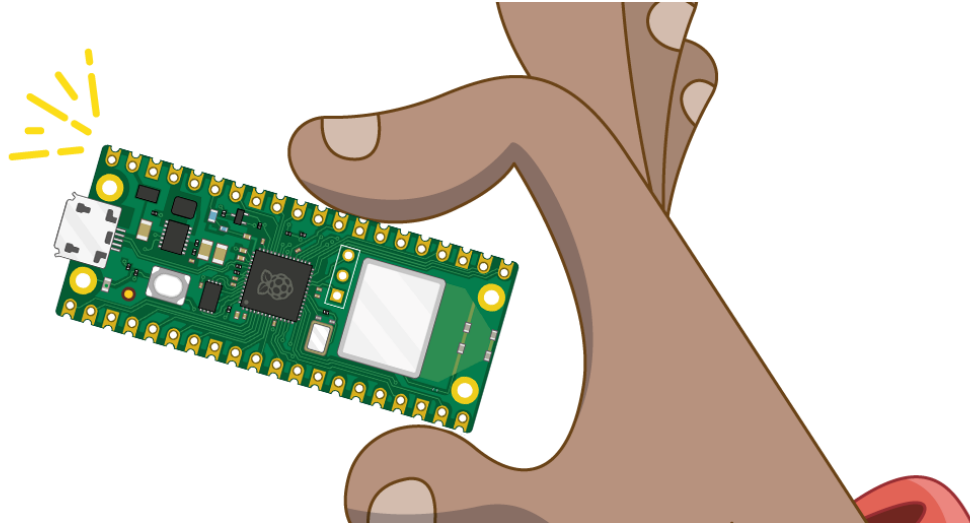


# Primeros pasos con su Raspberry Pi Pico W

Frambuesa Pi Pico   Pitón



## Contents

### ¡Guarda tu progreso!

Si desea volver a este proyecto más tarde, puede crear una cuenta de Raspberry Pi para guardar su progreso hasta el momento. En tu cuenta también verás todos los proyectos que completes.

## Iniciar sesión o registrarse

## Conecte su Raspberry Pi Pico W a una WLAN

Aquí, aprenderá a usar MicroPython para conectar su Raspberry Pi Pico W a una red de área local inalámbrica (WLAN), más comúnmente conocida como red WiFi.

```
Shell -
MicroPython v1.18-659-g7afd6452f-dirty on 2022-06-15; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
Waiting for connection...
Waiting for connection...
Waiting for connection...
Connected on 192.168.1.143
>>>
```

Las contraseñas deben mantenerse de forma segura y privada. En este paso, agregará su contraseña WiFi en su archivo de Python. Asegúrese de no compartir su archivo con nadie a quien no quiera decirle su contraseña.

Para conectarse a una red WiFi, necesitará conocer su identificador de conjunto de servicios (SSID). Este es el nombre de su red WiFi. También necesitará su contraseña WiFi. Por lo general, estos se pueden encontrar escritos en su enrutador inalámbrico, aunque debería haber cambiado la contraseña predeterminada a algo único.



En Thonny, importe los paquetes que necesitará para conectarse a su red WiFi, lea el sensor de temperatura integrado y encienda el diodo emisor de luz (LED) integrado.

```
servidor_web.py
import network
import socket
from time import sleep
from picozero import pico_temp_sensor, pico_led
import machine
```

Guarde este código ahora y elija la opción para guardar en **esta computadora**



A continuación, configure su Raspberry Pi Pico W para usar el LED integrado y, además, agregue el SSID y la contraseña de su red.

```
servidor_web.py
ssid = 'NAME OF YOUR WIFI NETWORK'
password = 'YOUR SECRET PASSWORD'
```



Ahora, comience a crear una función para conectarse a su WLAN. Debe configurar un `wlan` objeto, activar la conexión inalámbrica y proporcionarle al objeto su `ssid` y `password`.

```
servidor_web.py
def connect():
    #Connect to WLAN
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(ssid, password)
```



Si alguna vez ha conectado un dispositivo a una red WiFi, sabrá que no sucede instantáneamente. Su dispositivo enviará solicitudes a su enrutador WiFi para conectarse, y cuando el enrutador responda, realizarán lo que se llama un apretón de manos para establecer una conexión. Para hacer esto con Python, puede configurar un bucle que seguirá enviando solicitudes cada segundo hasta que se haya realizado el protocolo de enlace de conexión.

```
servidor_web.py
def connect():
    #Connect to WLAN
    wlan = network.WLAN(network.STA_IF)
```

```
wlan.active(True)
wlan.connect(ssid, password)
while wlan.isconnected() == False:
    print('Waiting for connection...')
    sleep(1)
```



Ahora imprima su configuración WLAN y pruébelo todo. Deberá llamar a su función. Mantenga todas sus llamadas a funciones en la parte inferior de su archivo, para que sean las últimas líneas de código que se ejecutan. Debido a que la conexión WiFi puede permanecer activa, incluso cuando detiene el código, puede agregar un `try/ except` que restablecerá la Raspberry Pi Pico W cuando se detenga el script.

```
servidor_web.py
def connect():
    #Connect to WLAN
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(ssid, password)
    while wlan.isconnected() == False:
        print('Waiting for connection...')
        sleep(1)
    print(wlan.ifconfig())

try:
    connect()
except KeyboardInterrupt:
    machine.reset()
```



**Prueba:** guarde y ejecute su código. Debería ver una salida en el shell que se parece a esto, aunque las direcciones IP específicas serán diferentes.

```
Waiting for connection...
Waiting for connection...
Waiting for connection...
Waiting for connection...
Waiting for connection...
('192.168.1.143', '255.255.255.0', '192.168.1.254', '192.168.1.254')
```

## El Raspberry Pi Pico W no se conecta



No necesita toda la información proporcionada por `wlan.ifconfig()`. La información clave que necesita es la dirección IP de Raspberry Pi Pico W, que es la primera información. Puede usar un **fstring** para generar la **dirección IP**. Al colocar un `f` delante de su cadena, las variables se pueden imprimir cuando están rodeadas por `{}`.

```
servidor_web.py
def connect():
    #Connect to WLAN
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(ssid, password)
    while wlan.isconnected() == False:
        print('Waiting for connection...')
        sleep(1)
    ip = wlan.ifconfig()[0]
    print(f'Connected on {ip}')
```

```
try:
    connect()
except KeyboardInterrupt:
    machine.reset()
```



Ahora puede devolver el valor de la dirección IP de su Raspberry Pi Pico W y almacenarlo cuando llame a su función.

```
servidor_web.py
def connect():
    #Connect to WLAN
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(ssid, password)
    while wlan.isconnected() == False:
        print('Waiting for connection...')
        sleep(1)
    ip = wlan.ifconfig()[0]
    print(f'Connected on {ip}')
    return ip
```

```
try:
    ip = connect()
except KeyboardInterrupt:
    machine.reset()
```

## Guarda tu proyecto



abrir un zócalo

---

¿Detectó un error? ¿Disfrutando del proyecto? ¿Alguna opinión sobre la web? ¡Háznos saber!

Enviar comentarios

Publicado por [Raspberry Pi Foundation](#) bajo una [licencia Creative Commons](#) . [Ver proyecto y licencia en GitHub](#)

[Accesibilidad](#)

[Política de cookies](#)

[Política de privacidad](#)