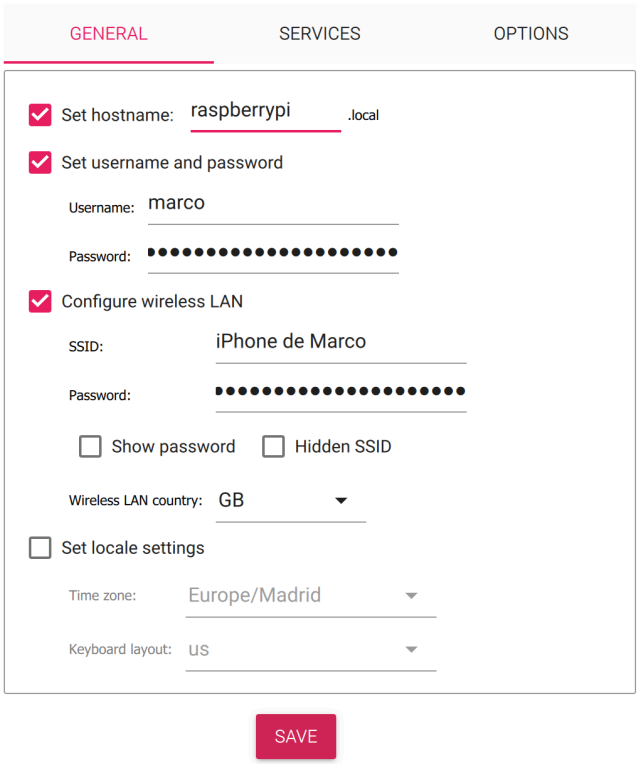
Puzzle 1

Introducción:

Para empezar con la práctica, antes tenía que descargarme el sistema operativo de mi Raspberry Pi 4B en la SSD que luego le integraría. En ella configuraría el wireless LAN poniendo el internet de mi teléfono móvil. Y buscando la IP de la Raspberry Pi podría trabajar en ella desde la terminal de mi portátil el cual estaría conectado al Wifi de mi teléfono también.

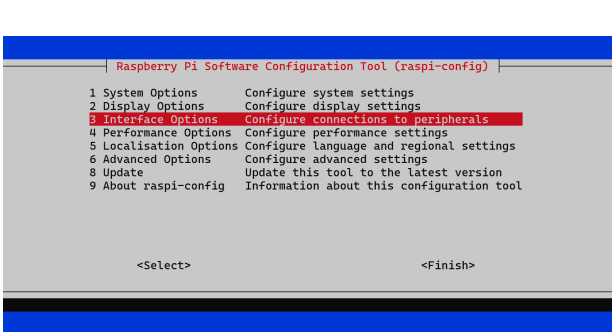


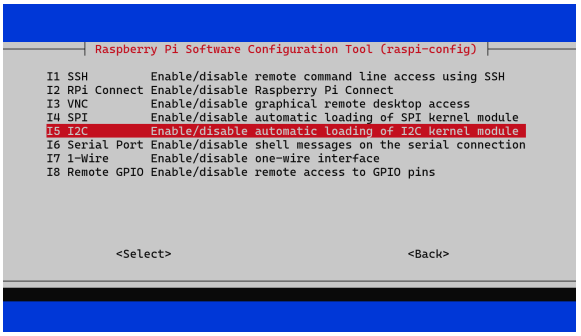
Configuración:

Una vez ya dentro de ella, tuve que habilitar la opción I2C entrando en la configuración de la Raspberry Pi con el siguiente comando:

```
(entorno) marco@raspberrypi:~/Desktop/program $ sudo raspi-config
```

Y dentro del menú entrar en las opciones de interfaz y activarlo.





Luego, en la dirección /Desktop/program creé un entorno virtual, en el cual, instalaré todas las librerías que me hagan falta para trabajar con mi RFID Itead PN532 NFC module.

```
marco@raspberrypi:~/Desktop/program $ source entorno/bin/activate
(entorno) marco@raspberrypi:~/Desktop/program $
```

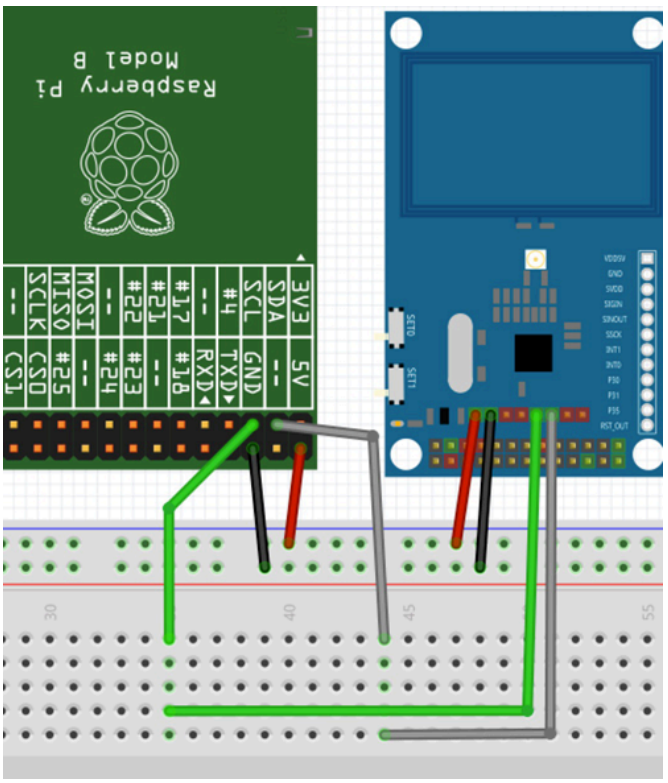
Necesitaba una librería para trabajar con las conexiones y en este caso me decanté por adafruit_pn532 (compatible también con Adafruit Blinka), ya que, la que tenía pensado desde un principio que era nfcpy no es compatible con I2C.

```
(entorno) marco@raspberrypi:~/Desktop/program $ pip install adafruit-circuitpython-pn532
```

Conexiones:

La Raspberry Pi tiene dos pines dedicados para trabajar con I2C. Estos son el GPIO 2 (SDA): Data pin y el GPIO 3 (SCL): Clock pin. Entonces tuve que conectar el PN532 correctamente:

- **VCC (cable rojo)** → 3.3V (en la imagen es conectado al de 5V pero eso es depende del módulo)
- **GND (cable negro)** → GND
- **SDA (cable gris)** → GPIO 2
- **SCL (cable verde)** → GPIO 3



Comprobé la detección de los pines del PN532, y efectivamente salió correctamente la dirección en el bus I2C 0x24.

```
marco@raspberrypi:~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- 24 -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Problemas:

Tardé mucho a la hora de empezar la práctica, por culpa de que intenté instalar primero el sistema operativo en una SSD demasiado grande y demasiado potente y después en una SSD que resultó estar corrupta. Pero como se dice normalmente, a la tercera va la vencida, y así fue.

Posteriormente, tuve problemas con la detección de la dirección y la conexión entre la Raspberry Pi y el PN532 que al final resultó ser por unos interruptores que tiene en el lateral la placa que no tenía apagados.

Programa Final:

Hice el código en un archivo sudo nano desde la terminal que lo llamé aa.py y que después ejecuté.

Aquí dejo la ejecución con la detección de una tarjeta y el código comentado.

```
(entorno) marco@raspberrypi:~/Desktop/program $ python aa.py
PN532 encontrado: chip 0x32, versión 1.6
Acerque una tarjeta NFC...
Tarjeta detectada! UID: 738F4A1C
```

```python
import board
import busio
from digitalio import DigitalInOut
from adafruit_pn532.i2c import PN532_I2C

# Configurar I2C en la Raspberry Pi
i2c = busio.I2C(board.SCL, board.SDA)

# Inicializar el módulo PN532
pn532 = PN532_I2C(i2c, debug=False)

# Obtener el firmware para comprobar comunicación
ic, ver, rev, support = pn532.firmware_version
print(f"PN532 encontrado: chip {ic:#x}, versión {ver}.{rev}")

# Configurar el PN532 para leer tarjetas NFC
pn532.SAM_configuration()

print("Acerque una tarjeta NFC...")
while True:
    uid = pn532.read_passive_target(timeout=1.0)
    if uid:
        print(f"Tarjeta detectada. UID: {uid.hex().upper()}")
        break
```