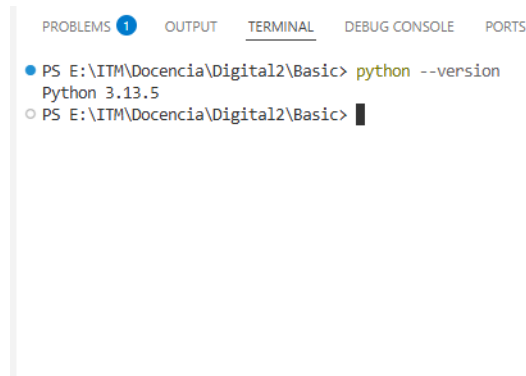


## TUTORIAL PARA LA CONEXIÓN Y PRUEBA DE SU ESP32

Este tutorial se realizará basado en que usted ya tiene instalado VS code, si no es así, deténgase a realizar la instalación de Python y de VS code en su equipo.

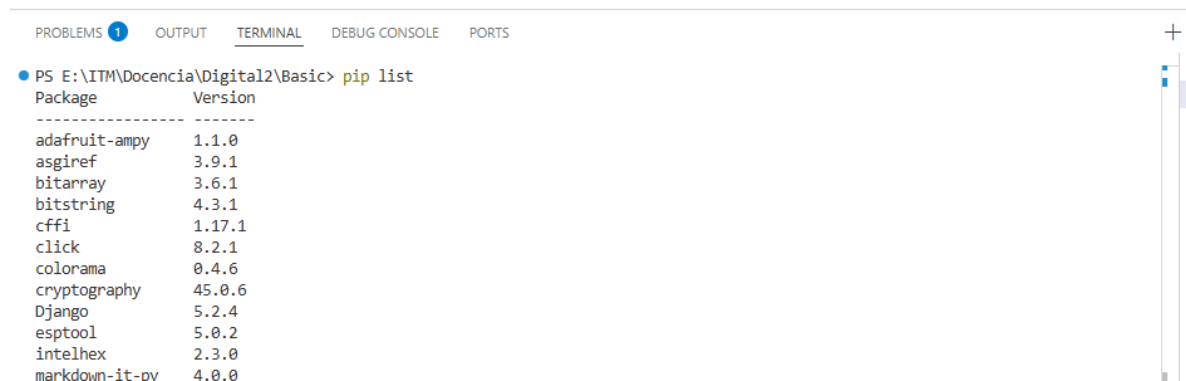
Para verificar que su visual studio pueda leer Python de la forma adecuada, ejecute en el terminal de VS Code el comando `python --version` o `python -V`.

Para sacar la terminal puede dar `ctrl+ñ` o ir a `view --> terminal`



```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE PORTS
● PS E:\ITM\Docencia\Digital2\Basic> python --version
Python 3.13.5
○ PS E:\ITM\Docencia\Digital2\Basic> █
```

Verifique que tenga instalado pyserial y esptool y paquetes de Python que pueda necesitar. Para la instalación de paquetes de Python se hace uso de pip. Inicialmente vamos a verificar los paquetes pre-instalados, para esto haga uso de `pip list`



```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE PORTS
● PS E:\ITM\Docencia\Digital2\Basic> pip list
Package Version
-----
adafruit-ampy 1.1.0
asgiref 3.9.1
bitarray 3.6.1
bitstring 4.3.1
cffi 1.17.1
click 8.2.1
colorama 0.4.6
cryptography 45.0.6
Django 5.2.4
esptool 5.0.2
intelhex 2.3.0
markdown-it-pv 4.0.0
```

Ahora, utilice `pip install` para instalar los paquetes necesarios, empezaremos con `pyserial` y `esptool`:

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE PORTS
typing_extensions 4.14.1
tzdata 2025.2
PS E:\ITM\Docencia\Digital2\Basic> pip install esptool
Requirement already satisfied: esptool in c:\users\yomin\appdata\local\programs\python\python313\lib\site-packages (5.0.2)
Requirement already satisfied: bitstring!=4.2.0,>=3.1.6 in c:\users\yomin\appdata\local\programs\python\python313\lib\site-packages (from esptool) (4.3.1)
Requirement already satisfied: cryptography>=43.0.0 in c:\users\yomin\appdata\local\programs\python\python313\lib\site-packages (from esptool) (45.0.6)
Requirement already satisfied: pyserial>=3.3 in c:\users\yomin\appdata\local\programs\python\python313\lib\site-packages (from esptool) (3.5)
Requirement already satisfied: reedsolo<1.8,>=1.5.3 in c:\users\yomin\appdata\local\programs\python\python313\lib\site-packages (from esptool) (1.7.0)
Requirement already satisfied: PyYAML>=5.1 in c:\users\yomin\appdata\local\programs\python\python313\lib\site-packages (from esptool) (6.0.2)
Requirement already satisfied: intelhex in c:\users\yomin\appdata\local\programs\python\python313\lib\site-packages (from esptool)
```

Haga el mismo procedimiento con pyserial.

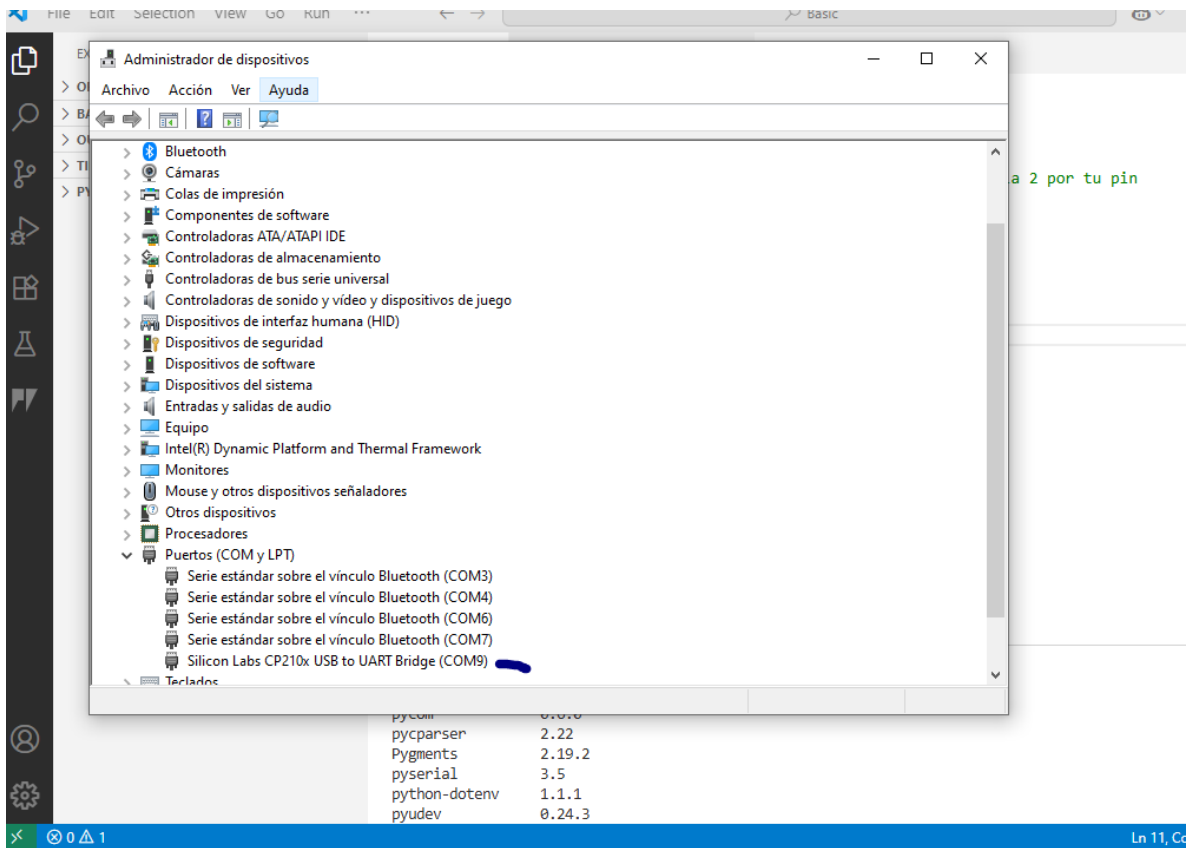
Despues de realizar este proceso, debería verlos a ambos listados al hacer pip list

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE PORTS
bitstring 4.3.1
cffi 1.17.1
click 8.2.1
colorama 0.4.6
cryptography 45.0.6
Django 5.2.4
esptool 5.0.2
intelhex 2.3.0
markdown-it-py 4.0.0
mdurl 0.1.2
pillow 11.3.0
pip 25.2
pycom 0.6.0
pycparser 2.22
Pygments 2.19.2
pyserial 3.5
python-dotenv 1.1.1
pyudev 0.24.3
PyYAML 6.0.2
pyzmq 27.0.1
reedsolo 1.7.0
rich 14.1.0
rich-click 1.8.9
rshell 0.0.36
six 1.17.0
```

De ahora en adelante, así instalará los paquetes de Python que necesite.

Para este paso, se asume que ya usted a pre-instalado el driver para la lectura del ESP32 en su computador, si no lo ha hecho, dicho driver esta disponible en: <https://www.silabs.com/software-and-tools/usb-to-uart-bridge-vcp-drivers?tab=downloads> Aseguresé de descargar la versión 6.7 para la instalación, y que se de la lectura del dispositivo una vez usted lo conecte al puerto USB.

Puede verificar la lectura adecuada del dispositivo verificando el administrador de dispositivos de su computador



Cree una carpeta donde vaya a almacenar todos los proyectos del curso.

Descargue el firmware del ESP32 para programación en micropython desde:

[https://micropython.org/download/ESP32\\_GENERIC/](https://micropython.org/download/ESP32_GENERIC/) descargue la ultima versión y guardela en la carpeta creada para los proyectos del curso.

Luego, vaya a VS code y abra la carpeta que creo para trabajar en el curso, cuando abra esa carpeta desde VS code, debería estar el archivo de firmware que descargo.

Ejecute el comando mostrado en la imagen para borrar el contenido del ESP32 (sustituya COM9 por su propio puerto)

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS

PS C:\Users\yomin> python -m esptool --port COM9 erase_flash
Warning: Deprecated: Command 'erase_flash' is deprecated. Use 'erase-flash' instead.
esptool v5.0.2
Connected to ESP32 on COM9:
Chip type:      ESP32-D0WDQ6 (revision v1.0)
Features:       Wi-Fi, BT, Dual Core + LP Core, 240MHz, Vref calibration in eFuse, Coding Scheme None
Crystal frequency: 40MHz
MAC:            7c:9e:bd:4b:68:00

Stub flasher running.

Flash memory erased successfully in 3.4 seconds.

Hard resetting via RTS pin...
PS C:\Users\yomin>
```

Luego haga uso de los comandos `cd`, `ls`, `D:` (SI ES NECESARIO), para dirigirse a la carpeta donde va a manejar el desarrollo del curso.

Y una vez allí corra el siguiente comando para instalar el firmware de micropython en el ESP32

```
PS E:\ITM\Docencia\Digital2> python -m esptool --chip esp32 --port COM9 --baud 460800 write_flash -z 0x1000 ESP32_GENERIC-202411
29-v1.24.1.bin
Warning: Deprecated: Command 'write_flash' is deprecated. Use 'write-flash' instead.
esptool v5.0.2
Connected to ESP32 on COM9:
Chip type:      ESP32-D0WDQ6 (revision v1.0)
Features:       Wi-Fi, BT, Dual Core + LP Core, 240MHz, Vref calibration in eFuse, Coding Scheme None
Crystal frequency: 40MHz
MAC:            7c:9e:bd:4b:68:00

Stub flasher running.
Changing baud rate to 460800...
Changed.

Configuring flash size...
Flash will be erased from 0x00001000 to 0x0019efff...
Wrote 1691664 bytes (1109554 compressed) at 0x00001000 in 28.3 seconds (477.9 kbit/s).
Hash of data verified.

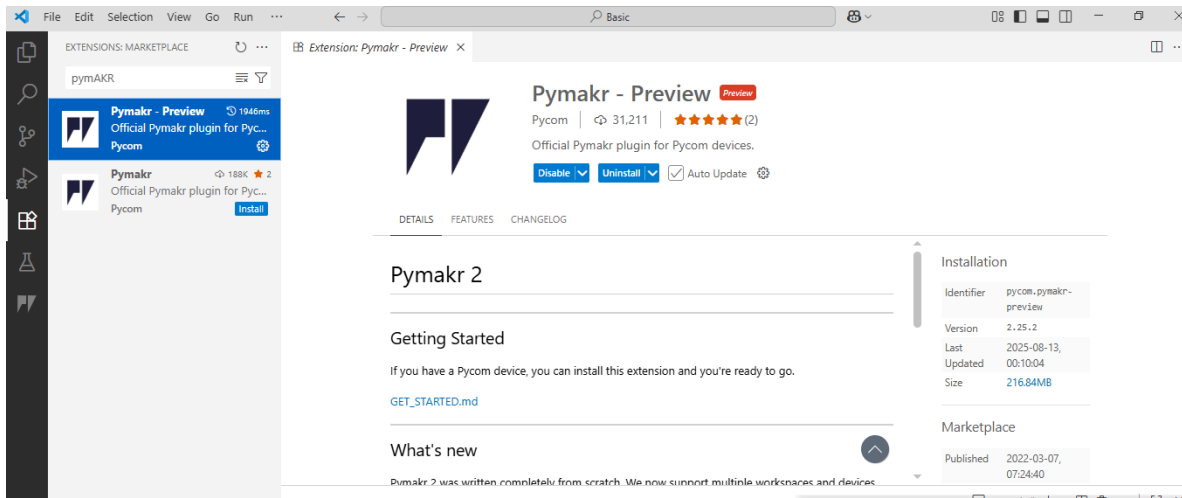
Hard resetting via RTS pin...
PS E:\ITM\Docencia\Digital2>
```

## AHORA SU ESP32 PUEDE SER PROGRAMADO EN MICROPYTHON

En VS Code vamos a instalar una extensión que nos permita programar en micropython, vaya a la sección de extensions



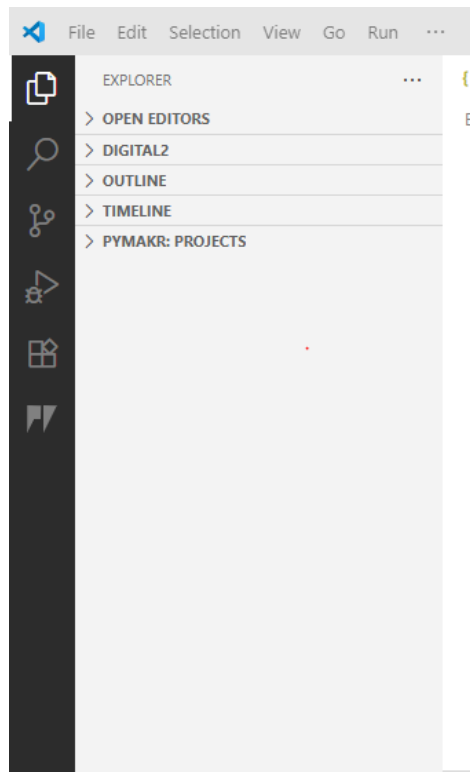
Y busque PyMakr para la instalación, Instale la preview, que cuenta con 5 estrellas



Después de realizar la Instalación, cierre visual y vuélvalo abrir para que los cambios sean añadidos.

Vaya a la carpeta donde va a manejar el curso y cree una carpeta que se llame “My-first-Project”

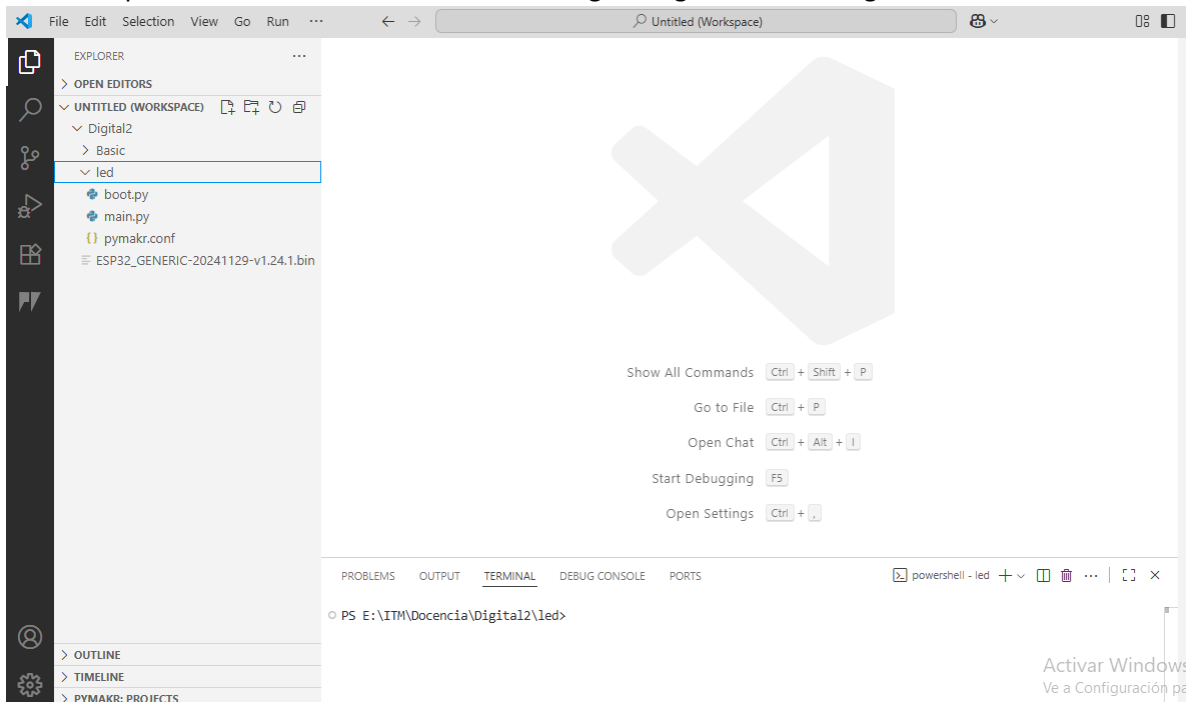
Luego abra visual studio code, la interface debe lucir así:



Despliegue pymark Project y dele en el símbolo “+”, para crear un proyecto, luego seleccione la carpeta que creo “My-first-Project”.

Dele el nombre al proyecto, indique que lo desea crear en la misma carpeta en la que esta y dígame que quiere que lo cree con Led-example.

Si llevo el proceso de forma adecuada debería llegar a algo similar a lo siguiente:



Cambie el contenido del pymkr.conf por lo siguiente :

```
{  
  "py_ignore": [  
    ".vscode",  
    ".gitignore",  
    ".git",  
    "env",  
    "venv"  
  ],  
  "name": "Basic",  
  "address": "COM9",  
  "username": "micro",  
  "password": "python",  
}
```

```
"sync_folder": "",  
"open_on_start": true  
}
```

Y el de main.py por:

```
from machine import Pin  
import time
```

# LED interno suele estar en GPIO2; si usas LED externo, cambia 2 por tu pin

```
led = Pin(2, Pin.OUT)
```

```
while True:
```

```
    led.on()    # encender
```

```
    time.sleep(0.2)
```

```
    led.off()   # apagar
```

```
    time.sleep(0.5)
```

Desde este punto, el docente los guiará