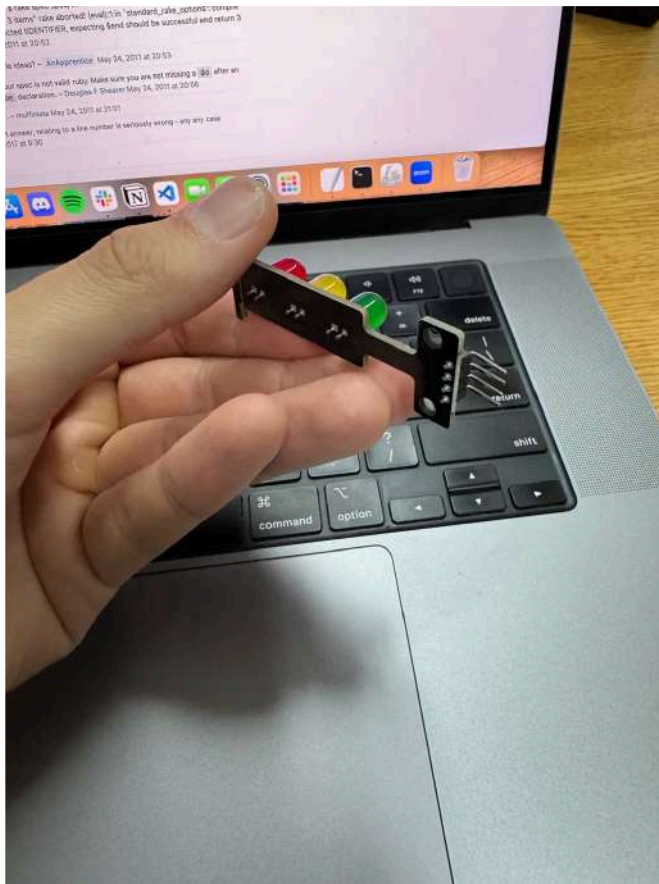


# TP3

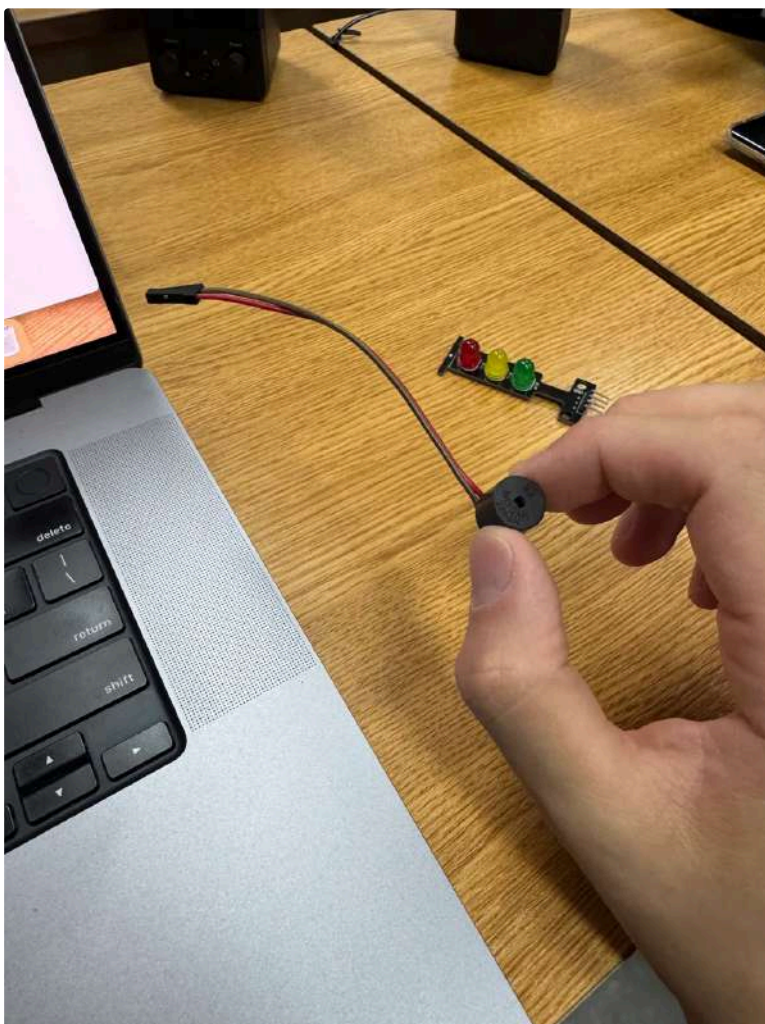
## 1. Descripción de Sensores y Actuadores:

### Actuadores:

**LEDs estilo semáforo:** Estos funcionan como indicadores visuales. Puedes encenderlos o apagarlos (o cambiar su color en secuencia) en respuesta a distintas condiciones, simulando el funcionamiento de un semáforo.



**Speaker (altavoz):** Este actuador convierte señales eléctricas en sonido. Puede emitir tonos.



### Sensores:

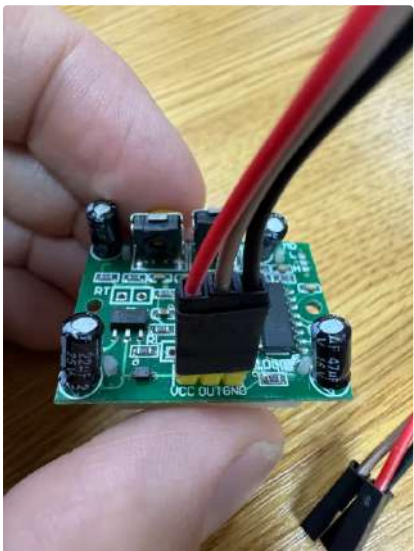
**Vcc:** Conecta el cable **rojo** al pin de **tensión** del sensor.

**GND:** Conecta el cable **negro** al **tierra** o **ground**.

**Señal:** Conecta el cable **marrón** al pin de **entrada de señal** de la Raspberry Pi.



Al realizar un debug de este pin de señal, se observará una serie de pulsos mientras el sensor detecta movimiento. Cuando deje de detectar movimiento, el pin enviará un valor de 0.



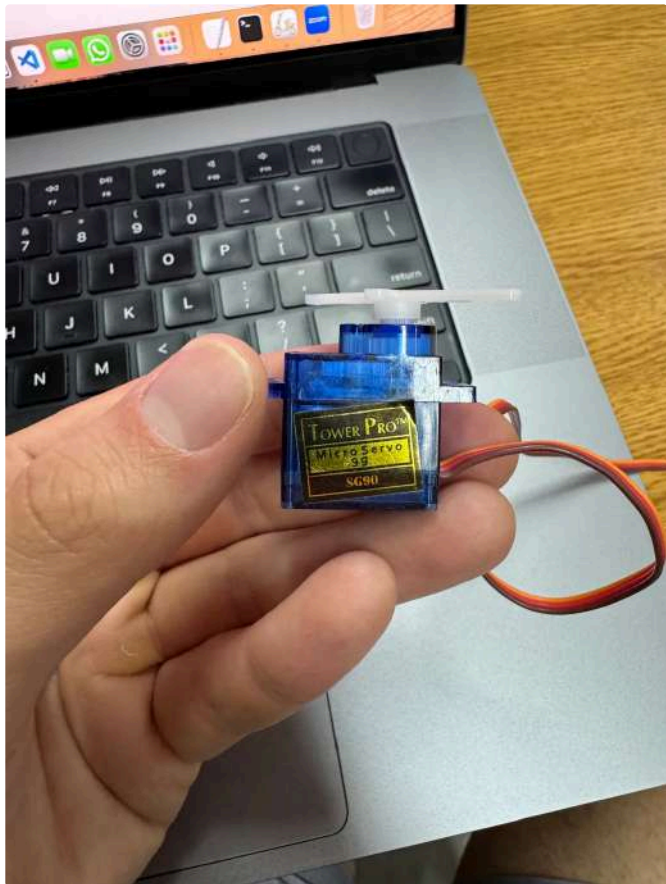
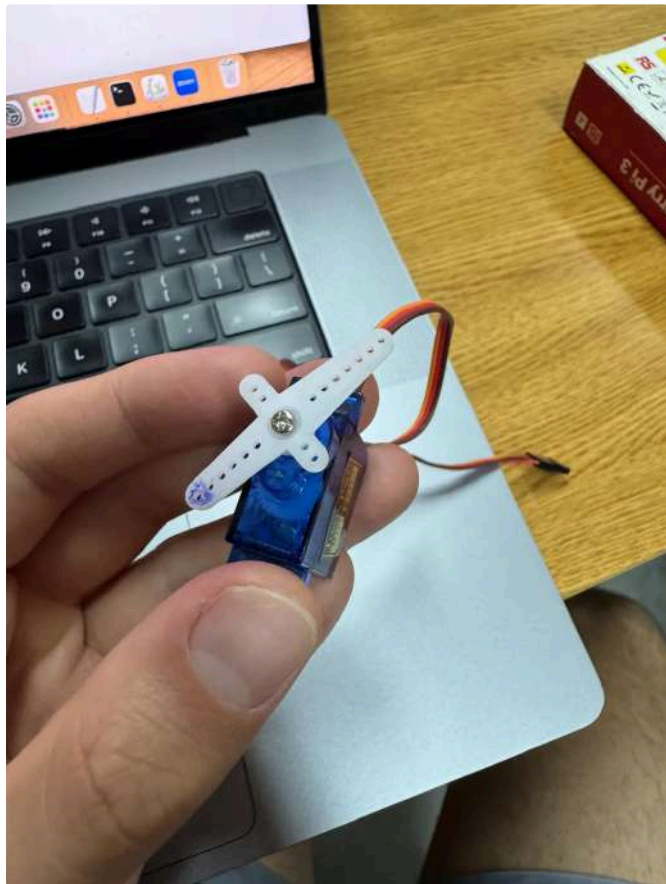
Encendimos la raspberry pi conectándola por usb a la pc para darle alimentación.



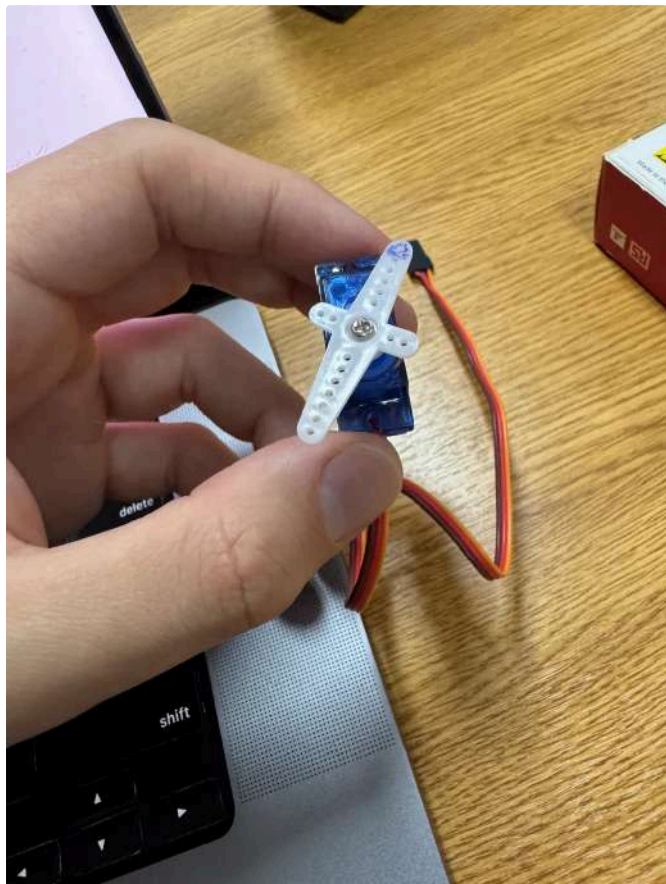




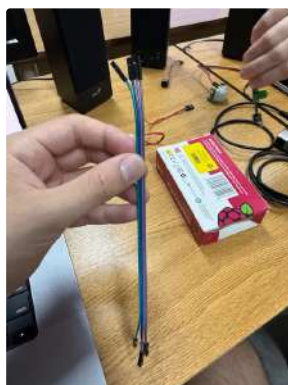
Un servo es un actuador que convierte señales en movimiento angular controlado. Se usa para girar o posicionar objetos.







Y ahora para el led



Buscar el GPIO según el modelo de Raspberry Pi que estamos usando, en este caso Raspberry Pi 2.

La IP de nuestro modelo (Raspberry Pi 2, Model B 02) es: **163.10.53.49** .

Credenciales de acceso:

Usuario: **admin**

Contraseña: **Cloud2024..**

```
→ Cloud Computing ssh admin@163.10.53.49
admin@163.10.53.49's password:
Linux raspberrypi 6.6.31+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.31-1+rpt1 (2024-05-29) aarc
```

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: Wed Oct 23 16:11:15 2024

^C

-bash-5.2\$

-bash-5.2\$ ls

-bash-5.2\$

Vamos a instalar node red

<https://nodered.org/docs/getting-started/raspberrypi>

-bash-5.2\$ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master

This script checks the version of node.js installed is 16 or greater. It will try to install node 20 if none is found. It can optionally install node 18 or 20 LTS for you.

If necessary it will then remove the old core of Node-RED, before then installing the latest version. You can also optionally specify the version required.

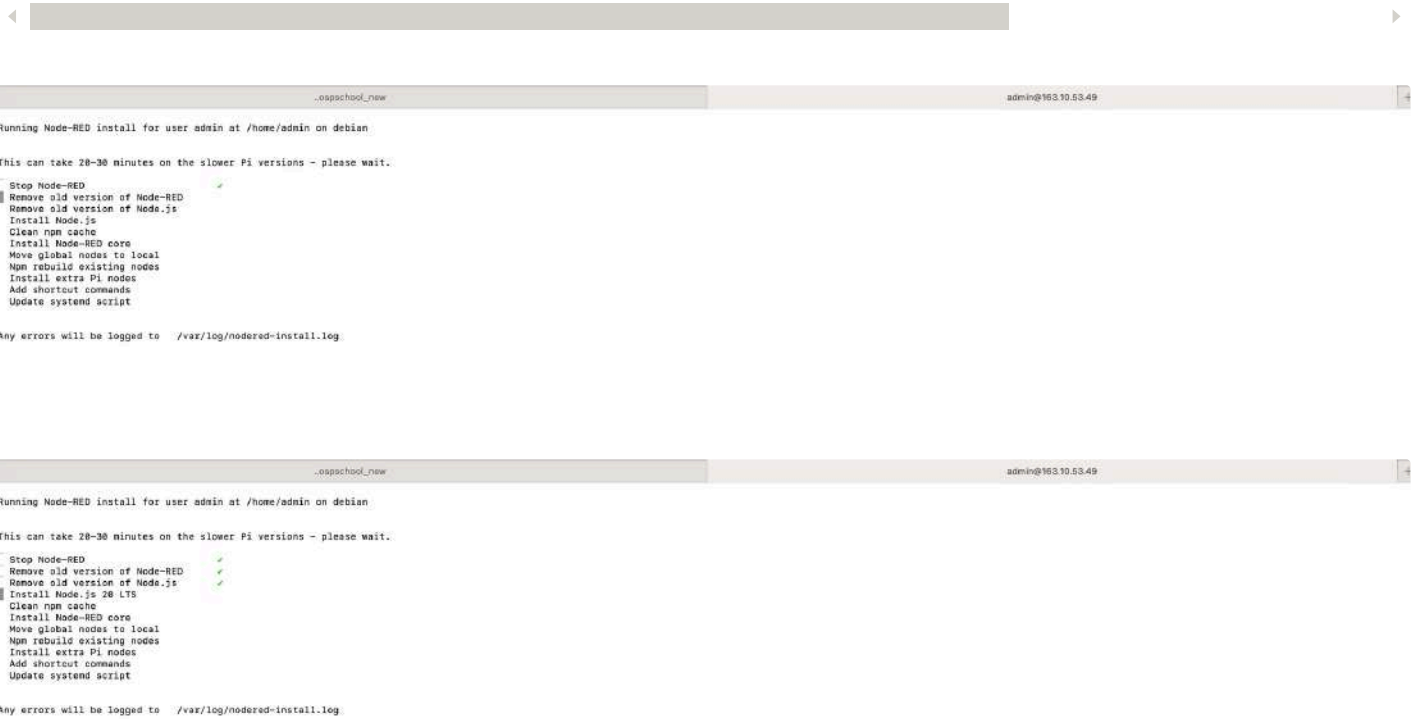
It also tries to run 'npm rebuild' to refresh any extra nodes you have installed that may have a native binary component. While this normally works ok, you need to check that it succeeds for your combination of installed nodes.

To do all this it runs commands as root - please satisfy yourself that this will not damage your Pi, or otherwise compromise your configuration. If in doubt please backup your SD card first.

See the optional parameters by re-running this command with --help

Are you really sure you want to do this ? [y/N] ? y

Would you like to install the Pi-specific nodes ? [y/N] ? y



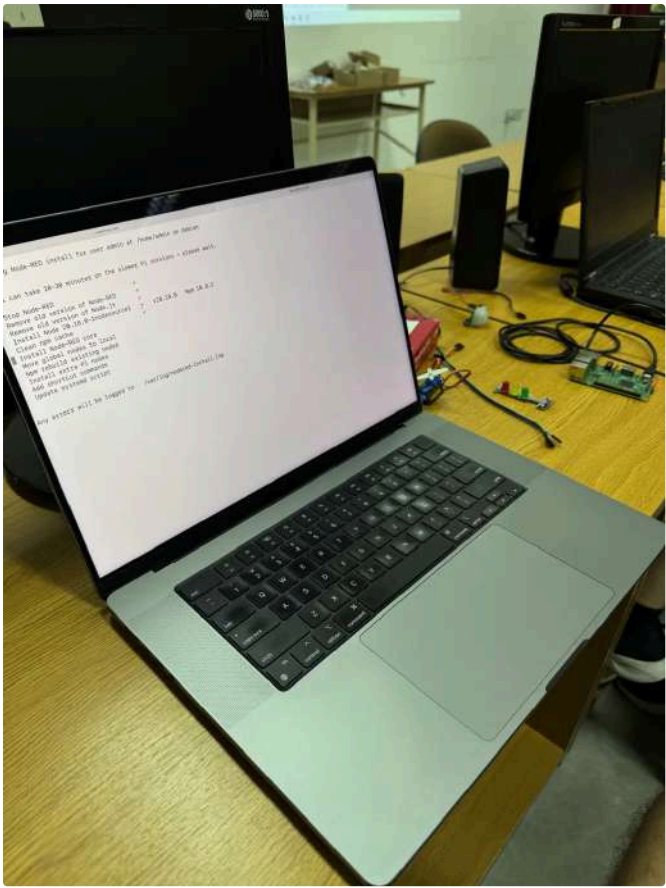
Consultar la documentación oficial de Node-RED para referencias confiables.

Nombrar cada sensor en Node-RED para facilitar su control.

Servo PWM: Usa modulación por ancho de pulso (PWM) para controlar la posición del servo.







CONFIGURACIÓN:

```
*****

Node-RED Settings File initialisation
=====
This tool will help you create a Node-RED settings file.

✓ Settings file · /home/admin/.node-red/settings.js

User Security
=====
✓ Do you want to setup user security? · No

Projects
=====
The Projects feature allows you to version control your flow using a local git repository.

✓ Do you want to enable the Projects feature? · No

Flow File settings
=====
✓ Enter a name for your flows file · flows.json
✓ Provide a passphrase to encrypt your credentials file ·

Editor settings
=====
✓ Select a theme for the editor. To use any theme other than "default", you will need to i
✓ Select the text editor component to use in the Node-RED Editor · monaco (default)

Node settings
=====
✓ Allow Function nodes to load external modules? (functionExternalModules) · Yes

Settings file written to /home/admin/.node-red/settings.js
-bash-5.2$
```

```
Settings file written to /home/admin/.node-red/settings.js
-bash-5.2$ netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 0.0.0.0:22               0.0.0.0:*               LISTEN
tcp      0      0 163.10.53.49:22         163.10.53.31:60385     ESTABLISHED
```

```
tcp      0      0 163.10.53.49:22      163.10.53.17:60277    ESTABLISHED
tcp6     0      0 :::22                 :::*                   LISTEN
-bash-5.2$
```

Hacemos esto

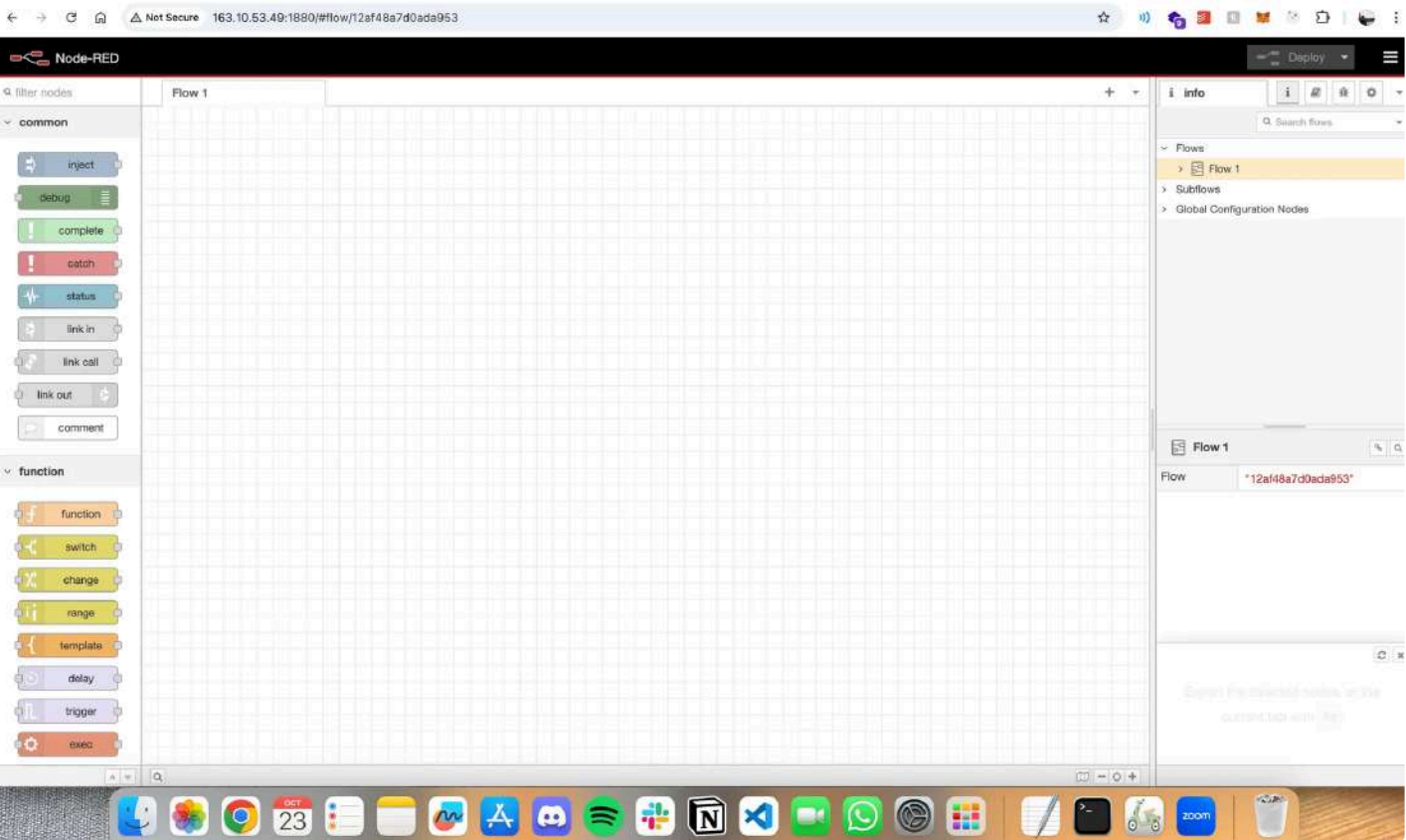
## Autostart on boot

If you want Node-RED to run when the Pi is turned on, or re-booted, you can enable the service to autostart by running the command:

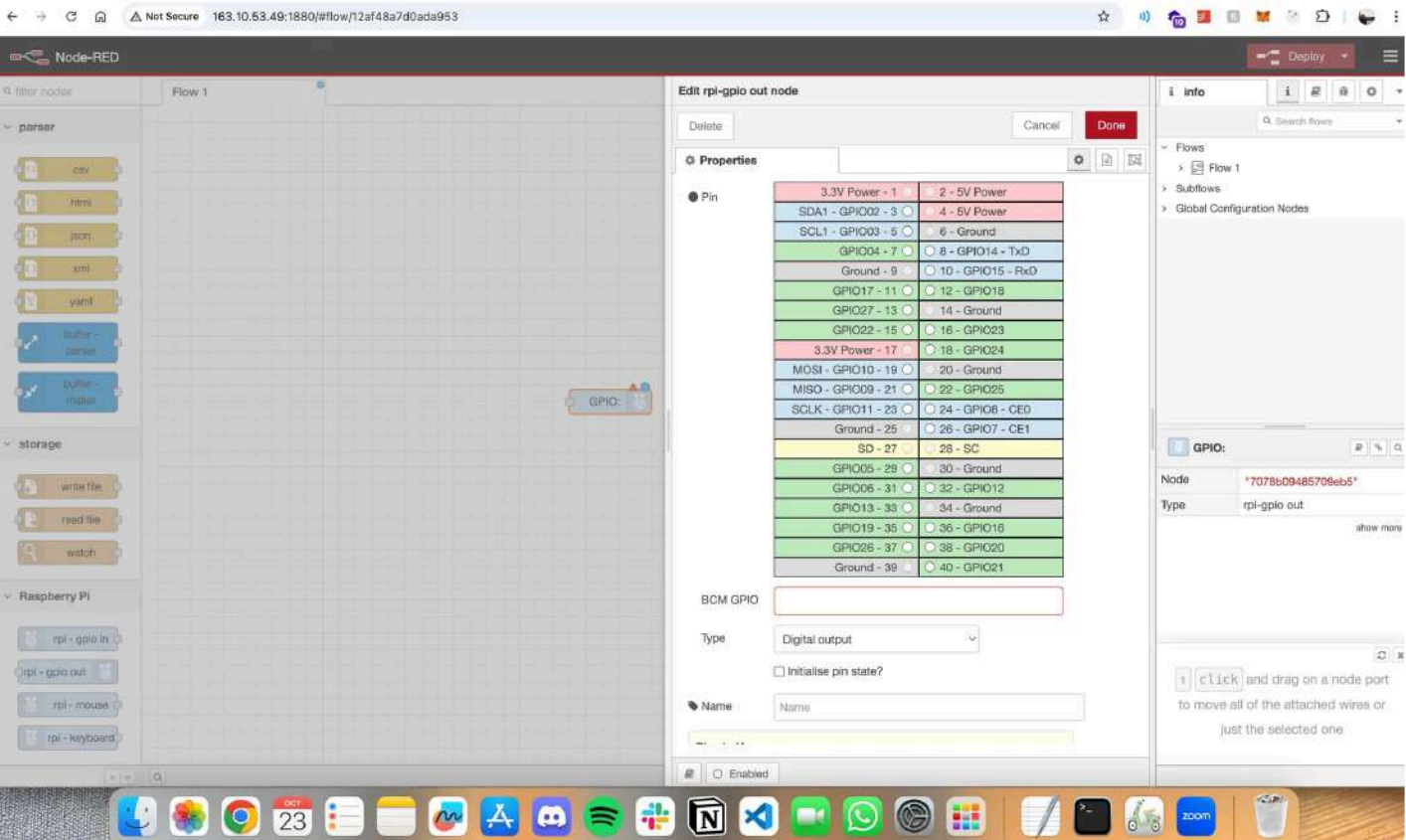
```
sudo systemctl enable nodered.service
```

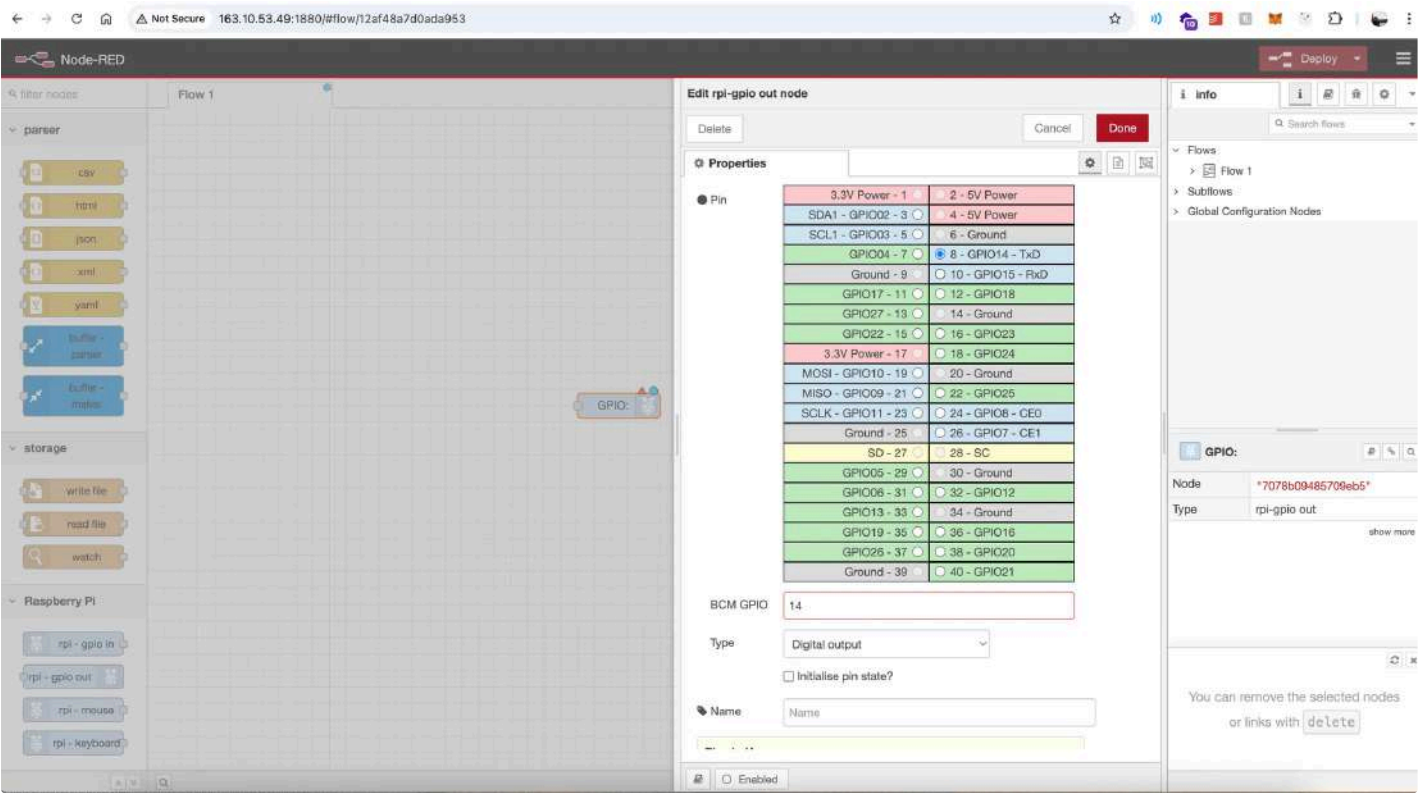
entramos ahora a: <http://163.10.53.49:1880/>

## Node-Red



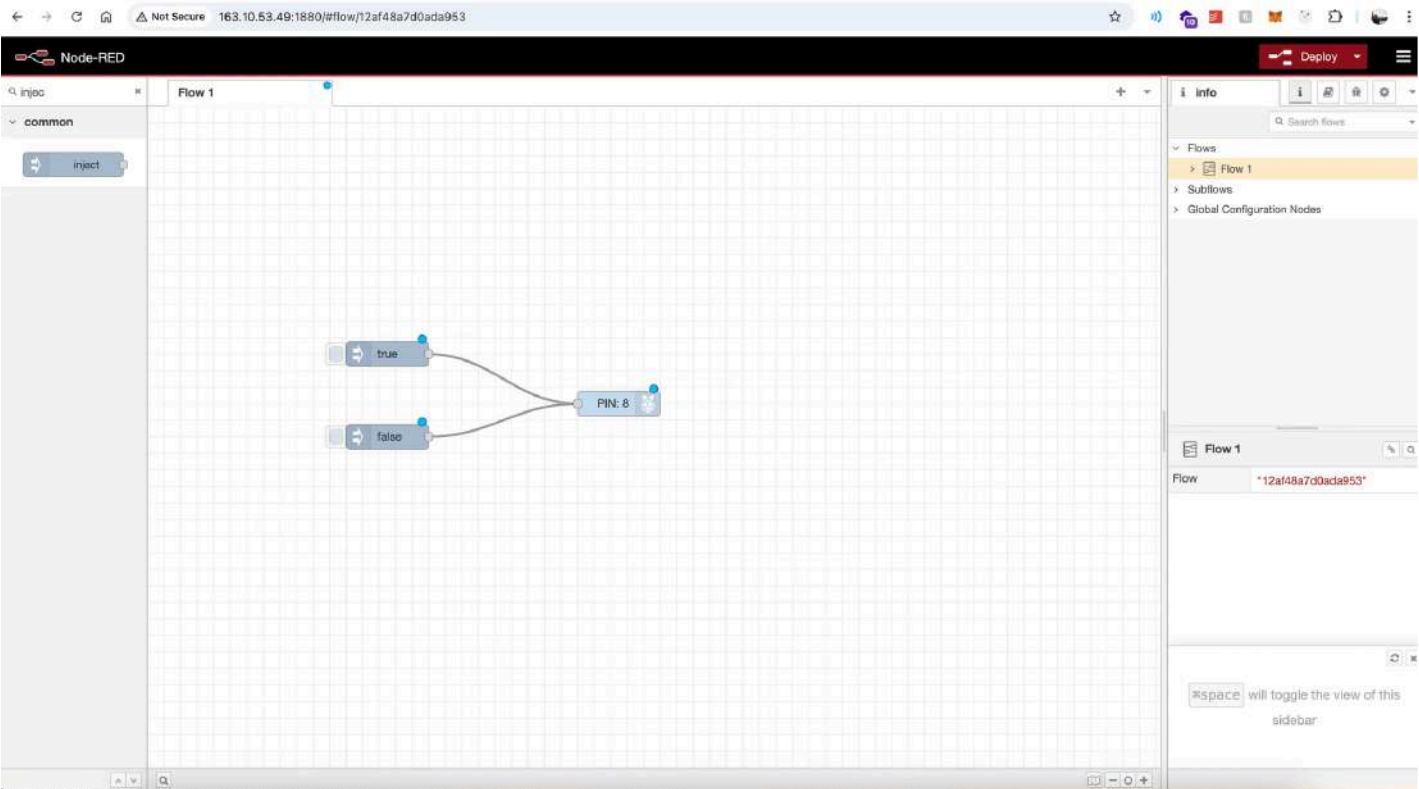
En el nodo **GPIO out** de Node-RED, no es necesario especificar el **ground**. Solo necesitamos configurar el pin de salida que queremos asignar, ya que el nodo permite mapear únicamente un pin a la vez.





Al configurar la opción Initialise pin state en el nodo GPIO, indicamos el valor con el que queremos que el pin inicie (por ejemplo, encendido o apagado).

En este caso, trabajaremos solo en modo digital. Para controlar el pin, usamos un nodo inject. Arrastramos el nodo de inject y lo conectamos al nodo GPIO, luego hacemos doble clic en **inject** para configurarlo. En esta configuración, podemos establecer el tipo de valor que queremos enviar (como un valor booleano), especificando qué queremos que transmita el nodo.



### Primeros pasos con Pines GPIO en Raspberry Pi

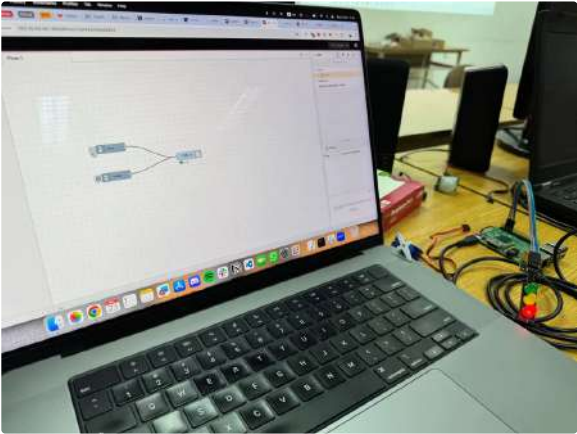
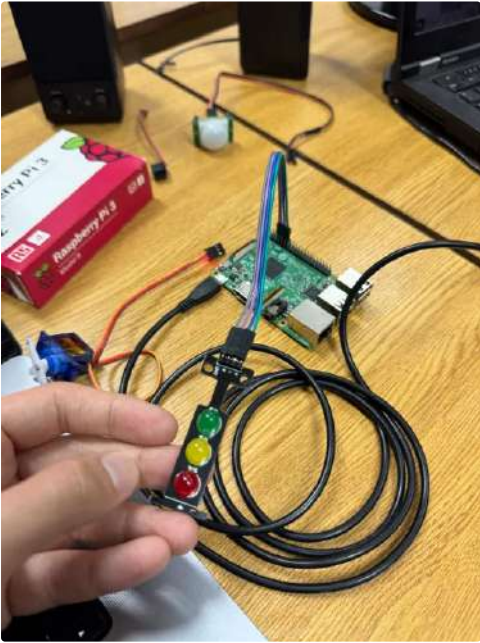
Programación en Python de los pines GPIO en Raspberry Pi.

<https://www.programoergosum.es/tutoriales/introduccion-a-pines-gpio-en...>

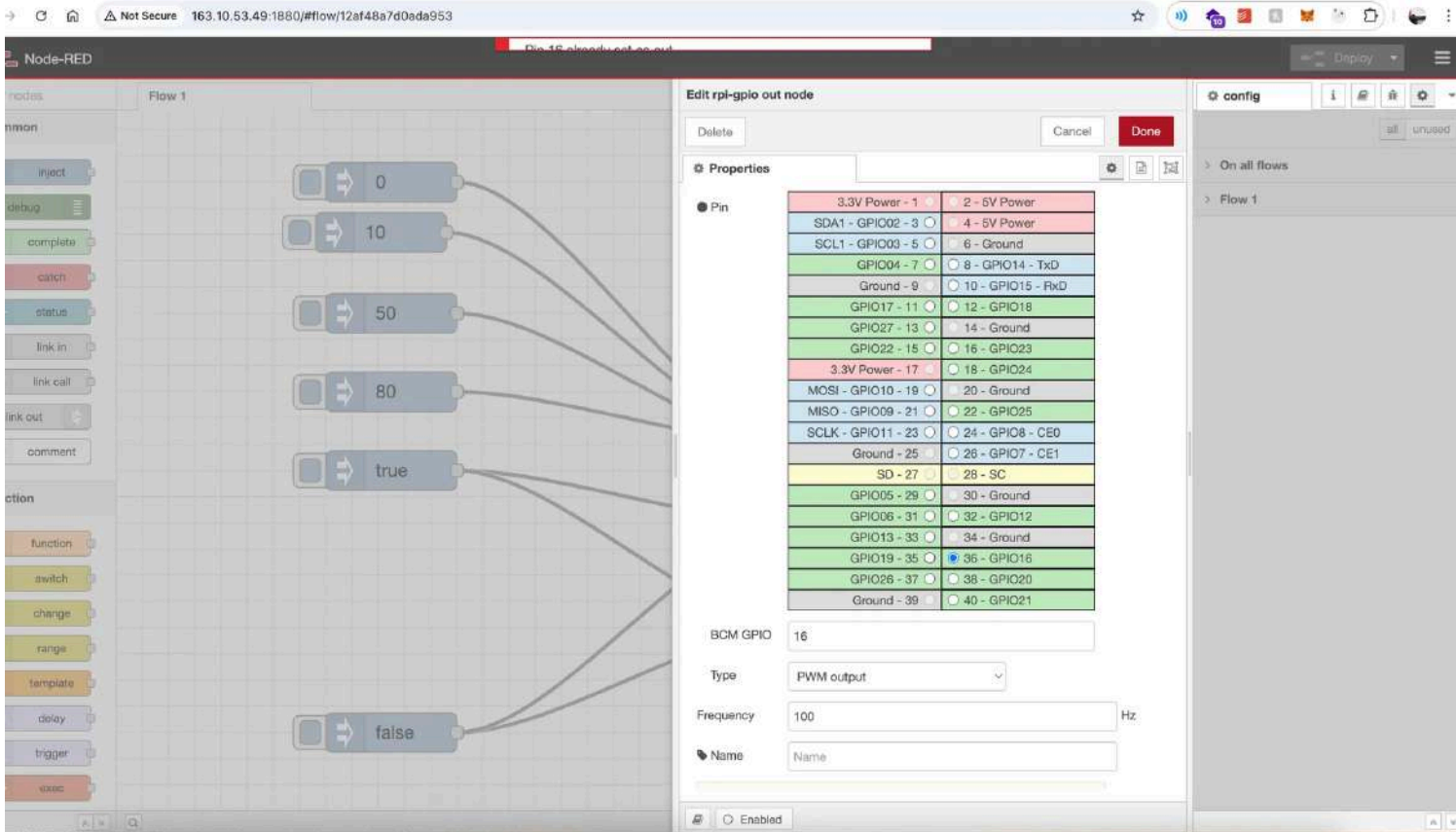
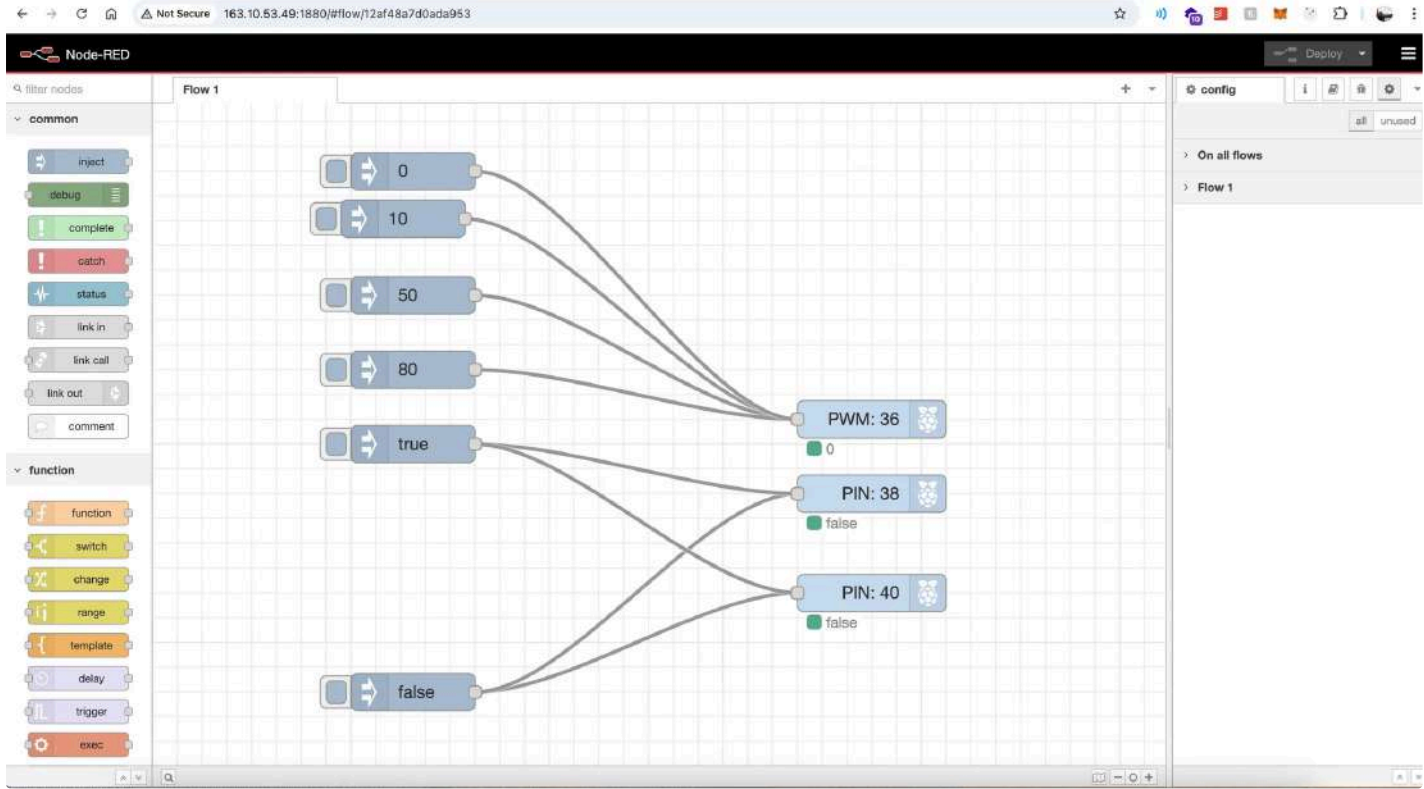
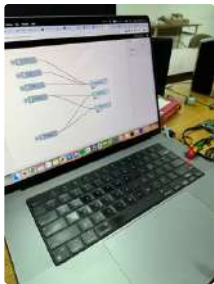
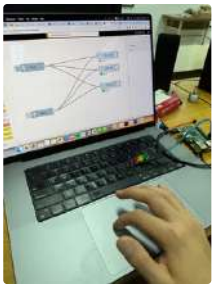
```
Ejemplo.py
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(7, GPIO.OUT)

while True:
    GPIO.output(7, True)
    time.sleep(1)
    GPIO.output(7, False)
    time.sleep(1)
```







Ejecuta el siguiente comando para apagar:

```
sudo poweroff
```

Esperar hasta que el LED verde deje de parpadear antes de desconectar el cable USB.

Para profundizar en Node-RED y Raspberry Pi, revisa los siguientes enlaces:

[Guía de inicio en Node-RED](#)

[Node-RED en Raspberry Pi](#)

[Guía para controlar un servo motor con Node-RED](#)