Sistema	Display	Interfaces	Rendimiento	Localización
Cámara:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		
SSH:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		
VNC:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		
SPI:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		
I2C:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		
Serial Port:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		
Serial Console:	<input checked="" type="radio"/> Activo	<input type="radio"/> Desactivado		
1-Wire:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		
Remote GPIO:	<input type="radio"/> Activo	<input checked="" type="radio"/> Desactivado		

Cancelar Aceptar

- Rendimiento: donde podemos modificar las características de rendimiento de nuestra RPi.



- Localización: donde ajustaremos los parámetros en función de donde se esté usando la RPi.

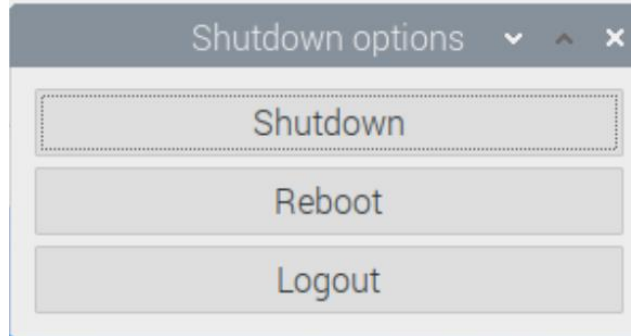


### 2.3.2. Apagado

Por último, hay que comentar que como en cualquier ordenador, la Raspberry Pi debe apagarse debidamente. Para ello clicas sobre el icono de raspberry en la izquierda y luego sobre Salir



Se nos abrirá una pestaña donde le tendremos que dar a Shutdown.



Nota: es posible también apagarla desde el terminal de comandos usando: `sudo shutdown -h now`  
Con este comando también es posible establecer una hora de apagado, por ejemplo: `sudo shutdown -h 22:00`

Recuerda SIEMPRE, apagar la RPi correctamente para evitar fallos y problemas.

### 2.3.3. Conclusión

Hemos visto como instalar un SO tradicionalmente mediante NOOBS, pero como hemos dicho, hay más opciones que puedes ver en los siguientes puntos, como por ejemplo la instalación mediante UART, útil en el caso que no se disponga de pantalla para la RPi.

## 3. Programación

Existen una gran cantidad de lenguajes de programación con los que se puede crear una infinidad de cosas, como juegos, bases de datos, programas o controlar dispositivos físicos.

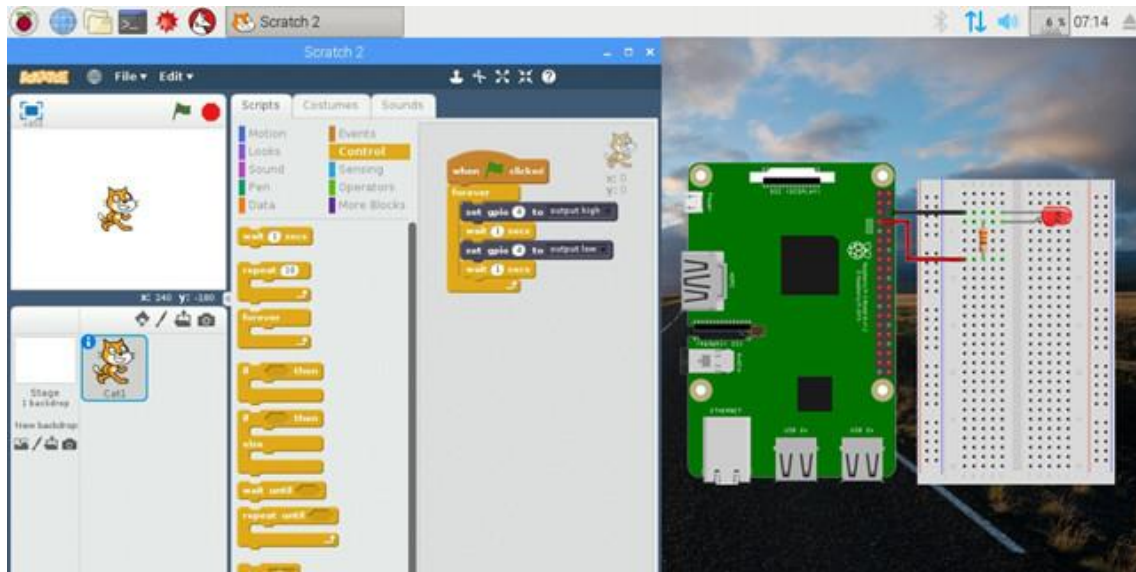
Para empezar a aprender programación, existen también muchos tutoriales y lenguajes más sencillos, uno de los más sencillos y visuales es **Scratch**, por otra parte, uno de los lenguajes más usados y potentes es **Python**, ambos los veremos a continuación con un poco más de detalle.

Además de estos dos lenguajes, si no te convencen, siempre puedes instalar un IDE (entorno de desarrollo) distinto para trabajar con otros lenguajes como Java, C/C++, HTML 5, etc.

### 3.1. Scratch

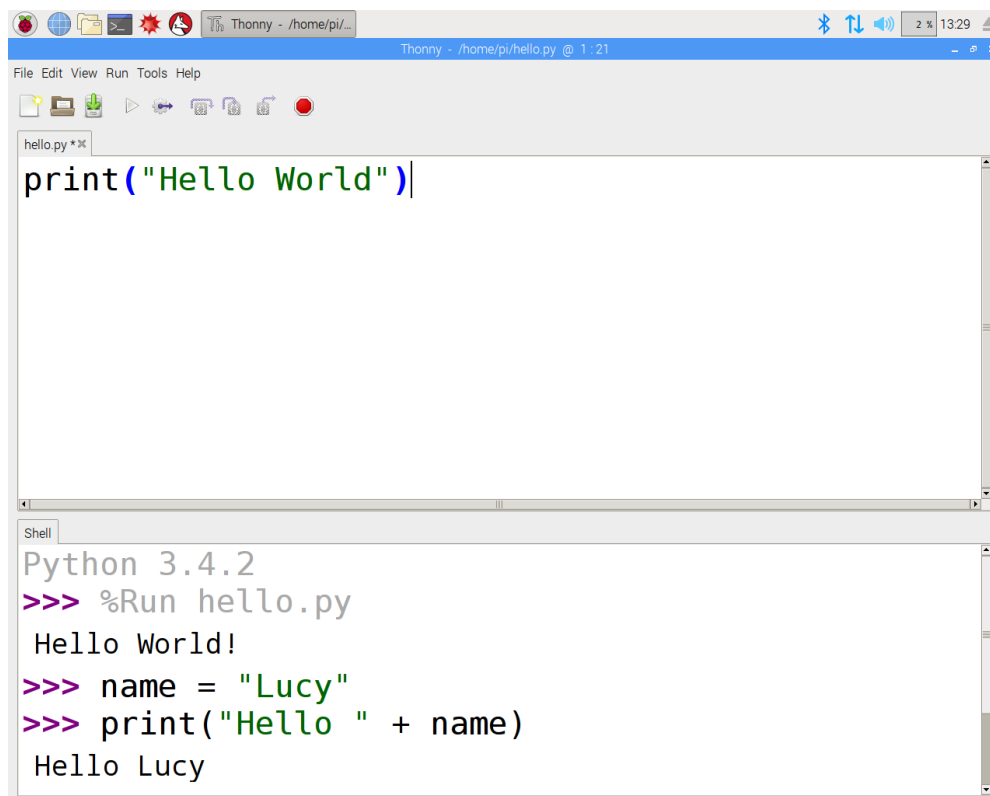
Con este lenguaje es muy fácil iniciarse en el mundo, tanto para niños como para cualquier persona sin noción previa.

Para empezar con **Scratch**, un lenguaje de programación visual desarrollado por el MIT, abre Scratch 2 desde la sección de Programación dentro del Raspberry Pi OS. No te preocupes, a pesar de su apariencia sencilla y amigable, es un potente entorno de programación totalmente funcional para desarrollar desde pequeños juegos o imágenes, hasta complejos proyectos de robótica. Este lenguaje se basa en programación paso a paso mediante bloques, los cuales son parte de código pre-escrito dentro de piezas clasificadas por colores. Hay muchos libros y tutoriales que enseñan este lenguaje.



### 3.2. Python

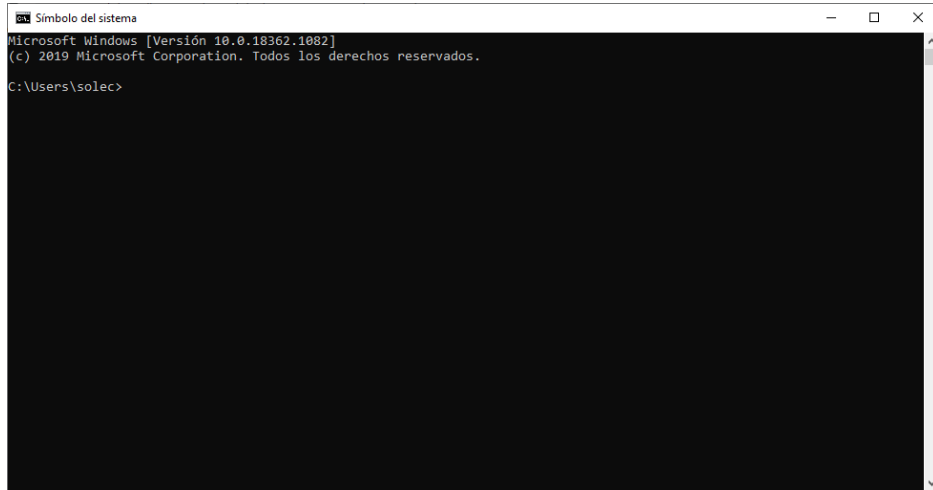
Por otra parte, **Python** está basado en programación mediante escribir instrucciones usando un lenguaje propio y un formato específico. Raspbian tiene un entorno de desarrollo (IDE) para programar con Python, como el programa Scratch 2, llamado Thonny Python IDE. Este tiene dos interfaces: "Normal Mode", y "Simple Mode", el cual es mejor para principiantes. Existen también muchos libros y tutoriales que ayudan a iniciarse.





## 4. Consola de comandos y conceptos básicos de Linux

El apartado está dedicado a personas que están comenzando su aventura con Raspberry y Linux, por lo que algunos términos pueden ser nuevos. La mayoría de los usuarios de Windows se han encontrado con la aplicación **Símbolo del sistema**. Esta herramienta le permite ingresar y ejecutar comandos en modo texto:



El lugar donde ingresamos comandos y ejecutamos scripts en Linux se llama **Shell**, una interfaz que te da acceso a los servicios del sistema operativo. La mayoría de las distribuciones de Linux utilizan una interfaz gráfica de usuario (GUI) como shell, principalmente para proporcionar facilidad de uso a sus usuarios.

Dicho esto, se recomienda usar una interfaz de línea de comandos (CLI) porque es más potente y efectiva. Las tareas que requieren un proceso de varios pasos a través de la GUI se pueden realizar en cuestión de segundos escribiendo comandos en la CLI. Aunque los pasos pueden diferir según la distribución que estás utilizando, generalmente puedes encontrar la línea de comando en la sección Utilidades.

Es frecuente denominar al CLI como consola de comandos, emulador de terminal, o shell. Aunque son conceptos distintos, en la práctica es frecuente que se usen (incorrectamente) como sinónimos ya que están muy relacionados.

Piensa que usando una línea de comandos, tienes que traducir todas las acciones que haces con la interfaz gráfica y el ratón, a línea de texto mediante comandos. Desde entrar a una carpeta, a copiar y pegar un archivo, todo tiene una traducción a comandos usando una especie de intérprete.

En Linux existe más de un intérprete de líneas de comandos. Hay varios, y algunos ejemplos importantes son Bash, Dash, KornShell (ksh), Tcsh, Fish o Z Shell (zsh), entre otros, cada uno con sus ventajas y desventajas como siempre.

**GNU Bash** es el CLI implementado mayoritariamente en las distribuciones de Linux. Las distribuciones Linux que emplean Bash como CLI por defecto son Debian, openSUSE, Ubuntu, Fedora, Slackware, Mandriva.

El Raspberry Pi OS, que se conocía por Raspbian, está basado en Debian, como su nombre hace ver. Por lo que Bash es el CLI incluido por defecto en Raspberry Pi.

## ¿Qué es Bash?

Bash, Bourne-Again SHell, fue lanzado en 1989. Es el predecesor de SH, The Bourne Shell, que es uno de los CLI originales de Unix. Los scripts de Bash son compatibles con SH, pero Bash añadió funciones adicionales como autocompletado o histórico de comandos, funciones muy útiles.

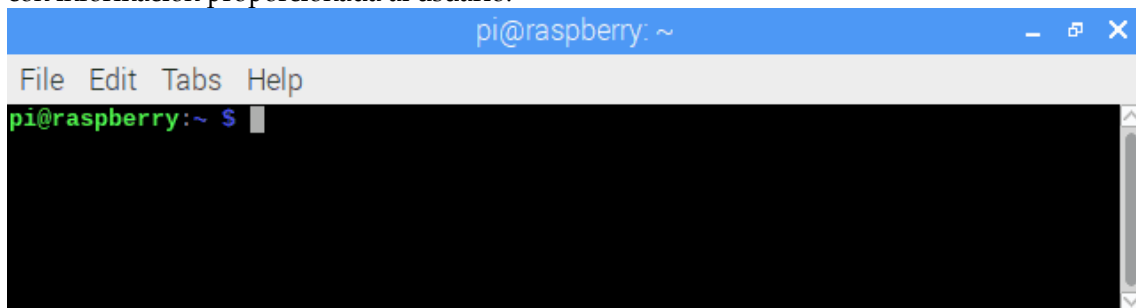
### 4.1. Comandos Bash

Entonces, si estás considerando usar Linux, aprender líneas de comando básicas será de gran ayuda.

Lo primero, para abrir el CLI en Raspberry, como hemos visto, existe directamente desde la barra de tareas del escritorio un botón para abrirlo.



Al inicio de cada línea en la que el usuario puede introducir un comando, veremos una cabecera con información proporcionada al usuario.



Donde:

- **pi**: indica el usuario conectado a la terminal, por defecto se llama pi
- **@raspberrypi**: indica el nombre de la máquina a la cual estamos conectados
- **~**: indica la ruta en la cual nos encontramos, en este caso ruta de inicio predeterminada
- **\$**: indicador para comenzar a escribir comandos (será **#** si estamos loggeados como usuario root, el cual tiene opciones avanzadas)

Una vez dicho esto, podemos ver como el recuadro blanco parpadea a la espera de que le introduzcamos una instrucción. Para ello solo debes escribirla y pulsar enter. Podemos probar a abrir el navegador escribiendo:

*Chromium-browser*

También puedes hacer que la aplicación que quieras abrir lance algo, por ejemplo, que el navegador abra nuestra página, para ello escribe:

*Chromium-browser www.solectroshop.com*

Algunas instrucciones o atajos importantes que debes conocer son:

- **control + C** : se usa cuando algún comando ejecuta un proceso, y queremos cancelarlo, podemos pulsar el atajo de teclado
- **clear** : se usa para limpiar la pantalla, despejando de todos los comandos escritos con anterioridad
- **tecla tabulación**: es la función de autocompletado, si empiezas a escribir un comando y pulsas dos veces la tecla de tabulación, sugiere la opción que empieza por lo que hemos escrito. En caso de haber más de una ocurrencia, se recorren cíclicamente las opciones disponibles. Por ejemplo, si escribimos *ch* y pulsamos tabulación, nos aparecerá un listado con las opciones:

```
pi@raspberrypi:~$ ch
chage      chardetect3  chfn      chown      chrt
chardet    chattr      chgpasswd chpasswd   chsh
chardet3   chcon      chgrp     chromium   chvt
chardetect chcpu      chmod     chroot
pi@raspberrypi:~$ chrom
```

- **tecla arriba:** es la función del historial de comandos. Al pulsar la tecla hacia arriba, nos aparecerá escrito el último comando introducido, listo para modificar o volver a introducir. Si sigues pulsando hacia arriba ves el penúltimo y así. Del mismo modo, puedes volver a bajar hacia el último con la tecla hacia abajo.
- **tecla ESC:** volver atrás, vuelves al menú anterior
- Comandos de **ayuda:** en Bash existen varios comandos para ofrecerte ayuda sobre diversas funciones.
  - Mostrar un resumen de la función del comando xxxxx:  
*whatis xxxxx*
  - Mostrar la localización más probable para el programa xxxxxx:  
*whereis xxxxx*
  - Listar las páginas de manual que tratan acerca del comando xxxxx:  
*apropos xxxxx*
  - Mostrar el manual de uso o configuración del programa xxxxx:  
*man xxxxx*
  - Mostrar las páginas de manual que contengan la palabra xxxxx:  
*man -k xxxxx*

Existen multitud de comandos de acciones. Puedes probar los siguientes comandos básicos en la línea de comandos o terminal:

- Navegar por directorios con comandos como **ls** , **cd** y **pwd**
- Crea, elimina y cambia el nombre de los directorios con los comandos **mkdir** , **rm** , **mv** y **cp**
- Crea un archivo de texto con un editor de línea de comandos como **Vi** , **Vim** , **Emacs** o **Nano**
- Prueba otros comandos útiles como: **chmod** , **chown** , **w** , **cat** , **more** , **less** , **tail** , **free** , **df** , **ps** , **uname** y **kill**
- Para realizar tareas que requieren permisos administrativos o raíz se usa el comando **sudo**, abreviatura de «SuperUser Do» (SuperUsuario hace)

## 4.2. Actualización general

Unos comandos importantes que debemos ejecutar para actualizar completamente nuestra RPi son:

```
sudo apt update
sudo apt full-upgrade
```

Al ejecutar el segundo, si existen actualizaciones nos dirá que para continuar pulsemos la tecla “s” o para no continuar con la tecla “n”.

Las actualizaciones llevan su tiempo, ocuparán cierta cantidad de memoria y al finalizar se debe reiniciar, por ejemplo, con el comando: *sudo reboot*.

```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda

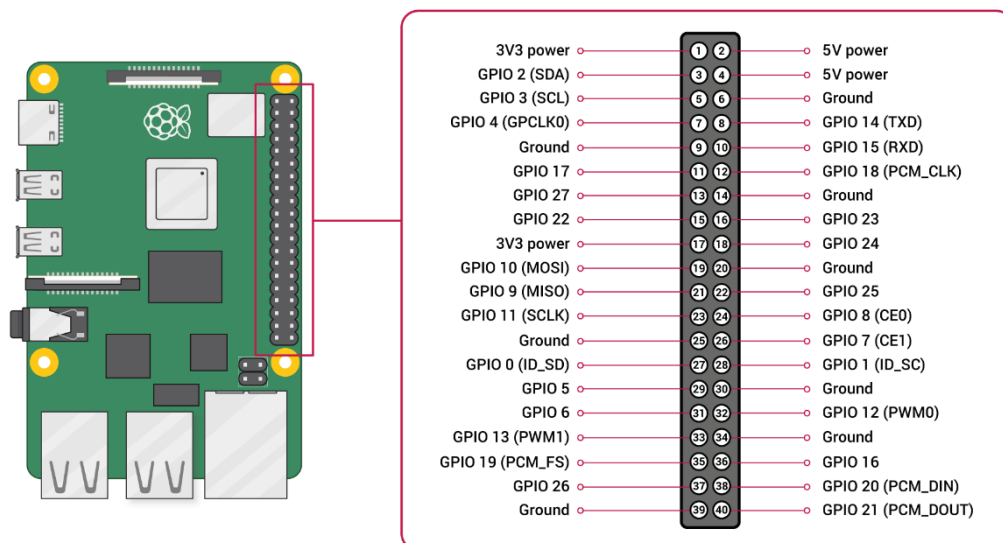
pi@raspberrypi:~$ sudo apt update
Des:1 http://archive.raspberrypi.org/debian buster InRelease [32,6 kB]
Des:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Des:3 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0
MB]
Des:4 http://archive.raspberrypi.org/debian buster/main armhf Packages [331 kB]
Descargados 13,4 MB en 10s (1.312 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 91 paquetes. Ejecute «apt list --upgradable» para verlos.
pi@raspberrypi:~$ sudo apt full-upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
libxv2-14 libgfortran3 libgmime-2.6-0 libncurses5 libssl1.0.2 uuid-dev
Utilice «sudo apt autoremove» para eliminarlos.
Se actualizarán los siguientes paquetes:
agnostics alacarte base-files bind9-host bluez-firmware firmware-atheros
firmware-brcm80211 firmware-libertas firmware-misc-nonfree firmware-realtek
libbind9-161 libcommons-compress-java libdns-export1104 libdns1104
libfm-data libfm-extra4 libfm-gtk-data libfm-gtk4 libfm-modules libfm4
libgs9 libgs9-common libgssdp-1.0-3 libgupnp-1.0-4 libhttpclient-java
libisc-export1100 libisc1100 libisccc161 libiscfg163 liblwres161 libpq5
libqt5concurrent5 libqt5core5a libqt5dbus5 libqt5gui5 libqt5network5
libqt5opengl5 libqt5printsupport5 libqt5sql5 libqt5sql5-sqlite libqt5test5
libqt5widgets5 libqt5xml5 libraspberrypi-bin libraspberrypi-dev
libraspberrypi-doc libraspberrypi0 libvlc-bin libvlc5 libvlccore9 libx11-6
libx11-data libx11-dev libx11-xcb1 libxmq5 lxpanel lxpanel-data
lxplug-bluetooth lxplug-cputemp lxplug-ejecter lxplug-network lxplug-ptbatt
npm pcmanfm pipanel ppprompt python3-zmq qt5-gtk-platformtheme
raspberrypi-bootloader raspberrypi-kernel raspi-config rc-gui
rpi-chromium-mods rpi-eepprom ruby-activesupport vlc vlc-bin vlc-data
vlc-l10n vlc-plugin-base vlc-plugin-notify vlc-plugin-qt vlc-plugin-samba
vlc-plugin-skins2 vlc-plugin-video-output vlc-plugin-video-splitter
vlc-plugin-visualization wolfram-engine wolframscript xserver-common
xserver-xorg-core
91 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 797 MB de archivos.
Se utilizarán 1.065 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]

```

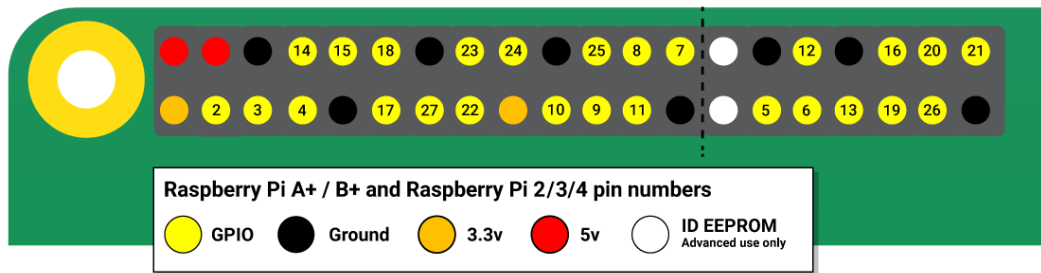
Para saber más sobre los comandos, entra en [nuestro artículo](#) sobre comandos en Linux para Raspberry.

## 5. Pines GPIO

En cuanto a lo que es directamente la placa Raspberry Pi, sabemos que está preparada para trabajar con programación física de componentes electrónicos mediante sus pines GPIO (general-purpose input/output). Estos pines de propósito general de entrada/salida son algunos de los que se pueden ver en dos filas de 20 pines. En la siguiente imagen podemos ver más claramente a que corresponde cada pin.



Cualquiera de los pines GPIO puede designarse (en el software) como un pin de entrada o salida y utilizarse para una amplia gama de propósitos, como conectar un LED o un sensor.



**Nota:** la numeración de los pines GPIO no está en orden numérico; Los pines GPIO 0 y 1 están presentes en la placa (pines físicos 27 y 28) pero están reservados para uso avanzado (ver más abajo).

### Voltajes

Dos pines de 5V y dos pines de 3V3 (3,3V) están presentes en la placa, así como varios pines de tierra (GND: 0V), que no son configurables. Los pines restantes son todos pines 3V3 de uso general, lo que significa que las salidas están configuradas en 3V3 y las entradas son tolerantes a 3V3.

### Salidas

Un pin GPIO designado como pin de salida se puede configurar en estado alto (3V3) o bajo (0V).

### Entradas

Un pin GPIO designado como pin de entrada se puede leer como alto (3V3) o bajo (0V). Esto se hace más fácil con el uso de resistencias internas pull-up o pull-down. Los pines GPIO2 y GPIO3 tienen resistencias pull-up fijas, pero para otros pines esto se puede configurar en el software.

### Más

Además de los dispositivos de entrada y salida simples, los pines GPIO se pueden usar con una variedad de funciones alternativas, algunas están disponibles en todos los pines, otras en pines específicos.

- PWM: Pines que admiten la Modulación de ancho de pulso, usada para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.
  - Software PWM disponible en todos los pines
  - Hardware PWM disponible en GPIO12, GPIO13, GPIO18, GPIO19
- SPI: El Bus SPI (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.
  - SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
  - SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)
- I2C: Circuito Integrado Interno (I<sup>2</sup>C, del inglés Inter-Integrated Circuit), se usa internamente para la comunicación entre diferentes partes de un circuito, por ejemplo, entre un controlador y circuitos periféricos integrados.
  - Datos (GPIO2); Reloj (GPIO3)

- Datos EEPROM: (GPIO0); Reloj EEPROM (GPIO1)
- Serie: Puertos para comunicación UART, en inglés de Universal Asynchronous Receiver-Transmitter, mediante pin de transmisión TX y pin de recepción RX.
  - TX (GPIO14); RX (GPIO15)

Existen 2 formas de numerar los pines de la Raspberry Pi, en modo GPIO o en modo BCM. Esto es importante a la hora de programar. La diferencia entre estas notaciones generalmente lo notamos cuando estamos controlando los GPIO con python u otro lenguaje y tenemos que ingresar en qué modo usaremos los gpio en GPIO.BOARD o GPIO.BCM.

La opción GPIO.BOARD especifica que se está refiriendo a los pines por su número, es decir los números impresos en nuestra Raspberry Pi (por ejemplo P1), en la imagen siguiente te mostramos el pin 1 y el pin 2, para seguir numerando sería de izquierda a derecha, siendo el que está debajo del pin 1 el pin 3, toda esa fila serían los pines impares y la opuesta los pines pares.

La opción GPIO.BCM se refiere a los pines por su número de "Broadcom SOC channel", estos no son correlativos como en el modo BOARD, en la imagen siguiente se muestran tanto los pines en BOARD como en BCM de las distintas versiones de Raspberry Pi, siendo los BCM los que su nombre comienza con GPIO y los centrales los pines BOARD. Los BCM serían por así decirlo como la numeración del procesador y los BOARD del conector.

Pin No.		
3.3V	1	2 5V
GPIO2	3	4 5V
GPIO3	5	6 GND
GPIO4	7	8 GPIO14
GND	9	10 GPIO15
GPIO17	11	12 GPIO18
GPIO27	13	14 GND
GPIO22	15	16 GPIO23
3.3V	17	18 GPIO24
GPIO10	19	20 GND
GPIO9	21	22 GPIO25
GPIO11	23	24 GPIO8
GND	25	26 GPIO7
DNC	27	28 DNC
GPIO5	29	30 GND
GPIO6	31	32 GPIO12
GPIO13	33	34 GND
GPIO19	35	36 GPIO16
GPIO26	37	38 GPIO20
GND	39	40 GPIO21

Para controlar GPIO con Python, primero debes importar una librería de código escrito previamente. El más común y difundido es el Rpi.GPIO, utilizado para crear miles de proyectos desde los primeros días de la RPi.

Otra librería interesante es [WiringPi](http://WiringPi), que hace que los pins sean más "portables" de unos códigos a otros, más universales.

Para instalar la última versión ponemos las siguientes instrucciones en la consola:

```
cd /tmp
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
```

Para ver estos pines en la Raspberry, podemos abrir la consola de comandos, y ejecutar el comando: **gpio readall**



Como su nombre indica, leerá todos lo que pertenezca a gpio, y nos mostrará la siguiente tabla informativa, donde se pueden ver cada pin y características.

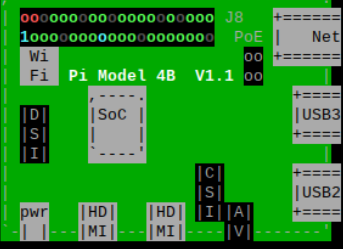
```
pi@raspberrypi:/tmp $ cd
pi@raspberrypi:~ $ gpio readall
```

-----Pi 4B-----											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
2	8	3.3v			1	2		5v			
3	9	SDA.1	IN	1	3	4		5v			
4	7	SCL.1	IN	1	5	6		0v			
		GPIO. 7	IN	1	7	8	1	IN	TxD	15	
		0v			9	10	1	IN	RxD	16	
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	
27	2	GPIO. 2	IN	0	13	14		0v		18	
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	
		3.3v			17	18	0	IN	GPIO. 5	5	
10	12	MOSI	IN	0	19	20		0v		24	
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	
		0v			25	26	1	IN	CE1	11	
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	
5	21	GPIO.21	IN	1	29	30		0v		1	
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	
13	23	GPIO.23	IN	0	33	34		0v		12	
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	
		0v			39	40	0	IN	GPIO.29	29	
										21	

```
pi@raspberrypi:~ $
```

Otro comando que nos puede interesar es: *pinout*. Con este nos mostrará información de manera más visual:

```
Archivo  Editar  Pestañas  Ayuda
pi@raspberrypi:~ $ pinout
```



```
Revision      : a03111
SoC           : BCM2711
RAM           : 1024Mb
Storage       : MicroSD
USB ports     : 4 (excluding power)
Ethernet ports : 1
Wi-fi        : True
Bluetooth    : True
Camera ports (CSI) : 1
Display ports (DSI) : 1

J8:
3V3 (1) (2) 5V
GPIO2 (3) (4) 5V
GPIO3 (5) (6) GND
GPIO4 (7) (8) GPIO14
GND (9) (10) GPIO15
GPIO17 (11) (12) GPIO18
GPIO27 (13) (14) GND
GPIO22 (15) (16) GPIO23
3V3 (17) (18) GPIO24
GPIO10 (19) (20) GND
GPIO9 (21) (22) GPIO25
GPIO11 (23) (24) GPIO8
GND (25) (26) GPIO7
GPIO0 (27) (28) GPIO1
GPIO5 (29) (30) GND
GPIO6 (31) (32) GPIO12
GPIO13 (33) (34) GND
GPIO19 (35) (36) GPIO16
GPIO26 (37) (38) GPIO20
GND (39) (40) GPIO21

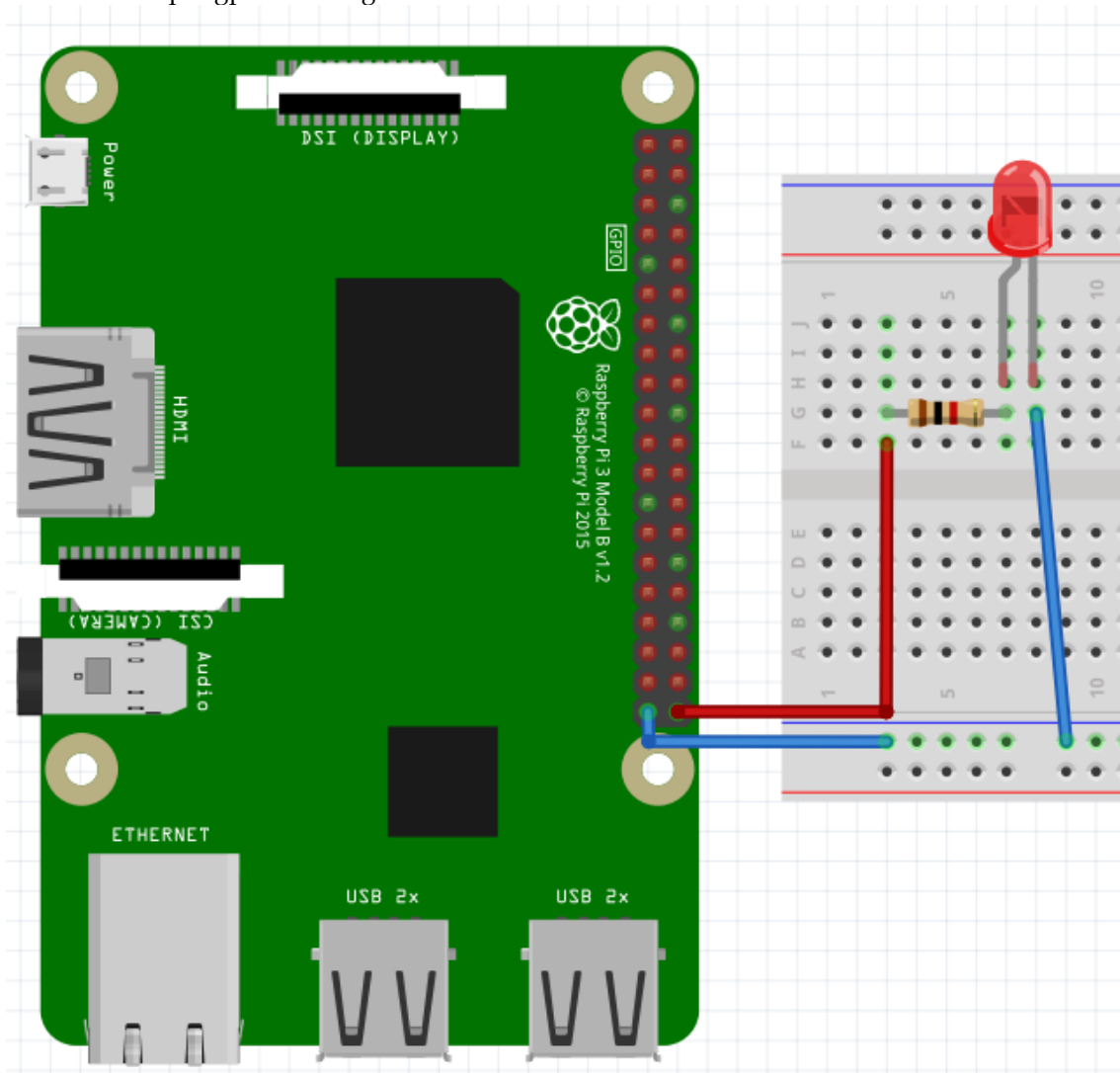
For further information, please refer to https://pinout.xyz/
pi@raspberrypi:~ $
```

## 5.1. Programación GPIO

Ya sabemos lo que son estos pines, pero ahora vamos a explicar cómo usarlos y programar dispositivos. Se puede usar la misma consola de comandos para cambiar la configuración de los pines, por ejemplo, que el pin 21 saque voltaje alto para encender un LED que tenga conectado.

### 5.1.1. Mediante consola

Esto se puede “programar” de varios modos y el más sencillo es usar simplemente comandos sobre la consola. Primero de todo conectaremos el LED a la RPi. Recuerda que siempre tiene que tener una resistencia en serie conectada, en nuestro caso usaremos una de  $1k\ \Omega$  y un LED rojo conectados al pin gpio21 del siguiente modo:



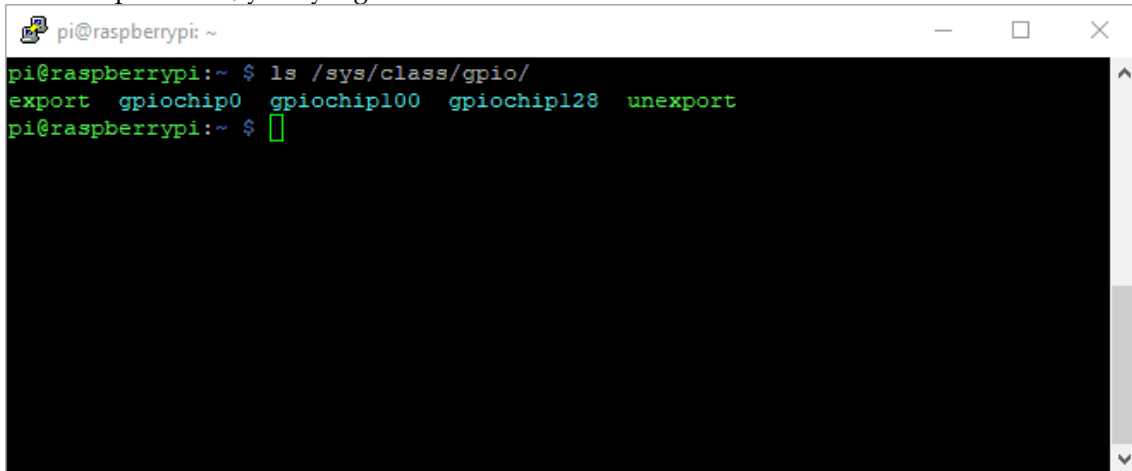
El pin 40 va a la resistencia, esta va al ánodo del LED, el cátodo del LED va a GND, en el pin 39.

A continuación, arrancamos la RPi y abrimos el terminal de comandos y ejecutamos el comando:

```
ls /sys/class/gpio/
```

El directorio `/sys` es uno de los muchos directorios virtuales de nuestro sistema. No lo encontraremos en la tarjeta SD. El sistema sólo “finge” que tal directorio existe, de hecho, todas las referencias a archivos en ese directorio son manejadas por el kernel (núcleo del SO en Linux).

El directorio `/sys` permite acceder a información sobre los controladores del sistema Linux. En este caso, nos fijamos en los controladores de clase GPIO, es decir, los puertos de entrada y salida. Como se puede ver, ya hay algunos archivos en este directorio:

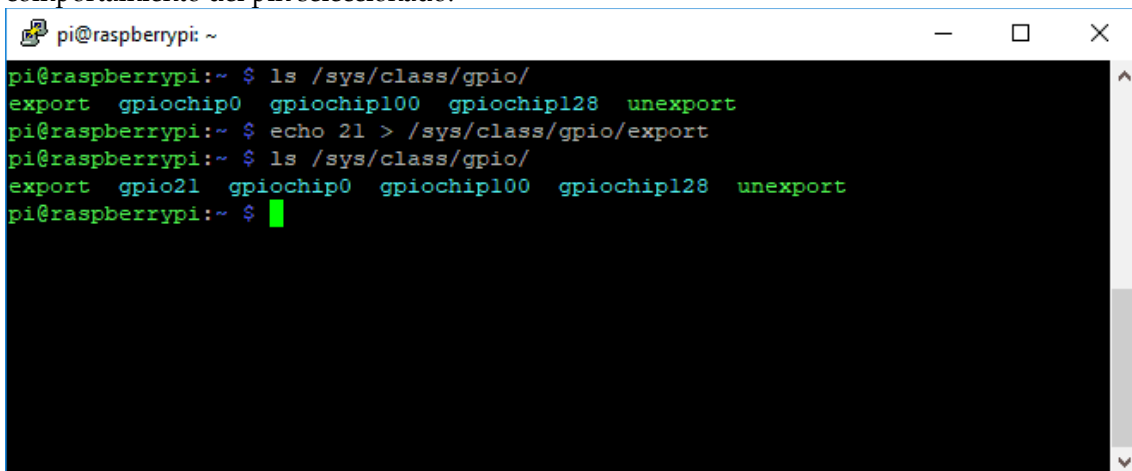


```
pi@raspberrypi:~ $ ls /sys/class/gpio/
export  gpiochip0  gpiochip100  gpiochip128  unexport
pi@raspberrypi:~ $
```

De forma predeterminada, el acceso a los pines no es posible, por lo que debemos habilitarlo. Usaremos un archivo llamado **export** para este propósito. Escribir en este archivo hará que el kernel nos proporcione el pinout apropiado del procesador. Queremos tener acceso al pin GPIO21, así que escribimos el número 21 en el archivo `/sys/class/gpio/export` escribiendo:

```
echo 21 > /sys/class/gpio/export
```

Ahora podemos verificar el contenido del directorio `/sys/class/gpio` nuevamente; encontraremos un nuevo subdirectorio llamado `gpio21`. Su contenido te permitirá controlar el comportamiento del pin seleccionado.



```
pi@raspberrypi:~ $ ls /sys/class/gpio/
export  gpiochip0  gpiochip100  gpiochip128  unexport
pi@raspberrypi:~ $ echo 21 > /sys/class/gpio/export
pi@raspberrypi:~ $ ls /sys/class/gpio/
export  gpio21  gpiochip0  gpiochip100  gpiochip128  unexport
pi@raspberrypi:~ $
```

De forma predeterminada, el pin exportado se establece en input ( **in** ). Queremos controlar un diodo, por lo que necesitamos una salida ( **out** ). Necesitamos cambiar la dirección del pin. Para ello, simplemente escribimos **out** en el fichero `/sys/class/gpio/gpio21/direction`:

```
echo out > /sys/class/gpio/gpio21/direction
```

De ahora en adelante, podemos controlar el diodo escribiendo los valores en el archivo de valores.

Para encender el diodo emitimos el comando:

```
echo 1 > /sys/class/gpio/gpio21/value
```

Por tanto, para apagarlo con 0:

```
echo 0 > /sys/class/gpio/gpio21/value
```

Cuando terminemos los experimentos, podemos limpiar, es decir, **deshabilitar el acceso al pin GPIO21**. Lo exportamos al principio, por lo que ahora necesitamos deshacer esta operación:

```
echo 21 > /sys/class/gpio/unexport
```

Como puedes ver en el mundo de Linux, todo es un archivo. El directorio `/sys` se comporta como archivos normales en la tarjeta SD. Podemos utilizar herramientas estándar de la consola o nuestros propios programas para controlar los dispositivos conectados. Curiosamente, ejecutar los ejemplos no requirió programación nuestra, son solo operaciones de archivo normales. Por supuesto, este no es un método eficaz, pero en muchos casos será suficiente.

Vale la pena mencionar que usando el acceso al **directorio/sys/class/gpio** también puede controlar los pines en otras plataformas. Esta funcionalidad la proporciona el kernel de Linux, no depende de la Raspberry Pi.

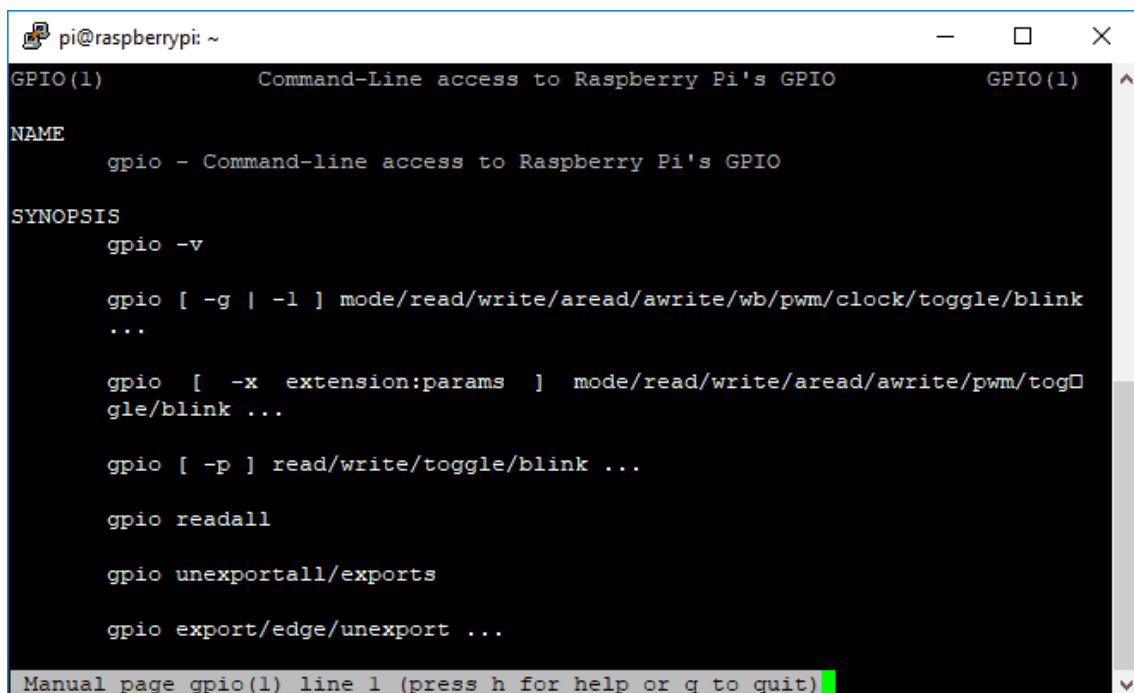
### 5.1.2. GPIO y WiringPi

Ya hemos aprendido el método de comunicación directa con el kernel. Esta es una opción interesante, pero no muy efectiva y no necesariamente conveniente en algunos casos. Afortunadamente, la Raspberry Pi viene con algunas bibliotecas y programas bastante buenos que te permiten acceder a los pines de manera más eficiente. Uno de ellos es WiringPi.

En esta parte del curso no escribiremos programas (solo scripts simples), por lo que usaremos un programa llamado **gpio** - con él puede usar fácilmente líneas GPIO desde la consola.

Puede encontrar más información sobre el programa en la extensa documentación:

```
man gpio
```



```
pi@raspberrypi: ~  
GPIO(1)      Command-Line access to Raspberry Pi's GPIO      GPIO(1)  
NAME  
    gpio - Command-line access to Raspberry Pi's GPIO  
SYNOPSIS  
    gpio -v  
  
    gpio [ -g | -l ] mode/read/write/aread/awrite/wb/pwm/clock/toggle/blink  
    ...  
  
    gpio [ -x extension:params ] mode/read/write/aread/awrite/pwm/tog  
    gle/blink ...  
  
    gpio [ -p ] read/write/toggle/blink ...  
  
    gpio readall  
  
    gpio unexportall/exports  
  
    gpio export/edge/unexport ...  
Manual page gpio(1) line 1 (press h for help or q to quit)
```

Podemos ver en el apartado SYNOPSIS las diferentes opciones de comandos.

Para usar comandos con WiringPi, verificaremos primero la versión de esta librería instalada. Para hacer esto, ejecuta `gpio -v`. Si el número de versión es 2.46 o superior, entonces todo bien y

puedes continuar. De ser inferior, usa estos comandos para instalar una versión superior, como hemos dicho anteriormente:

```
cd /tmp
wget https://unicorn.drogon.net/wiringpi-2.46-1.deb
sudo dpkg -i wiringpi-2.46-1.deb
```

Ahora vamos con un sencillo ejemplo de encendido de un LED con el mismo conexionado que el ejemplo anterior, pero esta vez en el pin 12 (Modo BCM), que es nuestro pin físico 32 en el conector.

Como se mencionó, la biblioteca WiringPi usa una numeración de pines diferente. Sin embargo, puede forzar el uso de los mismos números que para el directorio / sys . Quedémonos con esta numeración ahora.

Para empezar, configuramos el pin 12 como salida emitiendo el comando:

```
gpio -g mode 12 out
```

La opción **-g** de **gpio** nos hace usar la numeración de pines de la CPU.

Ahora podemos apagar y encender el LED escribiendo después del número de pin su estado, 0 lógico o 1 lógico respectivamente:

```
gpio -g write 12 0
gpio -g write 12 1
```

Obtuvimos la misma funcionalidad que antes, pero no tuvimos que hacer referencia a los archivos directamente, lo que hizo que todo el "código" fuera más corto.

## WiringPi

Por supuesto, para la prueba podemos intentar usar la numeración de la biblioteca WiringPi. Volvemos a la tabla con pines ( **gpio readall** ) y buscamos nuestro pin número 12. Según la numeración interna de la biblioteca, tendrá el número 26.

Physical	V	Mode	Name	wPi	BCM
1	2		5v		
3	4		5v		
5	6		0v		
7	8	0	IN	TxD	15
9	10	1	IN	RxD	16
11	12	0	IN	GPIO. 1	1
13	14		0v		
15	16	0	IN	GPIO. 4	4
17	18	0	IN	GPIO. 5	5
19	20		0v		
21	22	0	IN	GPIO. 6	6
23	24	1	IN	CE0	10
25	26	1	IN	CE1	11
27	28	1	IN	SCL.0	31
29	30		0v		
31	32	0	IN	GPIO.26	26
33	34		0v		
35	36	0	IN	GPIO.27	27
37	38	0	IN	GPIO.28	28
39	40	0	IN	GPIO.29	29

Entonces podemos emitir comandos:

```
gpio mode 26 out
gpio write 26 0
gpio write 26 1
```

Podemos comprobar como el funcionamiento será idéntico.

## 5.2. Scripts, programas simples

Ya podemos controlar salidas universales. Sin embargo, el método de emitir comandos manualmente no es muy conveniente. Afortunadamente, no tenemos que escribir manualmente todos los comandos en la terminal.

Podemos escribir scripts que se ejecutarán como "programas simples".

Un script es simplemente un archivo de texto que contiene los comandos que normalmente ingresamos desde el teclado en una ventana de terminal. Gracias a colocarlos en el archivo, no tenemos que volver a escribir todo, simplemente llame al archivo correspondiente.

### 5.2.1. Script encendido de LED

Comenzaremos con el mismo ejemplo extremadamente simple, el encendido de un LED. Haremos que nuestro primer script configure el pin apropiado como salida y lo ponga alto (es decir, encienda el LED). Por supuesto, no tiene mucho sentido hacer un "programa" para algo tan simple, pero con esto practicaremos la creación y ejecución del script en sí.

Primero, necesitamos crear un nuevo archivo con la extensión ".sh." Por ejemplo, puede ser led.sh, para ello usaremos el editor nano:

```
nano led.sh
```

En su contenido, ingresamos los comandos que normalmente ingresaríamos en la consola, tenemos que precederlos con una sola línea: **#!/bin/sh**. Entonces, el script que enciende el diodo conectado al pin 21 se vería así:

```
#!/bin/sh
gpio -g mode 21 out
gpio -g write 21 1
```

Guarde el archivo ( CTRL + X ). A continuación, debemos otorgarle a nuestro script los permisos adecuados para ser tratado como un "programa". Para esto usamos **chmod** :

```
sudo chmod +x led.sh
```

Para ejecutar el script, emitimos el comando:

```
./led.sh
```

Ahora el LED se encenderá sin más. Para apagarlo se necesitaría otro script distinto, que puedes probar por ti mismo. Vamos a poner ahora otro ejemplo más funcional, hacer que el led **parpadee** por sí solo. Creamos un nuevo archivo blink.sh :

```
nano blink.sh
```

En su contenido pegamos un bucle **while** simple (de momento no nos centraremos en la sintaxis):

```
#!/bin/sh
gpio -g mode 21 out
while true
do
    gpio -g write 21 1
    sleep 1
    gpio -g write 21 0
    sleep 1
done
```



Un bucle while, como su traducción indica, consiste en que algo se repita mientras algo se mantenga. En este caso, en modo breve, podemos ver que mientras sea “true” haga (do) las líneas siguientes. Por lo que se ejecutará en bucle lo que haya entre el “do” y el “done”.

El código anterior configura el pin número 21 como salida. Luego cambia de estado cada segundo (sleep 1). Guarde el archivo ( CTRL + X ), dale los permisos adecuados ( sudo chmod + x blink.sh ). La secuencia de comandos puede ser interrumpida por CTRL + C .

Ahora podrás ver como el LED parpadea cada segundo cuando lo ejecutemos.

Basándose en los ejemplos descritos aquí, ya tienes conocimientos para poder crear, por ejemplo, una simulación de semáforos simples.

## 6. Crear un usuario nuevo

La gestión de usuarios y contraseñas en Raspberry Pi es de vital importancia en la seguridad del equipo ya que, si vamos a conectar este a redes, debemos estar seguros de que nadie más pueda hacerlo.

Crear el propio usuario y eliminar el por defecto, es una medida de seguridad básica cuando sales a internet. Afortunadamente, la gestión de permisos, contraseñas, grupos y usuarios, así como la configuración de la seguridad son uno de los puntos fuerte de los sistemas basados en Linux como Raspberry Pi OS (Raspbian).

Simplemente por recordar, la contraseña por defecto en RPi OS es:

Username: pi  
Password: raspberry

Actualmente, y como hemos visto, al comienzo de la instalación del SO, nos pide la creación de una contraseña, y hemos visto desde el menú como poder cambiarla. Para cambiar la contraseña de nuestro usuario, desde una consola de comandos, usamos el comando:

*passwd*

A continuación, escribimos la contraseña actual y después la nueva dos veces.

```
pi@raspberrypi:~ $ passwd
Cambiando la contraseña de pi.
(actual) contraseña de UNIX:
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
pi@raspberrypi:~ $
```

Si lo que queremos es cambiar la contraseña de otro usuario, simplemente añadimos el nombre del usuario como parámetro

*sudo passwd nombreUsuario*

Siendo nombreUsuario el nombre del usuario que queremos cambiar la contraseña. Obviamente únicamente podemos ejecutar esta acción si somos usuarios root del sistema.

### 6.1. Crear y borrar usuarios

Gestionar usuarios es otra tarea de mantenimiento para cualquier sistema. Aunque seáis el único usuario que tendrá la RPi, tras su instalación es conveniente que creéis vuestro propio usuario, y loggeéis en él, y luego eliminéis el usuario por defecto 'Pi'.

Para crear un nuevo usuario empleamos este comando:

*adduser nombreUsuario*

```
pi@raspberrypi:~$ sudo adduser usuario
Adding user `usuario' ...
Adding new group `usuario' (1002) ...
Adding new user `usuario' (1002) with group `usuario' ...
Creating home directory `/home/usuario' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for usuario
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
pi@raspberrypi:~$
```

Borrar un usuario es igualmente sencillo, usando el comando:

*deluser nombreUsuario*

```
pi@raspberrypi:~$ sudo deluser usuario
Removing user `usuario' ...
Warning: group `usuario' has no more members.
Done.
pi@raspberrypi:~$
```

¡Sin más! Entonces si queremos eliminar el por defecto pondremos `deluser pi`. Por otro lado, en ocasiones veréis una forma alternativa de crear y borrar usuarios, mediante los comandos `'useradd'` y `'userdel'`. Ver que existen dos grupos de comandos puede generar cierta confusión.

La explicación es que `'useradd'` y `'userdel'` son archivos binarios del sistema, mientras que `'adduser'` y `'deluser'` son scripts perl que usan a los binarios anteriores.

Debemos acostumbrarnos a usar exclusivamente los anteriores `'adduser'` y `'deluser'` desde la consola de comandos. Por su parte `'useradd'` y `'userdel'` están más pensados para usar desde scripts. No obstante, aquí os dejamos la sintaxis de estos comandos alternativos:

*#crear un nuevo usuario*  
*useradd nombreUsuario*

*#crear usuario, version completa*  
*useradd -c "Nombre Usuario" -g grupo -d /home/usuario -s /bin/bash nombreUsuario*

*#borrar usuario*  
*userdel nombreUsuario*

*#borrar usuario y eliminar su directorio home*  
*userdel -r nombreUsuario*

## 6.2. Gestión de sesiones de usuario

Algunos comandos adicionales útiles para el control de sesiones de usuario. En primer lugar, para finalizar la sesión usamos el comando:

*logout*

Si durante una sesión queremos actuar brevemente como otro usuario, sin cerrar sesión, podemos cambiar de forma temporal con el siguiente comando

*su - nombreUsuario*

```
pi@raspberrypi:~$ su - usuario
Password:

SSH is enabled and the default password for the 'pi' user has not been
changed.
This is a security risk - please login as the 'pi' user and type 'pass
wd' to set a new password.

usuario@raspberrypi:~$
```

Para finalizar esta "sesión temporal" y volver a nuestro usuario "normal" simplemente hacemos:

*exit*

Si queremos ver el nombre del usuario con el que estamos conectado actualmente usamos:

*whoami*

```
pi@raspberrypi:~$ whoami
pi
pi@raspberrypi:~$
```

Aunque ya podemos verlo normalmente delante del @.

También podemos listar todos los usuarios que están conectados en el sistema con el comando:

*who*

Por último, si queremos obtener un registro del último inicio de sesión de los usuarios podemos usar:

*lastlog*

### 6.3. Grupos de usuarios

Los grupos de usuario facilitan la configuración de permisos en sistemas donde hay un gran número de usuarios, donde controlar los permisos de forma individual sería impracticable. Ten en cuenta que los sistemas basados en Unix y sus aplicaciones están pensadas para sistemas con múltiples usuarios (incluso cientos o miles).

Si tienes un sistema con 1 o 2 usuarios, es posible que puedas gestionar los permisos a nivel individual. Pero a medida que el número de usuarios crece, con decenas o cientos de usuarios, la gestión se te va a volver un poco impracticable y por ello vamos a explicar como hacerlo más sencillo mediante la gestión de grupos de usuarios.

Lo normal es trabajar con grupos de usuarios y conceder los permisos a carpetas o ejecución de programas a nivel de grupos de usuarios. Después, únicamente tienes que encargarte de que cada usuario forme parte de los grupos oportunos y tendrán los permisos que corresponden. Por ejemplo, los usuarios del grupo Finanzas tendrás permisos distintos al de Técnicos.

Para **crear un nuevo grupo de usuarios** simplemente escribimos:

*groupadd nombre\_grupo*

Siendo *nombre\_grupo* el nombre del grupo que queremos crear. Si lo que queremos es borrar un grupo de usuarios existente usamos:

*groupdel nombre\_grupo*

Para renombrar un grupo de usuarios tenemos el comando:

*groupmod -n nombre\_nuevo nombre\_anterior*

Para **añadir el usuario actual a un grupo** podemos usar este comando:

```
newgrp nombre_grupo
```

Si queremos añadir otro usuario a un grupo usaremos el comando:

```
adduser nombre_usuario nombre_grupo
```

Alternativamente podemos usar el siguiente comando (útil si queremos añadirlo a más de un grupo en un único comando)

```
usermod -a -G nombre_grupo1, nombre_grupo2, nombre_grupo3 nombre_usuario
```

Finalmente, para quitar a un usuario de un grupo de usuarios, usamos el comando

```
deluser nombre_usuario nombre_grupo
```

Que únicamente quitará al usuario del grupo de usuarios, no eliminará al usuario en sí.

Para **listar todos los grupos** a los que pertenece el usuario actual, simplemente, usamos:

```
groups
```

También podemos mostrar los grupos a los que pertenece otro usuario con:

```
groups nombre_usuario
```

Por último, si lo que queremos es listar todos los grupos existentes en la máquina actual usamos el comando:

```
cut -d: -f1 /etc/group
```

La instalación de Raspberry Pi OS crea ciertos grupos de usuarios por defecto. A continuación, un resumen de los principales grupos de usuarios por defecto y su propósito:

Grupo	Descripción
pi	Grupo de usuario. (Se crea un nuevo grupo automáticamente cada vez que se crea un usuario)
sudo	Acceso sudo
adm	Acceso a los ficheros de log ubicados en /var/log
cdrom	Acceso a unidades ópticas
audio	Acceso a dispositivos de audio (micrófonos, tarjeta de sonido, etc)
video	Acceso a dispositivos de video (tarjeta gráfica, framebuffer, webcams)
plugdev	Acceso a dispositivos de almacenamiento externo
input	Acceso a carpeta /dev/input/mice
netdev	Acceso a dispositivos de red
dialout	Acceso a puertos serie/modems, etc
gpio	Acceso al puerto GPIO
i2c	Acceso al bus I2C
spi	Acceso al bus SPI

## 7. Configuración IP estática

Hay muchas aplicaciones en la Raspberry Pi que nos permiten conectarnos desde otro ordenador a esta para hacer cosas como compartir archivos, un servidor web, o administrarla. Para eso debemos conocer la IP de esta e ingresarla en un programa o desde el navegador. Esto se puede hacer usando el comando *ip addr* en el terminal. Pero el problema viene cuando la IP es dinámica y cambia, teniendo que consultarla cada dos por tres.

Una IP es una dirección que asigna el router de nuestra casa a cualquier ordenador o dispositivo que se conecta a este. Esta IP suele cambiar cada cierto tiempo entre aparatos dependiendo de los que haya conectados. Esta dirección identifica el dispositivo dentro de la red para que sea fácil de acceder a él. Esto mismo ocurre cuando nuestro router se conecta a internet. Obtiene una IP para que otros ordenadores dentro de internet se puedan conectar al router para enviar y recibir datos.

Tanto las IPs de nuestro dispositivo son únicas dentro de nuestra red, como la IP del router es única en internet. Evitando así conflictos cuando intentamos conectarnos a un ordenador en concreto. Asignado una IP fija a nuestra Raspberry Pi dentro de nuestra red conseguiremos que no cambie y que siempre la conozcamos. El formato de la IP es de 4 números que van de 0 a 255 separados por puntos.

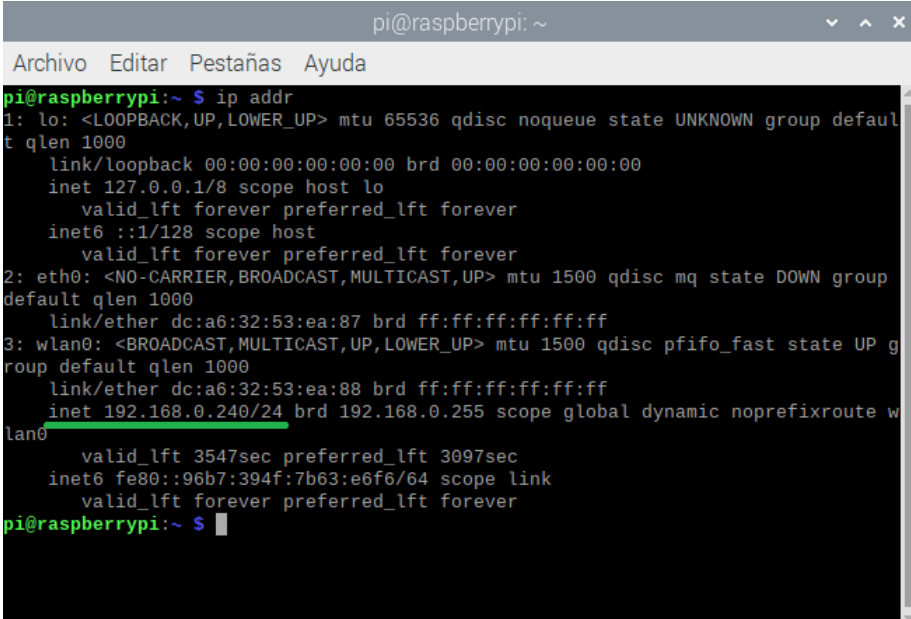
En este apartado veremos cómo asignar una IP fija o estática a una Raspberry Pi, evitando ese problema. Se puede configurar desde dos formas, desde el terminal, o la más sencilla, desde el propio menú de SO.

### 7.1. Configuración desde el menú

Lo primero que podemos hacer es ver nuestra actual IP desde el terminal, para ello, como hemos dicho, ponemos el comando:

*ip addr*

Entonces veremos, según si estemos conectados por cable Ethernet o wlan (Wireless LAN, o sea WiFi), nuestra dirección IP. En la imagen podemos ver como el valor de inet nuestra IP dinámica es 192.168.0.240/24. Ese /24 es la máscara.



```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether dc:a6:32:53:ea:87 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether dc:a6:32:53:ea:88 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.240/24 brd 192.168.0.255 scope global dynamic noprefixroute wlan0
        valid_lft 3547sec preferred_lft 3097sec
    inet6 fe80::96b7:394f:7b63:e6f6/64 scope link
        valid_lft forever preferred_lft forever
pi@raspberrypi:~$
```

En esta imagen vemos las interfaces de red de la Raspberry Pi y, si hay alguna conectada a la red, algunos valores de estas. Las que nos interesan son eth0 y wlan0. Eth0 se trata del cable de red y

wlan0 de la WiFi. Ahora nos haría falta saber cuál es la puerta de enlace y el DNS para tenerlo todo correcto y no tener problemas una vez asignemos una IP fija a la Raspberry Pi.

La puerta de enlace es la IP del dispositivo que está conectado directamente a internet. Normalmente la puerta de enlace se encarga de asignar las IPs automáticamente a los dispositivos de nuestra red.

En cuanto a el DNS (Domain Name System), son IPs son servidores que se encargan de traducir las direcciones web a IPs cuando navegamos por internet. Gracias a estos “traductores” podemos ir a las páginas web de forma más fácil sin tener que memorizar la IP de cada sitio web que visitamos. Normalmente la puerta de enlace se encarga de conectarse a las DNS, aunque podemos configurar otras diferentes en nuestro dispositivo.

Para conocer la puerta de enlace debemos ejecutar el comando:

*ip route show*

```
pi@raspberrypi:~ $ ip route show
default via 192.168.0.1 dev wlan0 proto dhcp src 192.168.0.240 metric 303
192.168.0.0/24 dev wlan0 proto dhcp scope link src 192.168.0.240 metric 303
pi@raspberrypi:~ $
```

Para terminar, tenemos que saber cuál es el valor de nuestra DNS. Esto valor lo sacaremos consultando el contenido del archivo /etc/resolv.conf. La forma más sencilla es ejecutando el comando

*cat /etc/resolv.conf*

```
pi@raspberrypi:~ $ cat /etc/resolv.conf
# Generated by resolvconf
domain technicolor.net
nameserver 192.168.0.1
pi@raspberrypi:~ $
```

De aquí podemos sacar las DNS donde pone nameserver. Esto será necesario si queremos navegar por internet una vez lo dejemos todo configurado.

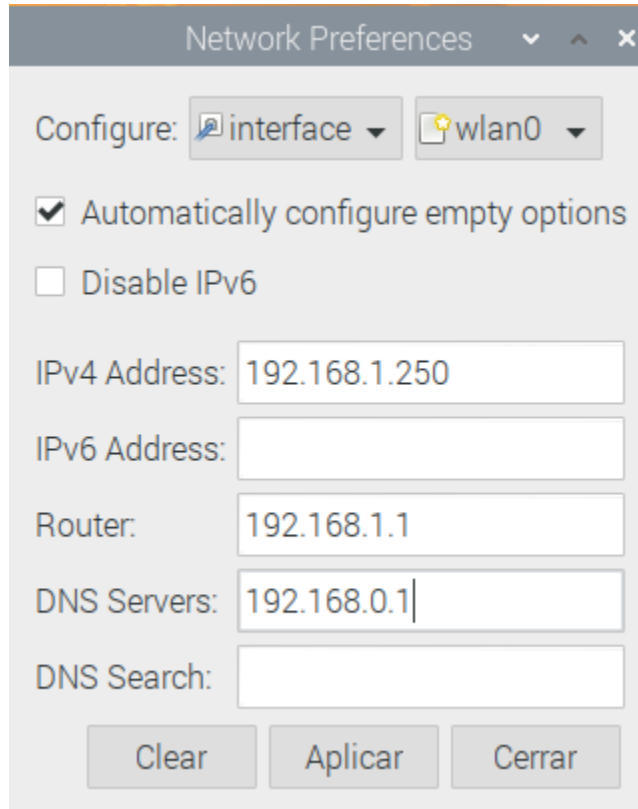
Ahora para hacerla estática, vamos hacemos clic derecho sobre el icono WiFi de la barra de tareas del escritorio. Entonces se nos abrirá la siguiente ventana, donde clicaremos en la primera opción, configuración de Redes inalámbricas y cableadas:



A continuación, se nos abrirá una ventana, donde deberemos introducir la IP que permanecerá estática. Para ello, seleccionamos si se trata de conexión eth0 o wlan0. Nosotros vamos a fijar la IP estática de la wlan0 a 192.168.1.250, y las direcciones IP del router y de los servidores DNS



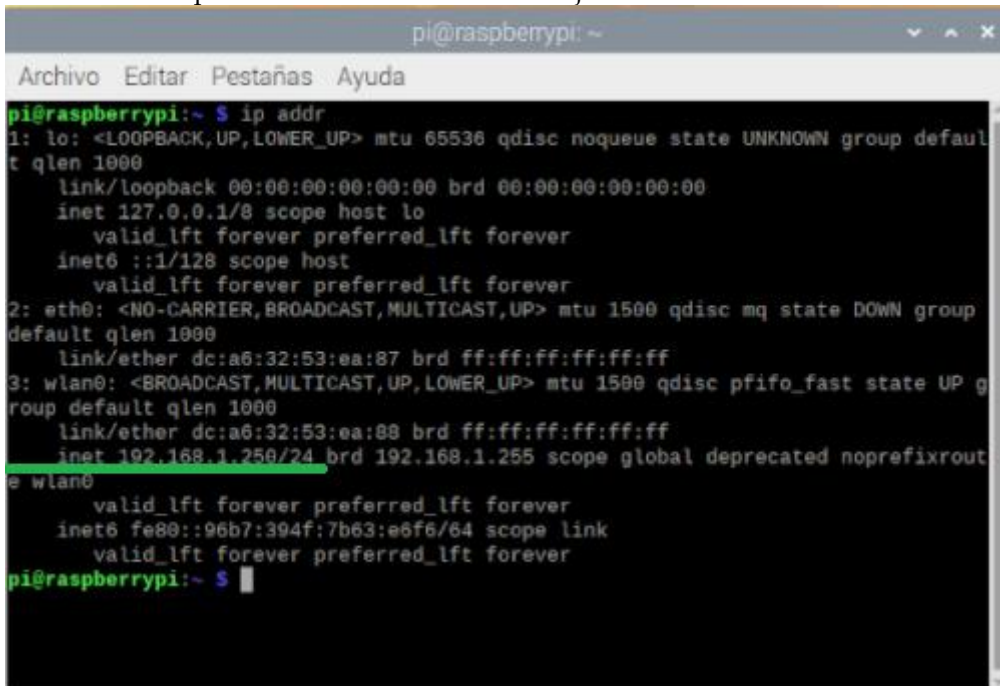
serán las que nos han salido anteriormente. En el caso de que nos salieran más de una dirección IP en DNS servers, añadiremos las dos separadas por un espacio. Por lo que rellenamos de la siguiente forma:

A screenshot of the 'Network Preferences' window. At the top, it says 'Configure: interface wlan0'. Below that, there are two checkboxes: 'Automatically configure empty options' (checked) and 'Disable IPv6' (unchecked). Then, there are text input fields for 'IPv4 Address' (192.168.1.250), 'IPv6 Address' (empty), 'Router' (192.168.1.1), 'DNS Servers' (192.168.0.1), and 'DNS Search' (empty). At the bottom, there are three buttons: 'Clear', 'Aplicar', and 'Cerrar'.

Le daremos a Aplicar y a Cerrar. Una vez hecho esto, debemos reiniciar la RPi. Para ello podemos hacerlo desde el escritorio, o escribiendo en el terminal el comando:

*sudo reboot*

Una vez reiniciada, abrimos el terminal y lo comprobamos poniendo el comando dicho anteriormente: `ip addr`. Vemos como se nos ha fijado la dirección IP.

A screenshot of a terminal window on a Raspberry Pi. The prompt is 'pi@raspberrypi: ~'. The command 'ip addr' has been entered and executed. The output shows details for three network interfaces: 'lo' (loopback), 'eth0' (ethernet), and 'wlan0' (wireless). The 'wlan0' interface is highlighted with a green underline, showing it has an IPv4 address of 192.168.1.250/24. The prompt returns to 'pi@raspberrypi: ~ \$'.

## 7.2. Configuración desde la consola

Este método es un poco menos visual, pero está bien aprenderlo. En el caso que no tengamos escritorio, no hace falta que instalemos uno para poder hacer lo mismo. Con la terminal nos podemos apañar de la misma manera. Para empezar, haremos lo mismo, conocer nuestra dirección IP, puerta de enlace y DNS. Usa los comandos del punto anterior. Para poder asignar una IP fija a nuestra Raspberry Pi solo tenemos que modificar el archivo `/etc/dhcpd.conf` con el siguiente comando:

```
sudo nano /etc/dhcpd.conf
```

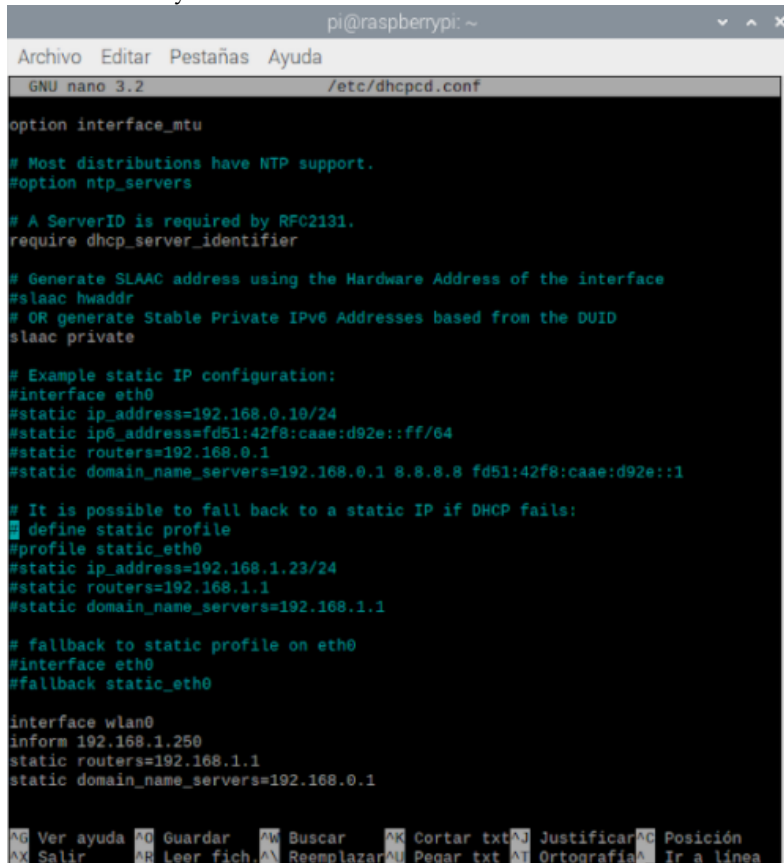
En este archivo debemos añadir las siguientes líneas al final del archivo, dependiendo de los datos que hayamos conseguido en el primer paso. En la línea **interface** escribiremos el nombre de la interfaz a la que queremos asignarle la IP estática. Si es el cable de red será la interface `eth0` y si es la WiFi será la interface `wlan0`. A continuación, añadiremos una línea llamada `static ip_address=`, con la IP que queremos asignar a la Raspberry Pi terminado en `/24`. En la siguiente línea escribiremos `static routers=192.168.1.1` donde `192.168.1.1` será la IP de la puerta de enlace. Y en la última línea pondremos `static domain_name_servers=` y los servidores DNS que tengamos, siempre separados por un espacio. En nuestro caso quedaría como sigue:

```
interface wlan0
static ip_address=192.168.1.250/24
static routers=192.168.1.1
static domain_name_servers=192.168.0.1
```

A continuación, le damos a `Ctrl+X`, nos dirá que guardemos, pulsamos `S` y ya estamos listos para reiniciar la RPi.

Como antes, podemos usar `ip addr` para comprobar que se ha cambiado.

OJO: Aunque si has hecho el punto anterior mediante el escritorio, ya lo tendrías cambiado de antes te aparecerá el archivo ya modificado como se muestra a continuación:



```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 3.2 /etc/dhcpd.conf

option interface_mtu

# Most distributions have NTP support.
#option ntp_servers

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate SLAAC address using the Hardware Address of the interface
#slaac hwaddr
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private

# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0

interface wlan0
inform 192.168.1.250
static routers=192.168.1.1
static domain_name_servers=192.168.0.1

^O Ver ayuda ^O Guardar ^M Buscar ^M Cortar txt ^J Justificar ^C Posición
^M Salir ^M Leer fich. ^M Reemplazar ^M Pegar txt ^T Ortografía ^M Ir a línea
```

Con esto ya tendrías fijada tu IP para poder tenerla disponible en tus proyectos.

## 8. Conectarse remotamente a la RPi

En muchas ocasiones vamos a querer controlarla de forma remota nuestra RPi, y más si no tiene conectado un monitor y un teclado, o la queremos dedicar a funciones como un servidor web o un bróker para IoT (Internet of Things). Las formas habituales de conectarnos de forma remota son SSH y VNC

### 8.1. Conectarse con SSH

La conexión remota con SSH es la más empleada, por ser rápida y segura. SSH son las siglas de Secure Shell (antes conocido como Secure Socket Shell) y, básicamente, nos permite iniciar una consola de comandos en un ordenador remoto de forma sencilla y segura.

SSH sigue un modelo cliente-servidor. El cliente inicia una petición al servidor, que autentifica la comunicación e inicia el entorno Shell. Múltiples clientes pueden conectarse a un mismo servidor.

Para podernos conectar a nuestra Raspberry Pi, esta va a funcionar como servidor. Antes de poder conectarnos tenemos que activar la opción en la configuración ya que, por seguridad, el servidor SSH viene desactivado en nuestro OS. Lo haremos como hemos visto en el menú de configuración.



También deberemos saber la dirección IP de la RPi, que por ello hemos comentado en otro punto como hacerla fija.

En principio solo funcionará desde nuestra red local, si quisieras hacerlo desde internet es posible, pero deberás configurar un mapeo de puertos en el router consultando su documentación.

#### 8.1.1. Linux

Desde Linux o Mac es muy simple, ya que hay un terminal nativo para estas conexiones. En Linux podemos conectarnos a la RPi simplemente poniendo en la consola terminal:

`ssh user@host`

donde 'user' es el nombre del usuario en el Raspberry Pi servidor y 'host' es la dirección IP o el nombre del equipo al que nos queremos conectar. Por ejemplo: `ssh pi@192.168.1.250`

```
login as: pi
pi@raspberrypi:~$ sudo passwd
Linux raspberrypi 4.9.59-v7+ #1047 SMP Sun Oct 29 12:19:23 GMT 2017 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Oct 29 12:19:23 GMT 2017 from 192.168.1.100

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~$
```

No es recomendable usar el usuario por defecto, por eso la opción SSH viene desactivada por defecto y además nos aparece ese mensaje de advertencia.

### 8.1.2. Windows

En Windows, deberemos descargarnos un cliente, y el más utilizado es PuTTY, el cual es sencillo y gratuito descargable [desde su web](#).

Download PuTTY: latest release (0.74)

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)  
Download: [Stable](#) | [Snapshot](#) | [Docs](#) | [Changes](#) | [Wishlist](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.74, released on 2020-06-27.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a [permanent link to the 0.74 release](#).

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the [development snapshots](#), to see if the problem has already been fixed in those versions.

#### Package files

You probably want one of these. They include versions of all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

##### MSI ('Windows Installer')

32-bit: [putty-0.74-installer.msi](#) (or by [FTP](#)) (signature)

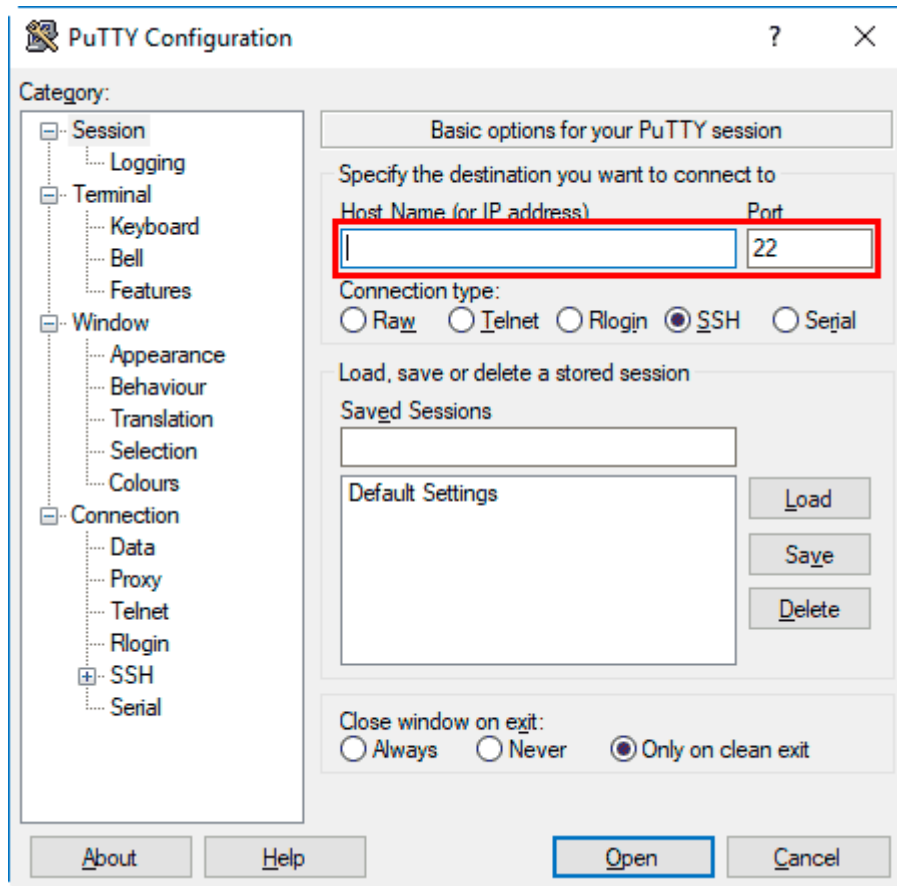
64-bit: [putty-64bit-0.74-installer.msi](#) (or by [FTP](#)) (signature)

##### Unix source archive

.tar.gz: [putty-0.74.tar.gz](#) (or by [FTP](#)) (signature)

#### Alternative binary files

Una vez lo ejecutemos, debemos poner el puerto, que por defecto en la RPi es el 22 e introducir la dirección IP.



Tras el login, tenemos una ventana de comandos similar a la que obtenemos en Linux o Mac. Opcionalmente podemos guardar la conexión poniéndole un nombre. Así podemos tener almacenadas las conexiones a varios servidores, y acceder cómodamente.

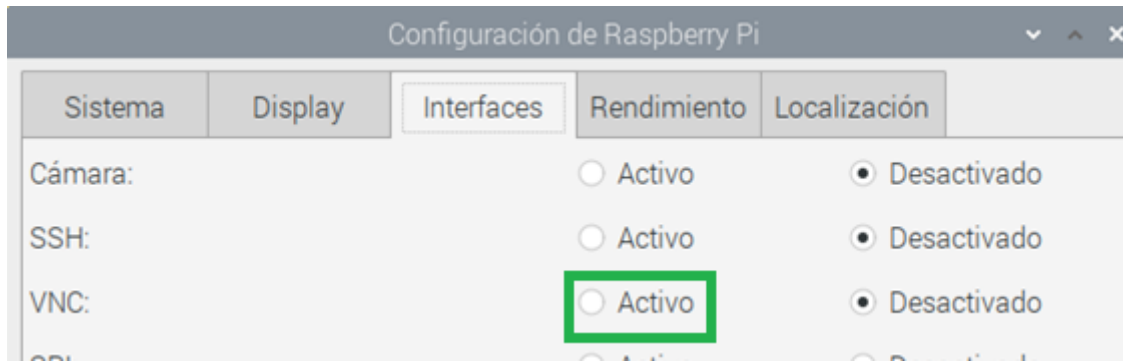
Así de fácil es realizar una conexión SSH, una herramienta casi imprescindible para trabajar con Raspberry Pi (en realidad, con cualquier servidor con Linux).

## 8.2. Conectarse mediante VNC

A diferencia de SSH, que únicamente abría una consola de comandos, con VNC podemos hacer un escritorio remoto completo. Es decir, que en la pantalla del cliente veremos la pantalla de Raspberry Pi, moveremos el ratón, y usaremos el casi como si estuviéramos conectados físicamente. La principal desventaja es que necesita transmitir una gran cantidad de datos, sobre todo por la necesidad de enviar la señal de video. Por otra parte, también es más inseguro respecto a SSH.

Un VNC, o Virtual Network Computing (Computación Virtual en Red) es un sistema de comunicación remota basado en el protocolo RFB (Remote FrameBuffer). VNC transmite la señal de video y los eventos de ratón y teclado entre varios ordenadores.

El ordenador al que nos vamos a conectar, en este caso nuestro Raspberry Pi, tiene que tener instalado el programa servidor RNC, el cual suele venir instalado por defecto en las nuevas versiones. Dentro del menú configuración, tendremos que activar la casilla de VNC.

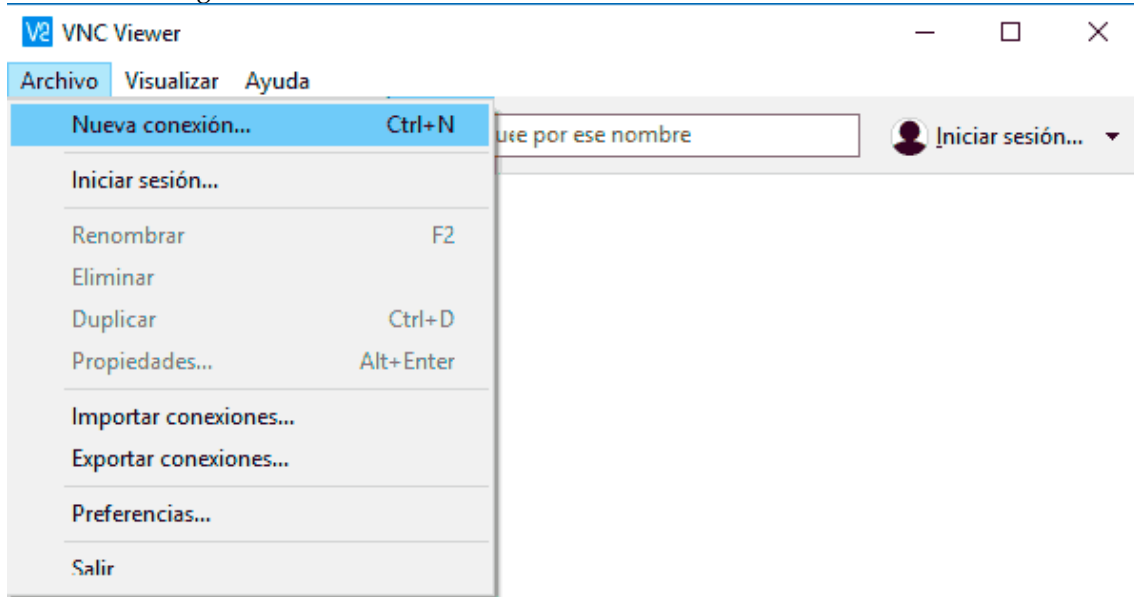


Si estamos ejecutando sistema operativo que no tiene instalado VNC server, podemos instalarlo ejecutando estos comandos:

```
sudo apt-get update
```

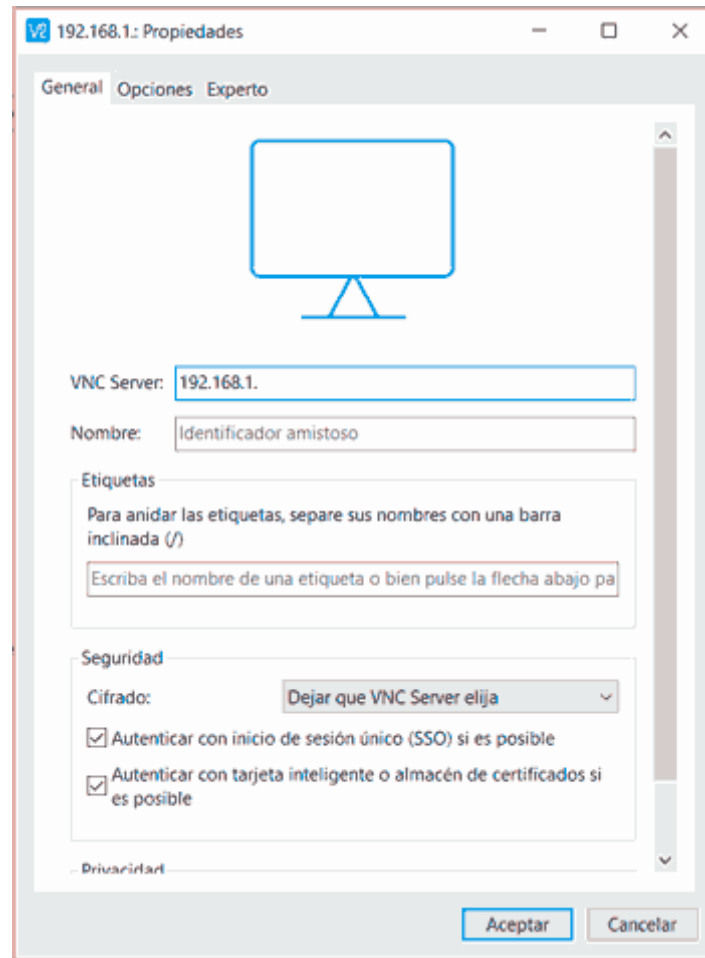
```
sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer
```

Después, configuramos el ordenador desde el que acceder de forma remota. Para ello usaremos en nuestro ordenador el Software **RealVNC Viewer**, que podremos instalar o incluso añadir la extensión a Google Chrome.



En primer lugar configuramos un ordenador remoto indicando la IP del sistema al que queremos conectarnos. Esta configuración se guarda para que sea cómodo conectarnos a nuestras máquinas





Al iniciar la conexión se nos pedirá el usuario y contraseña. Recordemos que por defecto es 'pi' como usuario y 'raspberrypi' como contraseña, pero que esto no es para nada seguro y es recomendable cambiarlo.

Es posible que la primera vez que conectemos con un VNC nos salga un cartel diciendo que la firma del dispositivo no es conocida. No pasa nada, es una medida de protección para impedir la suplantación. Si es la primera que nos conectamos, simplemente aceptamos para continuar.

Si la conexión ha ocurrido con éxito veremos en la pantalla de nuestro ordenador el video de la Raspberry Pi y podremos actuar con ella casi como si tuviéramos el monitor y el teclado conectado directamente. ¡Y así de fácil es trabajar remotamente!

## 9. Instalación y comunicación a través de UART

Otra forma de trabajar con RPi que ha quedado atrás es a través de UART, mediante los pines Serie que hemos visto anteriormente. Este método de instalación de SO a menudo se pasa por alto en el aprendizaje de Raspberry Pi. Sin embargo, vale la pena conocerlo, ¡definitivamente te será útil en el futuro!

Se utiliza sobretodo cuando no se dispone de una pantalla externa para la RPi, y se quiere usar un ordenador conectado a ella.

La comunicación UART es útil porque los puertos USB de la Raspberry son tipo HOST, como los del PC, están hechos para que conectes dispositivos a ellos, no para ser vistos como un dispositivo USB por otros dispositivos, como por ejemplo para un teclado. Sin embargo, mediante UART si que puedes hacer una comunicación en serie mediante la RPi y el ordenador.

## 9.1. Instalación Raspberry Pi OS Lite

Para esta instalación usaremos el sistema operativo en la **versión Lite**, es decir, sin interfaz gráfica. La imagen descargada de este RPi OS es mucho más pequeña, por lo que habrá más espacio en la tarjeta de memoria para nuestros archivos.

En el punto anterior del curso, describimos detalladamente el procedimiento para preparar una tarjeta microSD y la instalación del sistema operativo mediante NOOBS. Ahora vamos a hacer lo mismo, pero en lugar de la versión completa de Raspberry Pi OS, usaremos la versión Lite.

En el caso de la comunicación a través de UART, la interfaz gráfica o algunas aplicaciones como el juego Minecraft predeterminado no son necesarios, pero ocupan espacio y alargan el tiempo de instalación e inicio RPi.

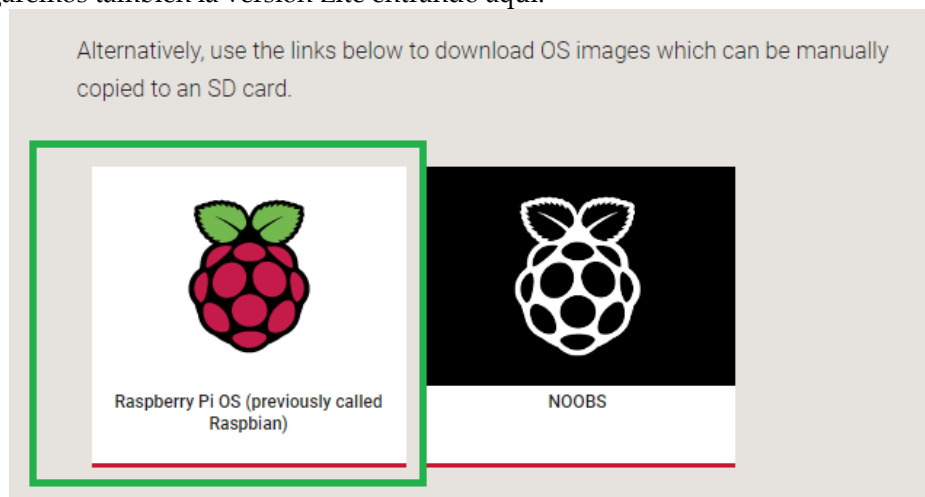
Lo primero que deberemos hacer, es descargar el “**Raspberry Pi Imager**” para instalar el **Raspberry Pi OS** de una forma sencilla. Para ello entramos en la siguiente web y descargamos los archivos.

<https://www.raspberrypi.org/downloads/>

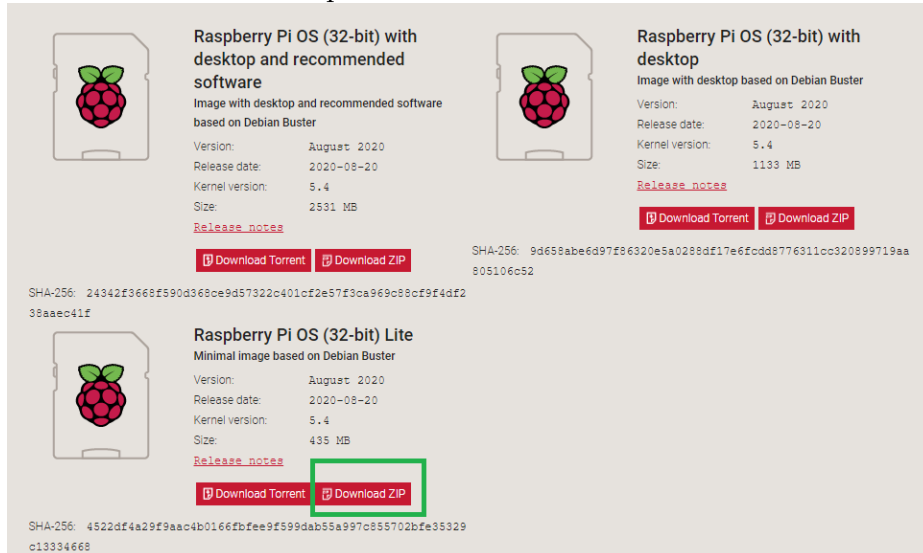
Descargaremos el correspondiente con nuestro sistema operativo.



Descargaremos también la versión Lite entrando aquí.



Y haciendo clic en la correspondiente:



**Raspberry Pi OS (32-bit) with desktop and recommended software**  
Image with desktop and recommended software based on Debian Buster

Version: August 2020  
Release date: 2020-08-20  
Kernel version: 5.4  
Size: 2531 MB

[Release notes](#)

[Download Torrent](#) [Download ZIP](#)

SHA-256: 24342f3668f590d368ce9d57322c401cf2e57f3ca969c88cf9f4df238a8ec41f

**Raspberry Pi OS (32-bit) with desktop**  
Image with desktop based on Debian Buster

Version: August 2020  
Release date: 2020-08-20  
Kernel version: 5.4  
Size: 1133 MB

[Release notes](#)

[Download Torrent](#) [Download ZIP](#)

SHA-256: 9d658abe6d97f86320e5a0288df17e6fcd8776311cc320899719aa805106c52

**Raspberry Pi OS (32-bit) Lite**  
Minimal image based on Debian Buster

Version: August 2020  
Release date: 2020-08-20  
Kernel version: 5.4  
Size: 435 MB

[Release notes](#)

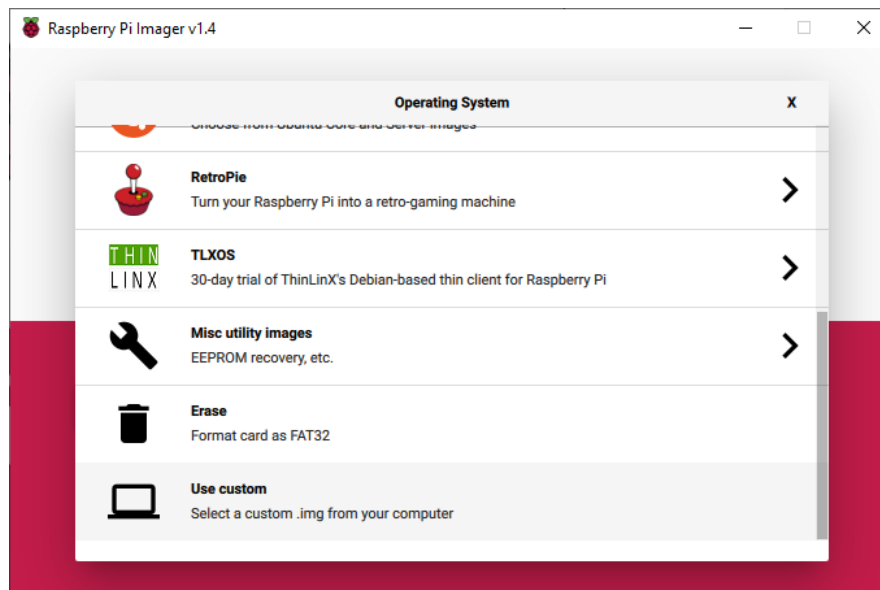
[Download Torrent](#) [Download ZIP](#)

SHA-256: 4522df4a29f9aac4b0166fbfee9f599dab55a997c855702bfe35329c13334668

Ahora abriremos el “Imager” y seleccionaremos el sistema operativo que habremos descargado, normalmente estará en un comprimido en la carpeta de descargas.



Para seleccionar el descargado iremos abajo, donde pone “Use custom”.



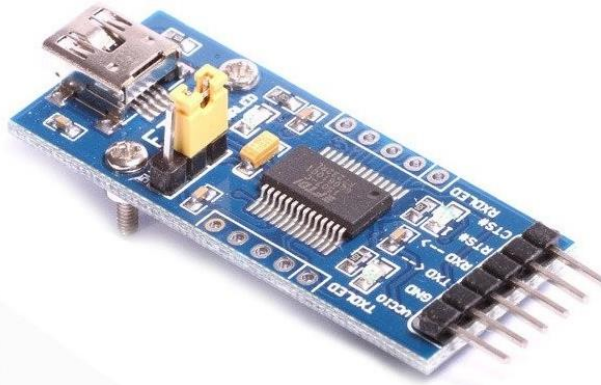
Una vez seleccionado, nos queda seleccionar la tarjeta SD y darle a “WRITE”:



Entonces nos aparecerá un mensaje de confirmación y al darle que sí, comenzará la instalación de la imagen del SO. Una vez finalizado, podemos conectar la SD a nuestra RPi y enchufarla. Nos aparecerá la siguiente pantalla a continuación.

## 9.2. Conexión puerto UART

Para la conexión de la Raspberry Pi con nuestro PC por el puerto UART necesitaremos un convertidor TTL UART, en nuestro caso usaremos [el siguiente](#):

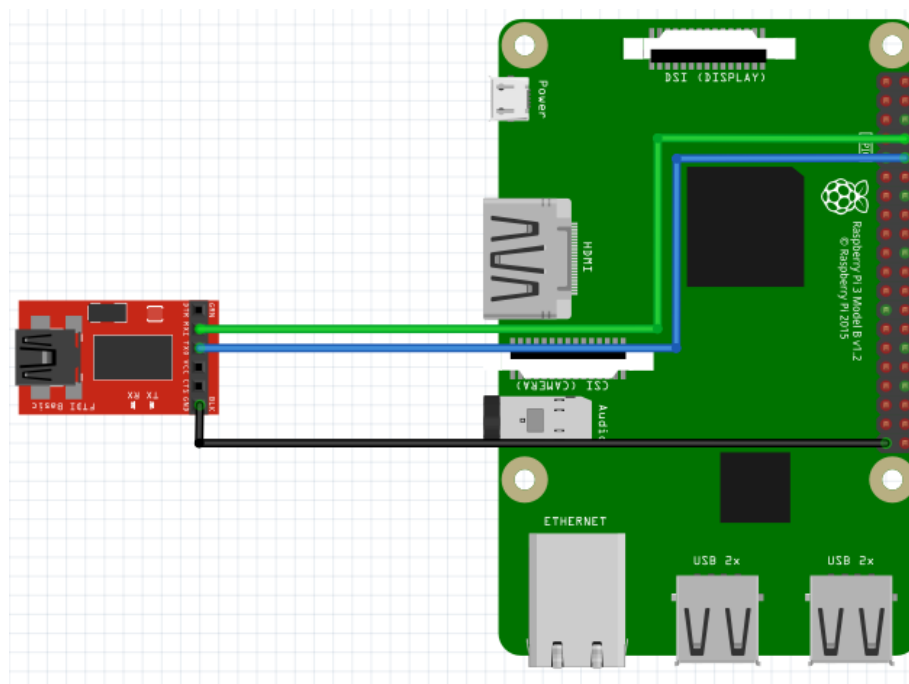


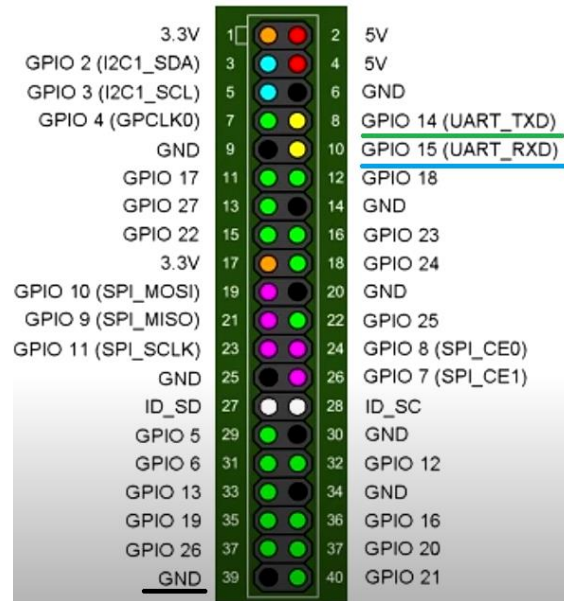
FTDI FT232RL Convertidor Mini USB a serie TTL UART

¡Importante! Para alimentar nuestra RPi, es recomendable hacerlo como siempre por el cable USB de alimentación, pero también es posible hacerlo mediante este convertidor con su salida de alimentación. Pero ¡ojo!, el convertidor utilizado para comunicarse con la Raspberry Pi debe estar funcionando con 3.3V, usar la versión de 5V puede dañar la placa Pi.

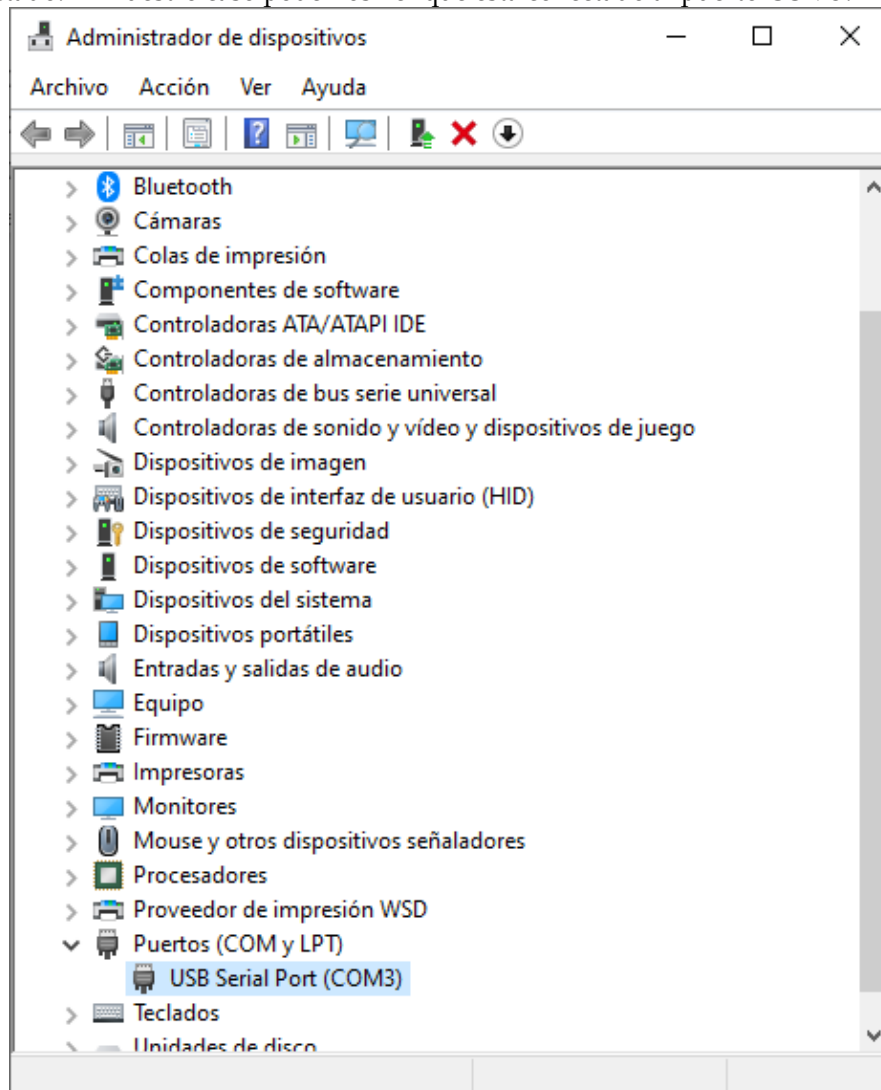
Por lo tanto, primero haremos las configuraciones necesarias, y una vez hechas, conectaremos la RPi a la alimentación.

El esquema que seguir será el siguiente:



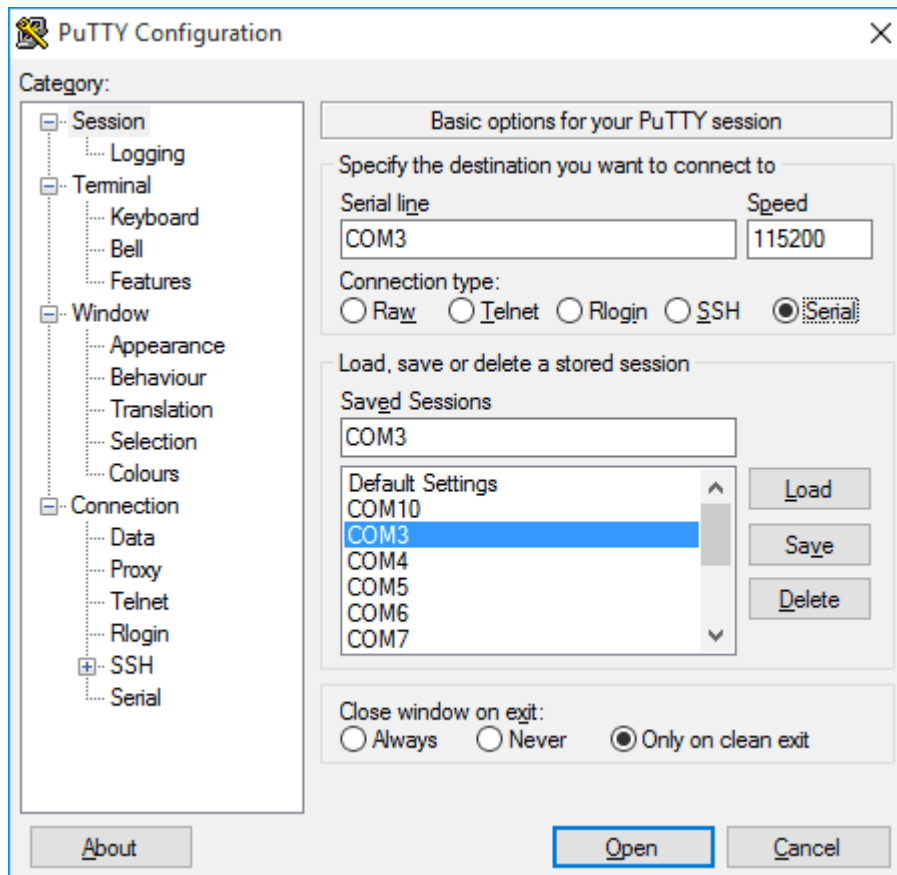


Una vez conectados los pines, conectamos el Adaptador por USB a nuestro ordenador. Ahora abrimos el Administrador de dispositivos y vamos a los puertos COM y LPT para ver en cual está conectado. En nuestro caso podemos ver que está conectado al puerto COM3.

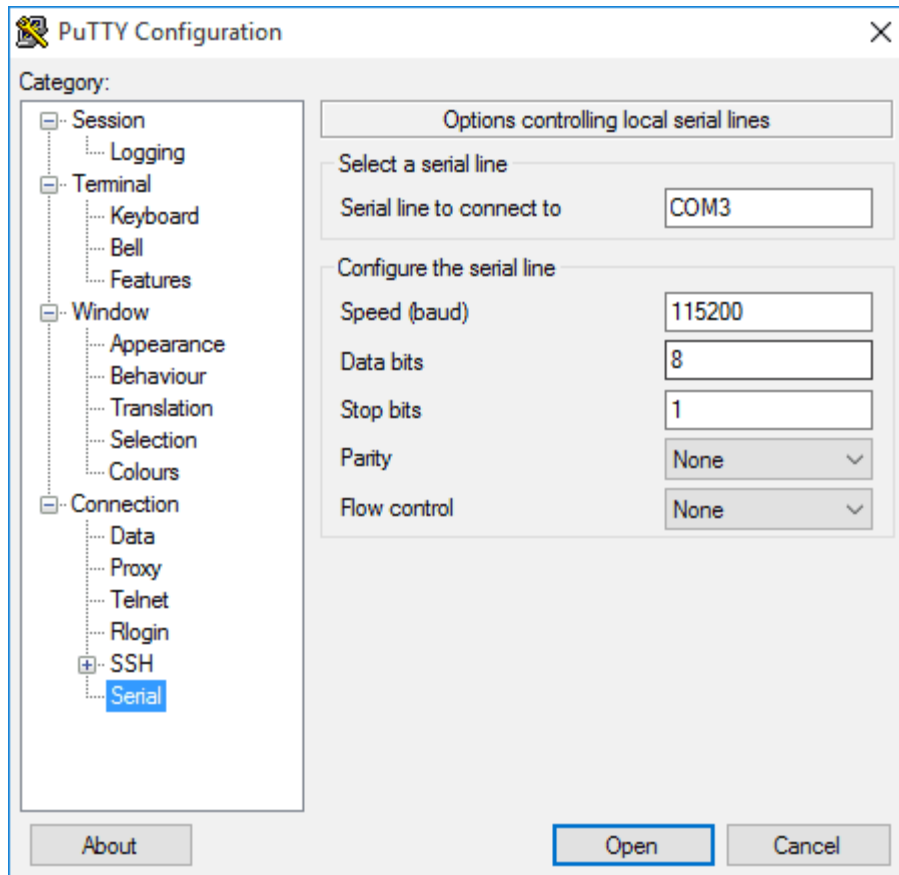




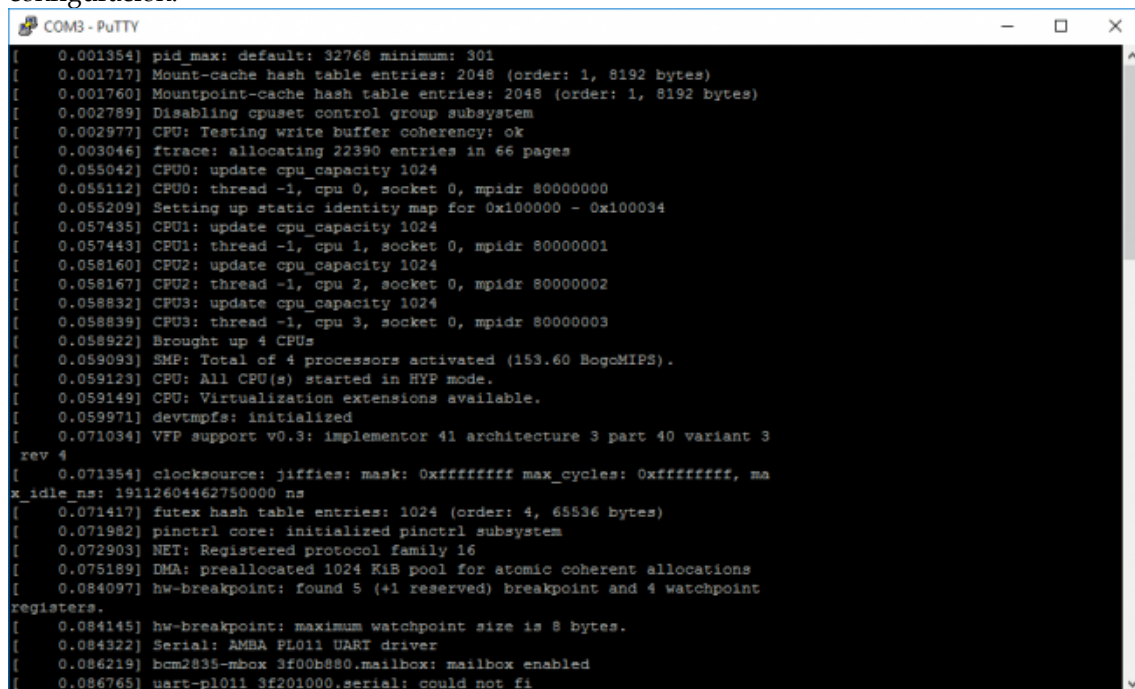
Una vez sabemos el Puerto COM, podemos abrir el cliente con el que nos comunicaremos. Como hemos dicho anteriormente, el más sencillo es PuTTY, el cual es gratuito y descargable [desde su web](#). Lo configuraremos de la siguiente manera, fijando la conexión Serie, una velocidad 115200, y el puerto COM:



Deberemos también desactivar el control de transmisión, en la pestaña Serie, buscamos “Flow Control” (control de flujo) y luego “None” (Ninguno) en la pestaña desplegable.



A continuación, le damos a “Open” para abrir la ventana del terminal. Ahora ya podemos enchufar la RPi a la alimentación, y en el ordenador nos saldrán una serie de líneas con la configuración.



```

[ 0.001354] pid_max: default: 32768 minimum: 301
[ 0.001717] Mount-cache hash table entries: 2048 (order: 1, 8192 bytes)
[ 0.001760] Mountpoint-cache hash table entries: 2048 (order: 1, 8192 bytes)
[ 0.002789] Disabling cpuset control group subsystem
[ 0.002977] CPU: Testing write buffer coherency: ok
[ 0.003046] ftrace: allocating 22390 entries in 66 pages
[ 0.055042] CPU0: update cpu_capacity 1024
[ 0.055112] CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
[ 0.055209] Setting up static identity map for 0x1000000 - 0x1000034
[ 0.057435] CPU1: update cpu_capacity 1024
[ 0.057443] CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
[ 0.058160] CPU2: update cpu_capacity 1024
[ 0.058167] CPU2: thread -1, cpu 2, socket 0, mpidr 80000002
[ 0.058832] CPU3: update cpu_capacity 1024
[ 0.058839] CPU3: thread -1, cpu 3, socket 0, mpidr 80000003
[ 0.058922] Brought up 4 CPUs
[ 0.059093] SMP: Total of 4 processors activated (153.60 BogoMIPS).
[ 0.059123] CPU: All CPU(s) started in HYP mode.
[ 0.059149] CPU: Virtualization extensions available.
[ 0.059971] devtmpfs: initialized
[ 0.071034] VFP support v0.3: implementor 41 architecture 3 part 40 variant 3
rev 4
[ 0.071354] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, ma
x_idle_ns: 19112604462750000 ns
[ 0.071417] futex hash table entries: 1024 (order: 4, 65536 bytes)
[ 0.071982] pinctrl core: initialized pinctrl subsystem
[ 0.072903] NET: Registered protocol family 16
[ 0.075189] DMA: preallocated 1024 KiB pool for atomic coherent allocations
[ 0.084097] hw-breakpoint: found 5 (+1 reserved) breakpoint and 4 watchpoint
registers.
[ 0.084145] hw-breakpoint: maximum watchpoint size is 8 bytes.
[ 0.084322] Serial: AMBA PL011 UART driver
[ 0.086219] bcm2835-mbox 3f00b880.mailbox: mailbox enabled
[ 0.086765] uart-pl011 3f201000.serial: could not fi
  
```

El primer arranque del sistema tarda unos minutos (hasta 5 minutos). Así que esperamos pacientemente hasta que se nos solicite iniciar sesión. Se debe ver algo así:

```
COM3 - PuTTY
Starting Configure Bluetooth Modems connected by UART...
Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
[ OK ] Started Avahi mDNS/DNS-SD Stack.
Starting Login Service...
Starting LSB: triggerhappy hotkey daemon...
Starting LSB: Autogenerate and use a swap file...
[ OK ] Started System Logging Service.
[ OK ] Started Restore Sound Card State.
[ OK ] Started Login Service.
[ OK ] Started LSB: triggerhappy hotkey daemon.
[ OK ] Started LSB: Autogenerate and use a swap file.
[ OK ] Started Configure Bluetooth Modems connected by UART.
Starting Load/Save RF Kill Switch Status of rfkill1...
Starting Bluetooth service...
[ OK ] Started Load/Save RF Kill Switch Status of rfkill1.
[ OK ] Started Bluetooth service.
[ OK ] Reached target Bluetooth.
Starting Hostname Service...
[ OK ] Started Hostname Service.
[ OK ] Started dhcpcd on all interfaces.
[ OK ] Reached target Network.
Starting /etc/rc.local Compatibility...
Starting Permit User Sessions...
[ OK ] Reached target Network is Online.
Starting LSB: Start NTP daemon...
[ OK ] Started Permit User Sessions.
[ OK ] Started /etc/rc.local Compatibility.
Starting Terminate Plymouth Boot Screen...
Starting Hold until boot process finishes up...

Raspbian GNU/Linux 8 raspberrypi ttyS0
raspberrypi login: 
```

Recuerda que al escribir la contraseña no aparecen caracteres ni asteriscos.

Después de iniciar sesión correctamente, deberíamos ver la vista de terminal "conocido de Linux", en la que ahora podemos trabajar.

```
COM3 - PuTTY
[ OK ] Started LSB: Autogenerate and use a swap file.
[ OK ] Started Configure Bluetooth Modems connected by UART.
Starting Load/Save RF Kill Switch Status of rfkill1...
Starting Bluetooth service...
[ OK ] Started Load/Save RF Kill Switch Status of rfkill1.
[ OK ] Started Bluetooth service.
[ OK ] Reached target Bluetooth.
Starting Hostname Service...
[ OK ] Started Hostname Service.
[ OK ] Started dhcpcd on all interfaces.
[ OK ] Reached target Network.
Starting /etc/rc.local Compatibility...
Starting Permit User Sessions...
[ OK ] Reached target Network is Online.
Starting LSB: Start NTP daemon...
[ OK ] Started Permit User Sessions.
[ OK ] Started /etc/rc.local Compatibility.
Starting Terminate Plymouth Boot Screen...
Starting Hold until boot process finishes up...

Raspbian GNU/Linux 8 raspberrypi ttyS0

raspberrypi login: pi
Password:
Last login: Wed Jul 5 11:46:31 UTC 2017 on ttyS0
Linux raspberrypi 4.9.35-v7+ #1014 SMP Fri Jun 30 14:47:43 BST 2017 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~$ 
```

Ahora ya tenemos disponible el mismo terminal de comandos que tendríamos disponible con una pantalla en nuestra RPi.

El sistema ya está instalado. Así que hagamos los ajustes más sencillos. Ingresamos el comando `sudo raspi-config` y hacemos clic en Enter. Aparecerá la ventana del programa de configuración preparado especialmente para la Raspberry Pi:

```
Raspberry Pi 3 Model B Rev 1.2

| Raspberry Pi Software Configuration Tool (raspi-config) |

1 Change User Password      Change password for the current u
2 Hostname                  Set the visible name for this Pi
3 Boot Options              Configure options for start-up
4 Localisation Options      Set up language and regional sett
5 Interfacing Options       Configure connections to peripher
6 Overclock                 Configure overclocking for your P
7 Advanced Options          Configure advanced settings
8 Update                   Update this tool to the latest ve
9 About raspi-config        Information about this configurat
```

## 10. Usar una cámara en RPi

¡Empezamos con la cámara en Raspberry! Gracias a ella es posible, entre otras cosas, hacer fotografías, transmitir la imagen por la red e incluso detectar movimiento. En el caso de los principiantes, muchos se preguntan como llevar a cabo esto ya que les puede parecer un nivel más complejo. Pero en este apartado vas a descubrir que no lo es.

Se pueden conectar muchas cámaras diferentes al RPi. Nos enfocamos en usar una cámara raspberry dedicada, es decir, Raspberry Pi Camera HD v2, la más extendida.

Primero, unas palabras sobre los parámetros técnicos de la cámara:

- Resolución máxima: 8 Mpx (3280x2464),
- Modos de video: 1080p30 (Full-HD), 720p60, 640x480p90 (VGA),
- Interfaz: CSI,
- Peso: 3 g,
- Dimensiones: 25 x 24 x 9 mm.



El módulo Raspberry Pi Camera HD utiliza CSI (Interfaz en Serie de Cámara), que está dedicado en especial para cámaras. Gracias a esto, todo el conjunto no carga el bus USB. Además, esta interfaz es compatible con un acelerador de gráficos (GPU) integrado en la Raspberry Pi.

## 10.1. Conexión de la cámara

Empezamos por conectar la cámara, que se limita solo a insertar la cinta:

- En la Raspberry Pi, enchufe la cinta en el conector descrito como CAMERA. La cinta debe mirar las clavijas plateadas hacia el conector HDMI.
- Coloque la cinta en la cámara de tal manera que los contactos plateados se dirijan hacia el objetivo.



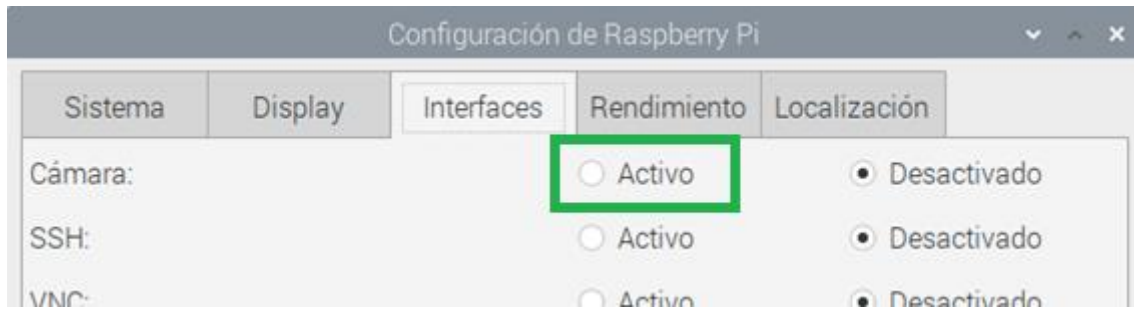
Presta atención únicamente a la posición de los contactos plateados indicados anteriormente, no al lado de la cinta donde este el texto. Cuando conectes la cinta, haz lo siguiente: levanta el bloqueo del conector (una especie de pletina), inserta la cinta y presiona hacia abajo el bloqueo del conector. No tienes que preocuparte por la posición de la cámara, la imagen se puede rotar mediante programación.



## 10.2. Hacer fotografías

Comenzamos con la función básica, que es tomar fotografías. Lo primero de todo, como siempre, iremos al menú de configuración y activaremos la cámara.





Para capturar las fotos, usaremos el programa **raspistill**. El uso básico del programa de captura de imágenes es sencillo. Simplemente escribiendo el comando:

```
raspistill -o test.jpg
```

En lugar de test.jpg podemos dar cualquier nombre de archivo. La foto se tomará exactamente 5 segundos después de dar el comando anterior. El archivo test.jpg se guardará en el directorio desde el que ejecutamos el programa raspistill.



Si tenemos un monitor conectado o usamos VNC para previsualizar la imagen, basta con buscar la foto y hacer clic dos veces en ella para ver la foto.

Si las fotos están borrosas, podemos cambiar el enfoque girando la lente de la cámara. La operación debe realizarse muy suavemente a mano o con una herramienta pequeña. ¡Tenga cuidado al ajustar, ya que es fácil rayar la lente!





### 10.3. Comandos

Algunos comandos útiles que vale la pena conocer son:

- **Desactivar vista previa.** Deshabilite la vista previa no deseada agregando la opción **-n** :  
`raspistill -n -o test.jpg`
- **Cambiar el temporizador de captura.** Se puede cambiar con la opción **-t** seguido del tiempo en milisegundos:  
`raspistill -n -o test.jpg -t 100`  
*Nota: para evitar problemas de calidad y demás se recomienda usar como mínimo 100ms.*
- **Rotación de imagen.** Basta con utilizar el parámetro **-rot** , que gira la imagen en el ángulo seleccionado, que será un múltiplo de 90 °:  
`raspistill -n -o test.jpg -t 100 -rot 180`
- **Reflexión vertical u horizontal.** Tenemos los comandos para reflexión horizontal ( **-hf** ) y verticalmente ( **-vf** ):  
`raspistill -n -o test.jpg -t 100 -hf`  
`raspistill -n -o test.jpg -t 100 -vf`
- **Cambiar la resolución.** No siempre nos interesa la resolución más alta, podemos ahorrar mucho espacio reduciendo el tamaño de la imagen descargada. Establecemos el ancho con la opción **-w** y el alto con **-h** . Una foto con resolución VGA estándar sería:  
`raspistill -n -o test_VGA.jpg -t 100 -w 640 -h 480`
- **Añadir fecha y hora al archivo.** El programa raspistill se puede ejecutar automáticamente, por ejemplo, periódicamente o cuando se detecta un evento específico. En ese caso, sería conveniente poder guardar varios archivos en lugar de sobrescribir el mismo test.jpg cada vez . El programa raspistill tiene dos opciones para usar la hora actual en el nombre del archivo.

La opción **-dt** agrega la fecha y hora actuales al nombre del archivo. Mientras que el parámetro **-ts** agrega el llamado marca de tiempo, que es el número de segundos desde el 1 de enero de 1970 (este es el método tradicional de representación de tiempo de Unix). Para utilizar estas posibilidades, en el nombre del archivo debemos insertar el marcador **%d** , que será reemplazado por la fecha en que se tomó la foto. Podemos emitir los siguientes dos comandos para la prueba en el día 20/10/2020:

```
raspistill -n -o test_%d.jpg -t 100 -dt #Se guardará como test_10202041040.jpg
raspistill -n -o test_%d.jpg -t 100 -ts #Se guardará como test_1603196029.jpg
```

- **Otros ajustes.** Existen múltiples comandos más. La información más importante se puede encontrar en la descripción que aparece tras llamar al comando **raspistill** sin ningún parámetro más, entonces aparecerá toda la información y comandos en un listado. Con los atributos que se describen allí, también puede cambiar parámetros como ISO, balance de blancos, brillo y contraste.

## 10.4. Transmisión de imágenes con Motion

Otra aplicación igualmente interesante para la cámara Raspberry Pi es el paquete Motion, que permite, entre otras cosas, transmitir la imagen a través de la red. Además, también se puede utilizar para rastrear el movimiento de objetos.

Empezamos instalando el programa introduciendo el comando:

```
sudo apt install motion
```

La versión instalada del paquete Motion por defecto no tiene soporte incorporado para nuestra cámara, por lo que necesitamos instalar el controlador apropiado ahora: Video4Linux (V4L) . Para ello, emitimos el siguiente comando, señalando que el penúltimo carácter es una "L" minúscula y no el número "1":

```
sudo modprobe bcm2835-v4l2
```

Ahora aparecerá un nuevo dispositivo en el directorio / dev , que corresponderá a la webcam: / dev / video0 .

Esta es la interfaz estándar utilizada por las cámaras en Linux.

El paquete Motion instalado tiene grandes posibilidades y muchas opciones disponibles. El archivo de configuración predeterminado es /etc/motion/motion.conf . El acceso al directorio / etc requiere derechos de administrador, así que edite el archivo con el comando:

```
sudo nano /etc/motion/motion.conf
```

Después de ejecutar el archivo, debemos realizar algunos cambios importantes. Primero, buscamos la línea donde está la opción **stream\_localhost**. Por defecto está encendido (on), lo cambiamos a apagado (off).

Puede usar el atajo **CTRL + W** para buscar texto en nano.

```

pi@raspberrypi: ~
GNU nano 2.7.4 File: /etc/motion/motion.conf Modified
# Quality of the jpeg (in percent) images produced (default: 50)
stream_quality 50

# Output frames at 1 fps when no motion is detected and increase to the
# rate given by stream_maxrate when motion is detected (default: off)
stream_motion off

# Maximum framerate for stream streams (default: 1)
stream_maxrate 1

# Restrict stream connections to localhost only (default: on)
stream_localhost off

# Limits the number of images per connection (default: 0 = unlimited)
# Number can be defined by multiplying actual stream rate by desired number of $
# Actual stream rate is the smallest of the numbers framerate and stream_maxrate
stream_limit 0

# Set the authentication method (default: 0)
# 0 = disabled

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

Guardamos el archivo y lo dejamos. Luego ejecutamos el paquete Motion con el comando:  
*sudo motion*

Cuando se inicia el programa, ejecutamos el navegador web en nuestro PC. En la barra de direcciones, ingresa la dirección IP con el puerto 8081 (por ejemplo, <http://192.168.0.250:8081/>). Después de un tiempo, deberíamos ver una pequeña transmisión de nuestra cámara web (1 FPS). Presiona CTRL + C para terminar el programa de la consola.

Para aumentar los FPS (Frames Per Second/fotogramas por segundo) podemos modificar las siguientes líneas que por defecto están establecidos en 20:

- framerate: ratio de frames

```

pi@raspberrypi: ~
GNU nano 2.7.4 File: /etc/motion/motion.conf Modified
# Image width (pixels). Valid range: Camera dependent, default: 352
width 640

# Image height (pixels). Valid range: Camera dependent, default: 288
height 480

# Maximum number of frames to be captured per second.
# Valid range: 2-100. Default: 100 (almost no limit).
framerate 20

# Minimum time in seconds between capturing picture frames from the camera.
# Default: 0 = disabled - the capture rate is given by the camera framerate.
# This option is used when you want to capture images at a rate lower than 2 $
minimum_frame_time 0

# URL to use if you are using a network camera, size will be autodetected (in$
[ Search Wrapped ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell

```

- stream\_maxrate: máximo ratio de frames en stream

```
pi@raspberrypi: ~
GNU nano 2.7.4 File: /etc/motion/motion.conf Modified
# Quality of the jpeg (in percent) images produced (default: 50)
stream_quality 50

# Output frames at 1 fps when no motion is detected and increase to the
# rate given by stream_maxrate when motion is detected (default: off)
stream_motion off

# Maximum framerate for stream streams (default: 1)
stream_maxrate 20

# Restrict stream connections to localhost only (default: on)
stream_localhost off

# Limits the number of images per connection (default: 0 = unlimited)
# Number can be defined by multiplying actual stream rate by desired number of
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell
```

Para aumentar la **resolución**:

- width: ancho (lo configuramos en 640),
- high: altura (lo configuramos en 480).

```
pi@raspberrypi: ~
GNU nano 2.7.4 File: /etc/motion/motion.conf Modified
# 2 : 60hz
# 3 : Auto
power_line_frequency -1

# Rotate image this number of degrees. The rotation affects all saved images $
# well as movies. Valid values: 0 (default = no rotation), 90, 180 and 270.
rotate 180

# Image width (pixels). Valid range: Camera dependent, default: 352
width 640

# Image height (pixels). Valid range: Camera dependent, default: 288
height 480

# Maximum number of frames to be captured per second.
# Valid range: 2-100. Default: 100 (almost no limit).
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell
```

## 10.5. Detección de movimiento en RPi

Como podemos ver, convertimos la Raspberry Pi en una webcam. Vale la pena recordar que el paquete Motion supera con creces las capacidades de las cámaras típicas. Te permite detectar movimiento, generar eventos después de detectar un intruso, tomar fotos o videos automáticamente, notificar eventos, etc.

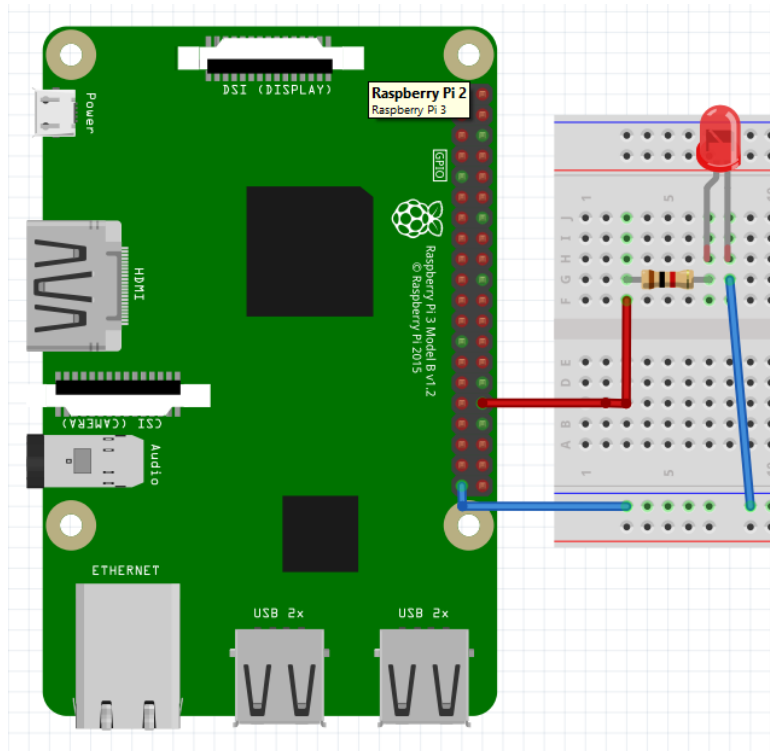
Puedes encontrar más información sobre el programa en el sitio web del proyecto: <https://motion-project.github.io/>

Para esta prueba, puedes ejecutar, por ejemplo, la opción responsable del seguimiento de movimiento. Para ello, basta con habilitar la opción en el archivo de configuración, que se llama **locate\_motion\_mode** (cambiamos de apagado(off) a encendido(on)). Los elementos en movimiento ahora se marcarán con un rectángulo en la imagen transmitida. Para que el elemento se delimite con un rectángulo rojo, **locate\_motion\_style** también debe cambiarse a redbox .

## 11. Señal PWM en Python

Una señal PWM es una señal generada por Modulación de Ancho de Pulso (Pulse Width Modulation), para más información puedes ir a nuestro [artículo del blog](#). En resumidas cuentas, y como hemos mencionado en el apartado de pines, la modulación PWM es usada para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga. En esta ocasión nos ocuparemos de generar PWM, manejar entradas y cámaras desde el nivel Python.

Vamos a comenzar con un ejemplo simple con un **diodo**. Conéctalo a través de una resistencia de  $1k\Omega$  al pin número 12, tal y como hemos visto antes y como se muestra en el diagrama a continuación:



Ahora, abriremos un IDE para programar en Python, como por ejemplo Thonny Python IDE, en el cual abriremos un nuevo sketch y copiaremos el siguiente código:

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(12, GPIO.OUT)

diodo = GPIO.PWM(12, 50) #Nueva instancia de PWM
duty = 0 # Ciclo de trabajo de la señal PWM
diodo.start(duty) # Activa la señal PWM
```

```
try:
    while True:
        duty+= 5
        if duty > 100:
            duty = 0
            diodo.ChangeDutyCycle(duty) #Establece un nuevo valor para el relleno
            time.sleep(0.05)
except KeyboardInterrupt:
    print('Fin')

diodo.stop()
GPIO.cleanup()
```

Vamos a explicar un poco cada línea. Como siempre, al principio del todo hay que importar las librerías correspondientes e iniciar las configuraciones, en nuestro caso las librerías de los pines GPIO y del tiempo y sus configuraciones.

Cuando queremos generar una señal PWM, tenemos que comenzar creando una instancia a la que asignaremos información sobre el pin y la frecuencia. Hacemos esto con la siguiente línea:

```
Nombre_instancia = GPIO.PWM (pin, frecuencia) #Nueva instancia de PWM
```

Recuerda que lo que haya después de un '#' es un comentario y no afecta al código. En nuestro caso la línea queda:

```
diodo = GPIO.PWM (12, 50) #Nueva instancia de PWM
```

Luego creamos una variable "relleno" llamada duty, que como su nombre indica, contendrá información sobre el llenado actual de la señal PWM (valores de 0 a 100 % sin signo), será el ciclo de trabajo de la señal. En resumidas cuentas, el porcentaje que la señal que está en estado alto, respecto al total de un ciclo. Esta variable duty la inicializaremos a 0.

```
duty = 0 # Ciclo de trabajo de la señal PWM
```

El siguiente paso es iniciar PWM, como argumento de esta función damos la terminación esperada.

```
diodo.start (duty) # Activa la señal PWM
```

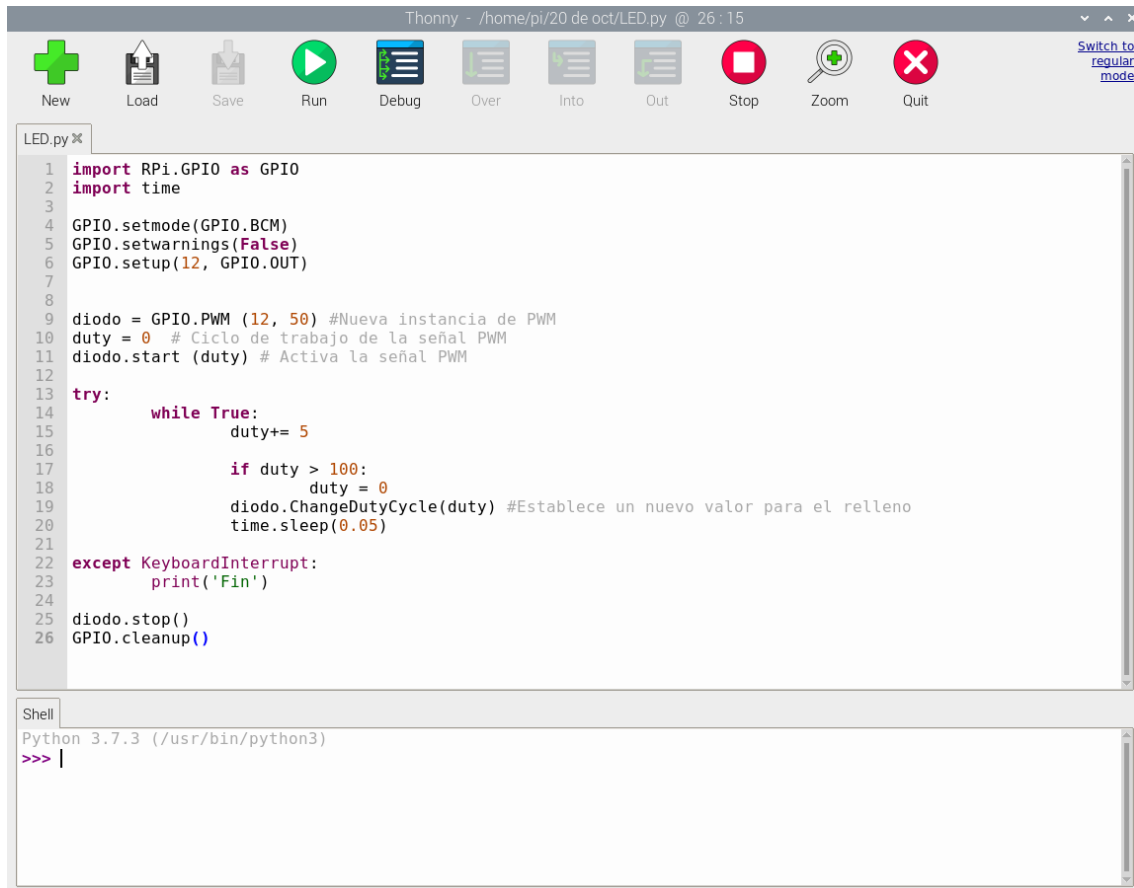
A partir de ahora, la señal PWM se generará en el pin 12. Su duty se puede cambiar todo el tiempo usando la siguiente instrucción:

```
diodo.ChangeDutyCycle(duty)
```

Entonces, al juntar la información de las partes anteriores del curso, podemos crear un programa simple que cambiará el PWM completando un ciclo. En la práctica, esto significará una iluminación suave del LED.

Hay también un `time.sleep(0.05)` el cual hace una espera dentro del bucle while, aunque sea mínima, esto siempre es necesario. Al final del programa hay una línea más `diodo.stop ()`, que probablemente no necesita explicarse. Por supuesto, es responsable de detener el "generador" PWM en un pin determinado. Mientras se ejecuta el programa, también podemos ajustar la frecuencia PWM. Para hacer esto, debemos introducir la siguiente instrucción: `diodo.ChangeDutyCycle (XXX)`, donde dentro del paréntesis puedes poner un número del 0 al 100.





Thonny - /home/pi/20 de oct/LED.py @ 26:15

New Load Save Run Debug Over Into Out Stop Zoom Quit

LED.py

```

1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setwarnings(False)
6 GPIO.setup(12, GPIO.OUT)
7
8
9 diodo = GPIO.PWM (12, 50) #Nueva instancia de PWM
10 duty = 0 # Ciclo de trabajo de la señal PWM
11 diodo.start (duty) # Activa la señal PWM
12
13 try:
14     while True:
15         duty+= 5
16
17         if duty > 100:
18             duty = 0
19             diodo.ChangeDutyCycle(duty) #Establece un nuevo valor para el relleno
20             time.sleep(0.05)
21
22 except KeyboardInterrupt:
23     print('Fin')
24
25 diodo.stop()
26 GPIO.cleanup()

```

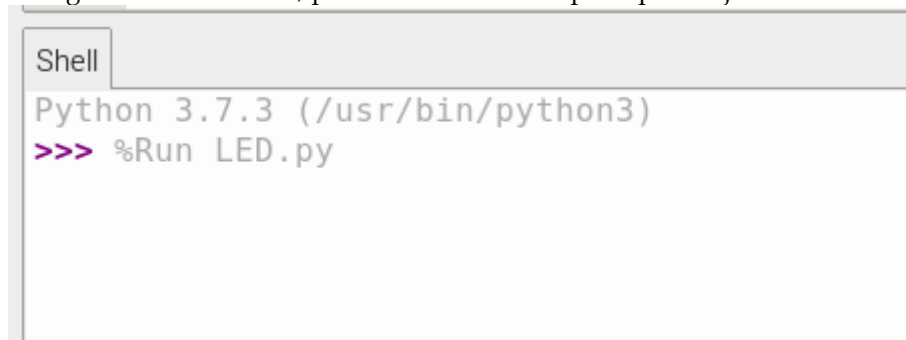
Shell

```

Python 3.7.3 (/usr/bin/python3)
>>> |

```

Como vemos, la tabulación de cada línea es importante, y debe quedar como la imagen anterior. Una vez lo tengamos de ese modo, podemos darle a Run para que se ejecute.



Shell

```

Python 3.7.3 (/usr/bin/python3)
>>> %Run LED.py

```

Verás como se va encendiendo el LED gradualmente hasta el máximo, entonces se apaga y repite el proceso. Para parar el programa puedes hacer CTRL+C o darle al botón Stop. Recuerda que si el LED se queda encendido, siempre puedes apagarlo manualmente con el terminal de comandos usando:

```
gpio -g write 12 0
```

## 12. Resumen

El objetivo del curso era cubrir completamente los conceptos básicos de la Raspberry Pi. Hemos mostrado cómo instalar el sistema operativo, configurarlo, usar la cámara y controlar los puertos GPIO. Desde el principio, asumimos que RPi puede ser utilizada por todos, incluso por alguien que nunca había tenido contacto con Linux. Por lo tanto, a lo largo de la guía incluimos contenido sobre cómo trabajar con este sistema Linux, así como con la consola y las diferentes formas posibles.

Para profundizar más sobre la Raspberry Pi, habrá más artículos, cursos y ejemplos, así como una enorme cantidad de información en la red. Te invitamos también a leer nuestro artículo de The MagPi para descubrir esta maravillosa revista oficial.