

# **Curso básico-intermedio de programación en Raspberry Pi**

Curso intersemestral

M.I. Omar Arturo Castillo Méndez  
7 de agosto de 2024

# Contenido

## Introducción

- Generalidades y requisitos

## Configuración

- Instalación de Raspbian

- Comandos básicos

- Ajustes de tarjeta RPi

- Definición de IP estática

- Conexión remota con SSH

- Actualización de paquetes

## Programación en RPi

- GPIO

- Ambiente virtual venv

- Sensor MLX90614

- Pandas



# Introducción

## Examen de diagnostico

- ¿Qué es una tarjeta Raspberry Pi?
- Si conoces algún lenguaje de programación, ¿Cuál o cuáles?
- ¿Qué es el lenguaje de programación Python?
- ¿Qué bibliotecas de Python conoces?
- En la configuración de puertos de una RPi, ¿Qué diferencia hay entre el BCM y BOARD?
- Escriba un código para encender un LED en una tarjeta Raspberry Pi usando la configuración de puertos BOARD.
- ¿Cuáles son tus expectativas del curso?



# Introducción

## ¿Qué es una Raspberry Pi?

Es una tarjeta de desarrollo, diseñada como una computadora modular con una arquitectura ARM y usando un sistema operativo basado en Linux (Raspbian OS).

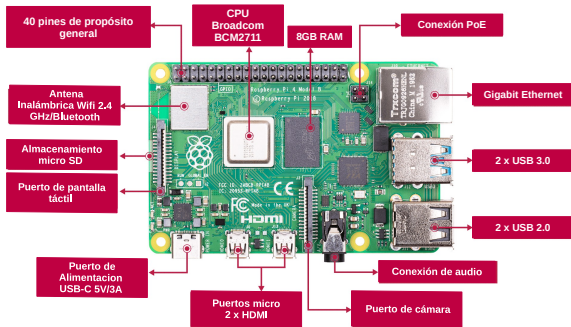


Figura 1: Tarjeta Raspberry Pi 4



# Configuración inicial

## Requisitos

### Material necesario:

- Tarjeta Raspberry. Pi(Cualquier versión)
- Fuente de alimentación de 5V a 3A.
- Tarjeta de almacenamiento micro SD de 64 Gb o superior clase 10 o superior.
- Mouse y teclado.



# Configuración inicial

## Instalación de Raspbian

### Instrucciones:

- Visitar la siguiente liga:  
<https://www.raspberrypi.com/software/>
- Descargar la versión correspondiente con su respectivo sistema operativo, y dar click en **descargar**
- Una vez instalado se mostrará la siguiente interfaz de Raspberry Pi Imager.



# Configuración inicial

## Instalación de Raspbian



Figura 2: Interfaz: Raspberry Pi Imager



# Configuración inicial

## Selección de Tarjeta

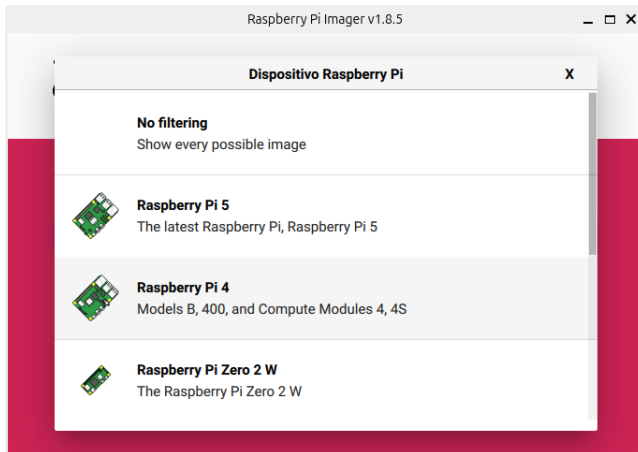


Figura 3: Selección de tarjeta Raspberry Pi





# Configuración inicial

## Selección de sistema operativo

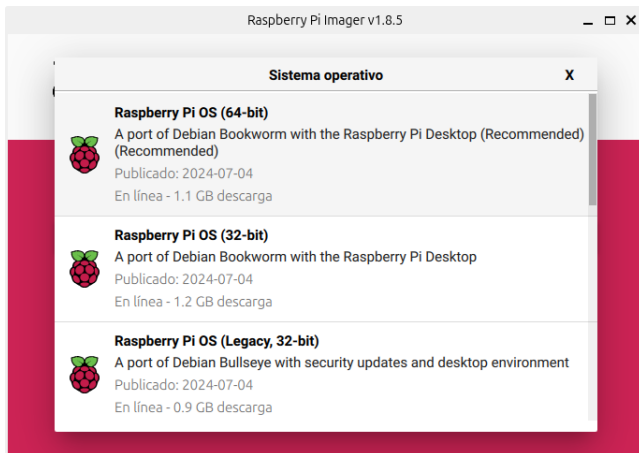


Figura 4: Selección de sistema operativo



# Configuración inicial

## Selección de unidad de almacenamiento



Figura 5: Selección de unidad de almacenamiento



# Configuración inicial

## Comandos básicos de la terminal

### Instrucciones:

- `sudo` : Comando para acceder a derechos de superusuario.
- `cd` : Cambiar de directorio a una ruta especificada.
- `cd ..` : Subir un nivel en la ruta.
- `ls` : Listar archivos de un fichero/carpeta.
- `pwd` : Mostrar la ruta del carpeta.
- `mkdir` : Crear una carpeta.
- `nano` : Editor de texto desde la terminal.
- `cp` : Copiar fichero o archivo hacia una ruta especificada.



# Configuración inicial

## Comandos básicos de la terminal

### Instrucciones:

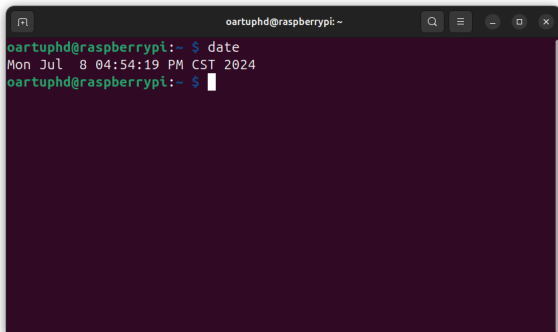
- `pwd` : Muestra la ruta de la ubicación de trabajo.
- `ifconfig` : Consultar la información de las interfaces de red.
- `sudo raspi-config` : Entrar a la configuración de la Raspberry Pi
- `pinout` : Muestra la distribución de pines de la tarjeta.



# Configuración inicial

## Cambiar la configuración de la RPi

Una vez terminada la instalación, se abre una terminal y se ejecuta el siguiente comando date:

A terminal window titled 'oartuphd@raspberrypi: ~' with standard window controls. The prompt is 'oartuphd@raspberrypi:~ \$'. The command 'date' has been entered and executed, resulting in the output 'Mon Jul 8 04:54:19 PM CST 2024'. The prompt is now 'oartuphd@raspberrypi:~ \$' with a cursor.

```
oartuphd@raspberrypi:~ $ date
Mon Jul 8 04:54:19 PM CST 2024
oartuphd@raspberrypi:~ $
```

Figura 6: Consulta la fecha del sistema

Para colocar la fecha actual usar en la terminal de la RPi: **sudo date -s 'YYYY-MM-DD HH:MM:SS'**.



# Configuración inicial

## Configuración de interfaces

Se ejecutará el siguiente comando: `sudo raspi-config`

Elegir la opción de **Interface Options**

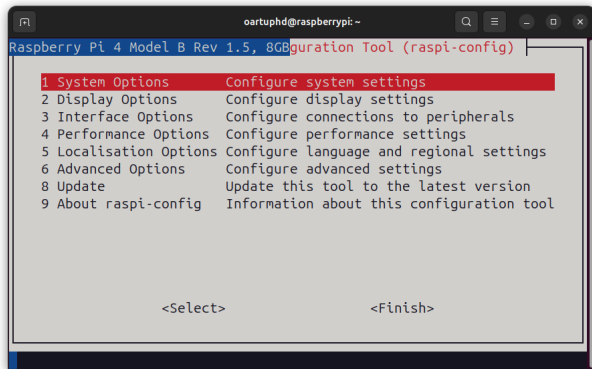


Figura 7: Opciones de configuración de la RPi



# Configuración inicial

## Configuración de interfaces

Selecciona una por una las interfaces de SSH, SPI, I2C, Serial Port, 1-Wire.

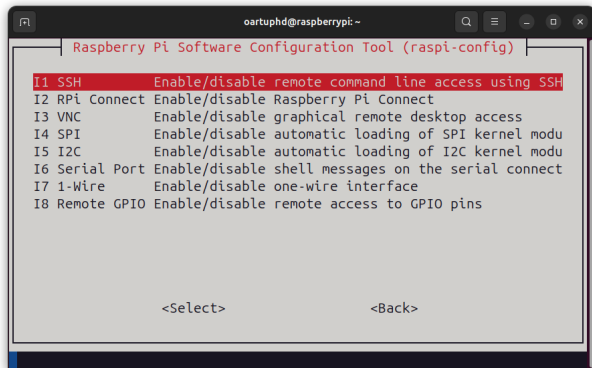


Figura 8: Selección de interfaces de la tarjeta de RPi.



# Configuración inicial

## Definir una IP estática

Para poder trabajar de manera remota sin la necesidad de usar un monitor externo mediante una terminal SSH(Secure Shell). Ejecutamos el siguiente comando en la terminal "**sudo nmtui**"

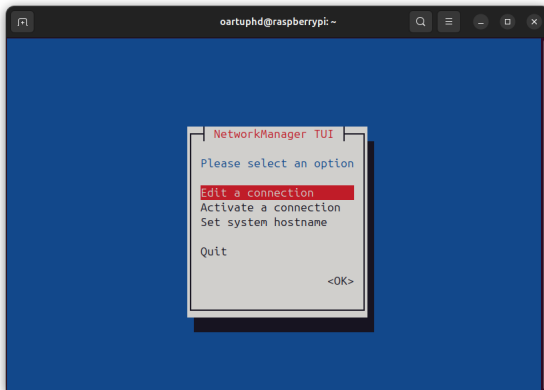


Figura 9: Administrador configuración de red.





# Configuración inicial

## Definir una IP estática

Seleccionar el tipo de interfaz a convenir, si esta conectada vía Ethernet o Wi-Fi:

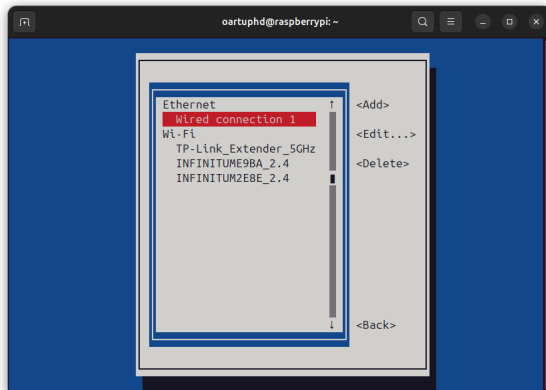


Figura 10: Seleccionar la interfaz a modificar



# Configuración inicial

## Definir una IP estática

En el apartado de IPv4 es donde se va a configurar la IP estática.

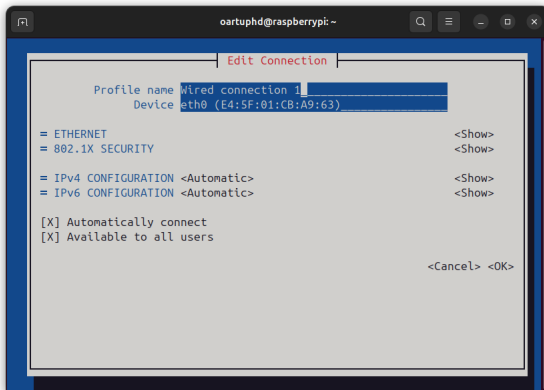


Figura 11: Se seleccionó una conexión via Ethernet



# Configuración inicial

## Definir una IP estática

Para consultar la IP a la cual esta conectada la RPi, vamos a salir del menú anterior y ejecutar en la terminal el siguiente comando: **"ifconfig"**

```
oartuphd@raspberrypi: ~  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.255.255.0  
    inet6 ::1 netmask ffffffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff  
    loop 0  
    RX packets 112777 bytes 9752 (9.5 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 112 bytes 9752 (9.5 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.7 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 fe80::208:1fff:fe00:4956 prefixlen 64 scopeid 0x0<global>  
    ether 08:00:27:00:00:00 txqueuelen 1000 (Ethernet)  
    RX packets 1916 bytes 352111 (343.8 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 1209 bytes 186032 (181.6 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
oartuphd@raspberrypi:~$
```

Figura 12: Información de la red mediante el comando ifconfig



# Configuración inicial

## Definir una IP estática

Una vez anotada la información anterior acceder el Network manager de la RPi, y capturar en: **Address, Gateway y DNS Server**

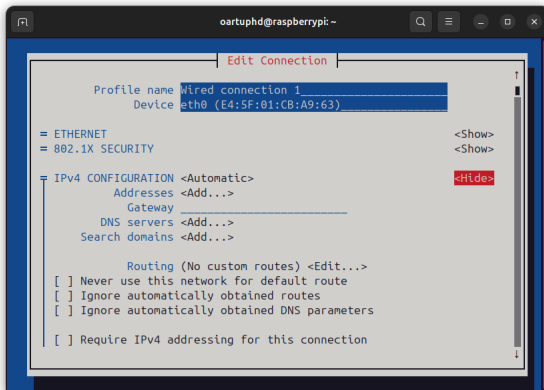


Figura 13: Network manager de la RPi



# Configuración inicial

## Conexion SSH con Windows

Para crear la conexión SSH con Windows se utilizará el siguiente programa PuTTY:

### Instrucciones:

- Visitar la siguiente liga: <https://putty.org/>
- Descargar la versión correspondiente con su respectivo sistema operativo.
- Seguir los pasos de instalación.



# Configuración inicial

## Conexión SSH con Windows

Terminada la instalación ubicar el programa PuTTY y ejecutarlo a continuación mostrará lo siguiente:

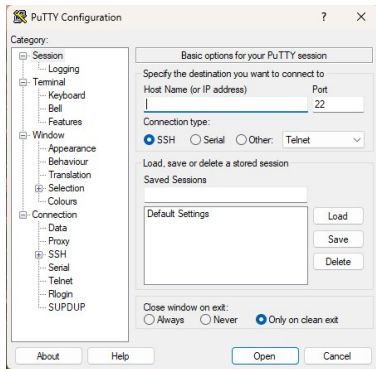


Figura 14: Configuración de PuTTY

En el campo de **Host Name**, introducir la IP que previamente se fijo en la RPi.



# Configuración inicial

## Conexión SSH con Windows

Cuando se abra la terminal capturar el **usuario** que definieron al inicio de la instalación en Raspbian y su respectiva contraseña:

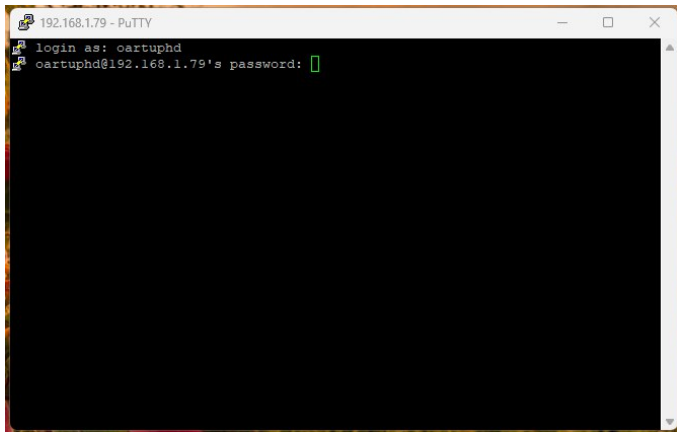


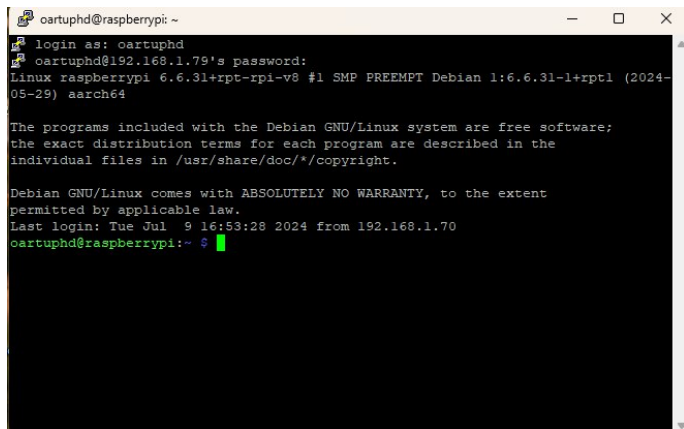
Figura 15: Captura de usuario y contraseña para acceso a la RPi



# Configuración inicial

## Conexión SSH con Windows

Posteriormente aparecerá una advertencia de conexión aceptarla y finalmente aparecerá la terminal de la RPi:



```
oartuphd@raspberrypi: ~
login as: oartuphd
oartuphd@192.168.1.79's password:
Linux raspberrypi 6.6.31+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.31-1+rpt1 (2024-05-29) aarch64

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: Tue Jul 9 16:53:28 2024 from 192.168.1.70
oartuphd@raspberrypi:~ $
```

Figura 16: Terminal de la RPi





# Configuración inicial

## Actualización de paquetes

A continuación vamos actualizar todo el software instalado a la última versión usando el administrador de paquetes **apt** (Advanced Packaging Tool)

### Comandos de actualización:

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



# Programación en RPi

## Creación de un archivo

Para la creación de un archivo desde la terminal se tomaran los siguientes pasos:

### Pasos a seguir:

- Definir la ubicación donde estará la carpeta de trabajo (*cd ~/*).
- Crear la carpeta con el nombre a conveniencia (*mkdir nombre-de-la-carpeta*).
- Usar el gestor de texto de la terminal para crear un archivo con extensión de Python(*nano ejemplo.py*).



# Programación en RPi

## Creación de un archivo

Con las instrucciones anteriores la terminal mostrará la siguiente ventana:

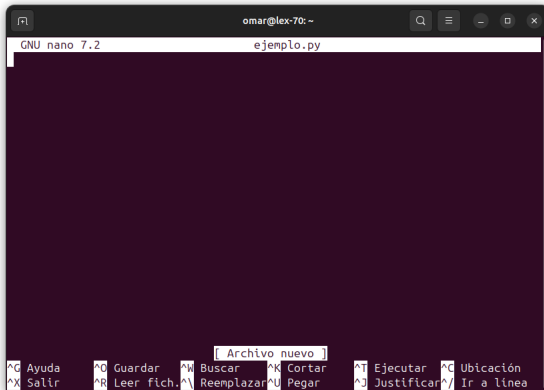


Figura 17: Editor de texto nano.



# Programación en RPi

## Actividad 1:

### Creación de un archivo de texto

- Crear una carpeta en la dirección del *home* con sus iniciales **abcd**, dentro de esa misma agregar otra carpeta con el nombre : **dia1**
- Dentro de la carpeta **dia1**, crear un archivo con la extensión *.txt*
- Con el editor de texto *nano*, escribir una breve descripción de los comandos que se han utilizado hasta el momento.



# Programación en RPi

## ¿Qué es el GPIO?

El GPIO o *Entrada/Salidas de propósito general* es un pin genérico en un chip. En este caso la RPi cuenta con dos configuraciones de GPIO:

### GPIO

- GPIO.BOARD: Este indica a los pines según su numeración.
- GPIO.BCM: se refiere a los pines por su numero de *Broadcom SOC channel*.

Para mostrar los pines en la RPi se puede usar el comando *pinout* en la terminal.



# Programación en RPi

## GPIO.BOARD

3v3 Power	1		2	5v Power
GPIO 2 (I2C1 SDA)	3		4	5v Power
GPIO 3 (I2C1 SCL)	5		6	Ground
GPIO 4 (GCLK0)	7		8	GPIO 14 (UART TX)
Ground	9		10	GPIO 15 (UART RX)
GPIO 17	11		12	GPIO 18 (PCM CLK)
GPIO 27	13		14	Ground
GPIO 22	15		16	GPIO 23
3v3 Power	17		18	GPIO 24
GPIO 10 (SPI0 MOSI)	19		20	Ground
GPIO 9 (SPI0 MISO)	21		22	GPIO 25
GPIO 11 (SPI0 SCLK)	23		24	GPIO 8 (SPI0 CE0)
Ground	25		26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27		28	GPIO 1 (EEPROM SCL)
GPIO 5	29		30	Ground
GPIO 6	31		32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33		34	Ground
GPIO 19 (PCM FS)	35		36	GPIO 16
GPIO 26	37		38	GPIO 20 (PCM DIN)
Ground	39		40	GPIO 21 (PCM DOUT)

Figura 18: Enumeración de pines según BOARD



# Programación en RPi

## GPIO.BCM

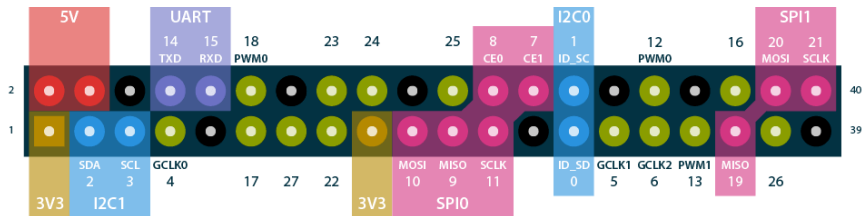


Figura 19: Enumeración de pines según BCM



# Programación en RPi

## Uso del GPIO

Para el uso del GPIO usando Python vamos a programar lo siguiente:

### Encender un LED

- Se va encender y apagar un led con un retardo de 1 segundo.
- Uso de la biblioteca **time** y **GPIO.BOARD**





# Programación en RPi

## Uso del GPIO



The image shows a terminal window titled 'omar@lex-70: ~/rpi-prog' with a search icon, a menu icon, and window control buttons. The terminal is running GNU nano 7.2 editing a file named 'actividad2.py \*'. The code inside the file is as follows:

```
import RPi.GPIO as GPIO
import time

pin = 10

GPIO.setmode(GPIO.BOARD)
GPIO.setup(pin, GPIO.OUT)
while True :
    GPIO.output(pin, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(pin, GPIO.LOW)
    time.sleep(1)
```

At the bottom of the terminal, there is a status bar with keyboard shortcuts for various actions:

^G Ayuda	^O Guardar	^W Buscar	^K Cortar	^T Ejecutar	^C Ubicación
^X Salir	^R Leer fich.	^_ Reemplazar	^U Pegar	^J Justificar	^_ Ir a línea

Figura 20: Programa para encender un LED



# Programación en RPi

## Actividad 2:

### Activación del GPIO

- Crear un programa para controlar 3 LED.
- Con un retardo de 2 segundos entre cada LED.
- Usar la configuración del GPIO.BOARD.
- El archivo debe estar guardado en la carpeta que se creo anteriormente.



# Programación en RPi

## Instalación de venv

El modulo venv, admite la creación de entornos virtuales, sobre una instalación de Python existente y queda aislada para administrar de mejor manera la gestión de los paquetes que se instalan y evitar dañar la configuración actual.

### Instalación de venv

- `sudo apt install python3-venv`



# Programación en RPi

## Configuración de venv

### Instalación de venv

- Crear una carpeta donde ubicar el ambiente virtual
- Ejecutar para crear el ambiente virtual `python -m venv nombred carpeta`
- Para activarlo `source nombred carpeta /bin/activate`
- Para instalar un paquete `nombred carpeta/bin/pip install nombred biblioteca`



# Programación en RPi

## Actividad 3:

### Instalacion de venv

- Crear una carpeta dentro de **Documentos** con sus iniciales **abcd**
- Una vez creada su carpeta, dentro crear una nueva con el nombre de **virtual**
- Activar el ambiente venv en la terminal.
- Una vez activado, crear un archivo de texto **.txt** con los comandos utilizados al mismo nivel de su carpeta **abcd**



# Programación en RPi

## Sensor de temperatura: MLX90614

### Instalación de MLX90614

- Visitar la dirección  
`https://pypi.org/project/PyMLX90614/`
- Instalar mediante pip con el ambiente virtual.



# Programación en RPi

## Sensor de temperatura: MLX90614

### Programa básico de MLX90614

```
1      from smbus2 import SMBus
2      from mlx90614 import MLX90614
3
4      bus = SMBus(1)
5      sensor = MLX90614(bus, address=0x5A)
6      print sensor.get_amb_temp()
7      print sensor.get_obj_temp()
8      bus.close()
```



# Programación en RPi

## Actividad 4:

### Requisitos de modificación

- El programa debe estar mostrando en la terminal de manera continua el valor de la temperatura.
- Debe haber un intervalo de mínimo de espera un 1 segundo por cada valor mostrado.
- Usar el programa debe contener comentarios con la descripción de como es la estructura del programa.





# Programación en RPi

## Pandas

Es una biblioteca de código abierto para estructura y análisis de datos programada en Python.

Pandas provee dos tipos de clases de manejo de datos:

### Pandas:

- **Series:** Es un arreglo unidimensional para contener datos de cualquier tipo.
- **DataFrame:** Es una estructura bidimensional que contiene datos en forma de tabla con filas y columnas.



# Programación en RPi

## Pandas: Tipos de datos

En Python se pueden manejar varios tipos de datos como los que se muestran:

### Tipo de datos:

- **Entero**(int): 2, 30, 500
- **Flotante**(float): 0.25, 58.36, 100.0
- **Cadena**(str): "Juan", "Hello", "jhgsd"
- **Booleano**(bool): Verdadero(True) o Falso(False)



# Programación en RPi

## Instalación de pandas

### Instalación de pandas en RPi:

```
sudo apt install python3-pandas
```

### Como importar pandas:

Desde la consola de python escribir el siguiente comando:

```
import pandas
```



# Programacion en RPi

## Pandas: Series

### Ejemplo de Series:

```
1      import pandas as pd
2      vector =
          pd.Series([1.3,0.4,'NaN',1000,
                     'tex'])
3      print(vector)
```



# Programación en RPi

## Actividad 5:

### Uso de Series

- Crear un archivo de Python capturando los siguientes datos: 125, Curso, 0.25, Hola, 872354, True, 0.2554
- Escriba una breve descripción dentro del archivo de python



# Programación de RPi

## Pandas: DataFrame

### Ejemplo de DataFrame:

```
1      import pandas as pd
2      matriz = pd.DataFrame( {
3          "A": 1.0,
4          "B": pd.Timestamp("20240101"),
5          "C": pd.Series([1,2,3,4]),
6          "F": "dato",
7      })
8      print(matriz)
```



# Programación en RPi

## PWM

La modulación por ancho de pulso o PWM, se usa para controlar el ancho de una señal digital con el propósito de controlar a su vez la potencia que se entrega. Modificando el ancho del pulso activo.

### Comandos para RPi

- Se usará la biblioteca gpiozero basado en Python.
- Esta genera una señal PWM por software.
- Los pines 0 y 1 están restringidos para uso de PWM.

