

# Laboratorio WIFI

Belky Valentina Giron Lopez  
[est.belky.giron@unimilitar.edu.co](mailto:est.belky.giron@unimilitar.edu.co)  
 Docente: José De Jesús Rúgeles

**Resumen** — En este informe se presenta el desarrollo de la práctica de laboratorio “Análisis de redes WiFi”, en la cual se exploró el funcionamiento de la comunicación inalámbrica a través de diferentes configuraciones con la Raspberry Pi Pico 2W. Durante la práctica se realizaron ejercicios como la identificación de la dirección MAC del dispositivo, el escaneo de redes disponibles, la creación de un punto de acceso inalámbrico, el cambio de canal y la medición del nivel de señal (RSSI) a distintas distancias.

**Abstract** -- This report presents the development of the "WiFi Network Analysis" lab exercise, in which the operation of wireless communication was explored through different configurations using the Raspberry Pi Pico W. During the exercise, activities included identifying the device's MAC address, scanning for available networks, creating a wireless access point, changing channels, and measuring the signal strength (RSSI) at various distances.

## I. INTRODUCCIÓN

El WiFi es una tecnología que permite la comunicación inalámbrica entre distintos dispositivos a través de ondas de radio, haciendo posible la transmisión de datos sin necesidad de cables. Funciona principalmente en las bandas de 2.4 GHz y 5 GHz; en el caso de la Raspberry Pi Pico W, se utiliza la banda de 2.4 GHz, que está dividida en 13 canales. Durante esta práctica, el objetivo fue aplicar de forma práctica los conceptos fundamentales de las redes WiFi y aprender a manejar la interfaz de red del microcontrolador Pico W. Para ello, se emplearon scripts en Thonny que permitieron realizar tareas como el escaneo de redes disponibles, la creación de un punto de acceso propio y el análisis de la intensidad de la señal recibida.

Para el desarrollo de la práctica se utilizaron los programas proporcionados en el repositorio del docente, adaptados al entorno de Thonny. El procedimiento se realizó en las siguientes fases:

1. **Obtención de la dirección MAC:** Se ejecutó el script `mac_wifi.py` para identificar la dirección única del módulo WiFi del Pico W.
2. **Escaneo de redes:** Se empleó `scanner_wifi.py` para detectar redes disponibles, registrando el RSSI y el canal.

3. **Creación de punto de acceso (AP):** Se configuró la raspberry como AP, permitiendo la conexión de un dispositivo externo y el control remoto de un LED mediante una interfaz web.
4. **Cambio de canal:** Se modificó el canal de operación del AP y se verificó mediante un escaneo.
5. **Medición del RSSI:** Se realizaron mediciones simuladas del RSSI en distintas distancias (1 m a 25 m).

## Desarrollo experimento

### Dirección MAC del dispositivo

Durante la práctica, la dirección MAC del dispositivo se mantuvo constante tras varios reinicios, confirmando que este valor está asociado al hardware del módulo WiFi.

```
<untitled> * x
1 # Digital Communications - UMMG
2 # jose.rugeles@unimilitar.edu.co
3 #
4 # Description:
5 # Minimal MicroPython script that prints the Raspberry Pi Pico W Wi-Fi
6 # MAC address (Station/STA interface) in colon-separated hexadecimal.
7 # It enables the Wi-Fi interface, reads the 6-byte MAC, formats it,
8 # prints the result, and (optionally) disables the interface.
9
10 import network
11
12 # Create and enable the Wi-Fi station interface
13 wlan = network.WLAN(network.STA_IF)
14 wlan.active(True)
15
16 # Read the 6-byte MAC address and format as colon-separated hex
17 mac_bytes = wlan.config('mac')
18 mac_str = ':'.join(f'{b:02X}' for b in mac_bytes)
19
20 print("Raspberry Pi Pico W MAC (STA):", mac_str)
21
22 # Turn Wi-Fi OFF to save power if no further Wi-Fi actions are needed
23 wlan.active(False)
```

```
Shell x
>>> %Run -c $EDITOR_CONTENT

MPY: soft reboot
Raspberry Pi Pico W MAC (STA): 28:CD:Cl:06:4C:4F
>>>
```

- La MAC del Pico (STA) identificada en el Procedimiento, ¿permanece constante entre reinicios? ¿Por qué?

Sí, la dirección MAC del Pico 2W se mantiene igual tras los reinicios porque está grabada directamente en el hardware del

módulo WiFi. Es un identificador único que no cambia, sin importar que el dispositivo se apague o reinicie.

### Escaneo de redes Wi-Fi

En el escaneo de redes, se detectaron las siguientes redes con mayor RSSI:

```
MPV: soft reboot
Scanning for Wi-Fi access points...
CH 3 | RSSI -63 dBm | BSSID DE:74:EF:5B:B0:39 | SSID: OneScreen_7280
CH 11 | RSSI -64 dBm | BSSID 28:37:37:47:86:36 | SSID: LABCOM
CH 8 | RSSI -64 dBm | BSSID C0:25:2F:44:F5:38 | SSID: <hidden>
CH 4 | RSSI -69 dBm | BSSID 90:6A:94:3E:72:48 | SSID: DAP-9D0588CPG52CA4
CH 6 | RSSI -71 dBm | BSSID 6E:29:3F:53:22:08 | SSID: iPhone de Tomas
CH 10 | RSSI -72 dBm | BSSID 10:F0:68:77:21:2F | SSID: Recover.Me-372120
CH 1 | RSSI -73 dBm | BSSID D2:14:3D:5A:44:6C | SSID: DIRECT-he-BRAVIA
CH 4 | RSSI -81 dBm | BSSID 10:F0:68:77:20:61 | SSID: UMG-PRIVR-CLL100
CH 4 | RSSI -82 dBm | BSSID 10:F0:68:77:20:60 | SSID: UMG-PUBR-CLL100
CH 6 | RSSI -92 dBm | BSSID 8E:26:35:F2:EE:42 | SSID: iPhone de Juan Sebastián
Waiting 10 seconds before next scan...

Scanning for Wi-Fi access points...
CH 3 | RSSI -58 dBm | BSSID DE:74:EF:5B:B0:39 | SSID: OneScreen_7280
CH 8 | RSSI -63 dBm | BSSID C0:25:2F:44:F5:38 | SSID: <hidden>
CH 11 | RSSI -66 dBm | BSSID 28:37:37:47:86:36 | SSID: LABCOM
CH 10 | RSSI -68 dBm | BSSID 10:F0:68:77:21:2F | SSID: Recover.Me-372120
CH 4 | RSSI -70 dBm | BSSID 90:6A:94:3E:72:48 | SSID: DAP-9D0588CPG52CA4
CH 6 | RSSI -77 dBm | BSSID 6E:29:3F:53:22:08 | SSID: iPhone de Tomas
CH 6 | RSSI -95 dBm | BSSID 8E:26:35:F2:EE:42 | SSID: iPhone de Juan Sebastián
Waiting 10 seconds before next scan...
```

Canal (CH)	RSSI (dBm)	BSSID	SSID
1	-53	D8:17:9F:18:D3:09	OneScreen_7280
8	-63	60:14:B3:C9:67:E6	iPhone de Tomas
6	-65	9E:DC:71:37:1F:8C	hidden
1	-65	D8:17:9F:1F:3B:10	OneScreen_7280
8	-70	60:14:B3:C9:67:E6	iPhone de Tomas
3	-71	6C:70:9F:3E:EE:14	LABCOM
6	-74	9E:DC:71:37:1F:8C	hidden
11	-75	C8:3A:6B:9C:53:C4	iPhone de Juan Sebastián
9	-75	C8:3A:6B:90:53:C4	iPhone de Juan Sebastián

Como se puede ver, la mejor señal la tuvo OneScreen\_7280 (-53 dBm) y la más débil fue la del iPhone de Juan Sebastián (-75 dBm).

- Compare los canales de los tres AP con mayor RSSI. ¿Hay solapamiento (1,6,11, etc en 2.4 GHz)? ¿Qué implicaciones tiene?

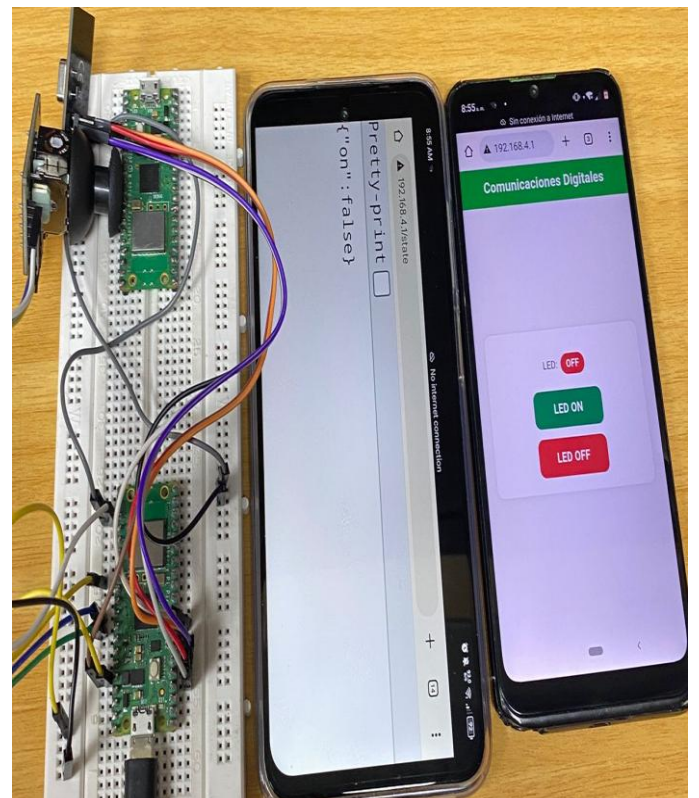
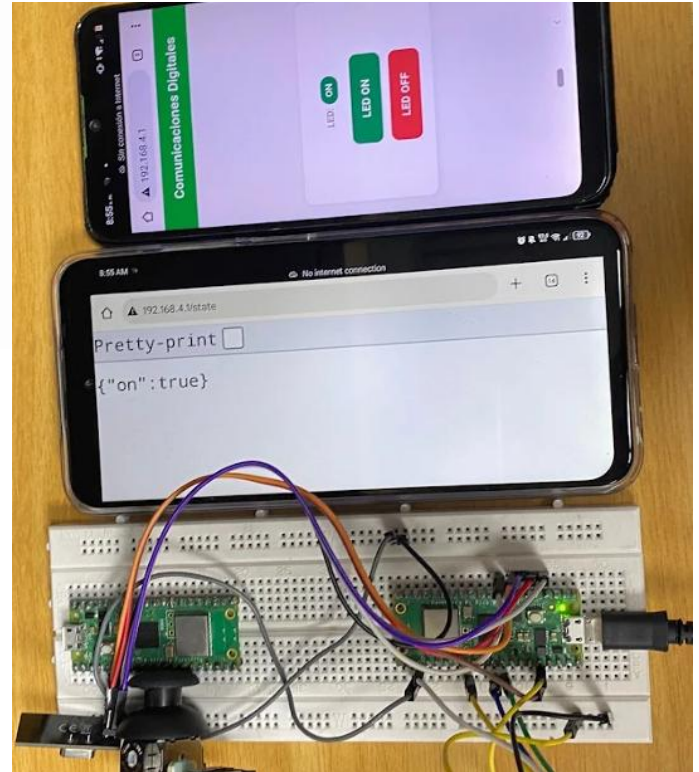
Los tres puntos de acceso con mejor señal usan los canales 1, 6 y 8. En WiFi, los canales 1, 6 y 11 son los únicos que no se solapan entre sí, pero el canal 8 sí se cruza parcialmente con el 6. Esto puede causar un poco de interferencia entre redes cercanas, lo que afecta la estabilidad y velocidad de la conexión.

- ¿El RSSI fluctúa entre iteraciones? ¿Cuáles podrían ser las causas?

Sí, el RSSI cambia entre cada medición. Esto pasa porque la

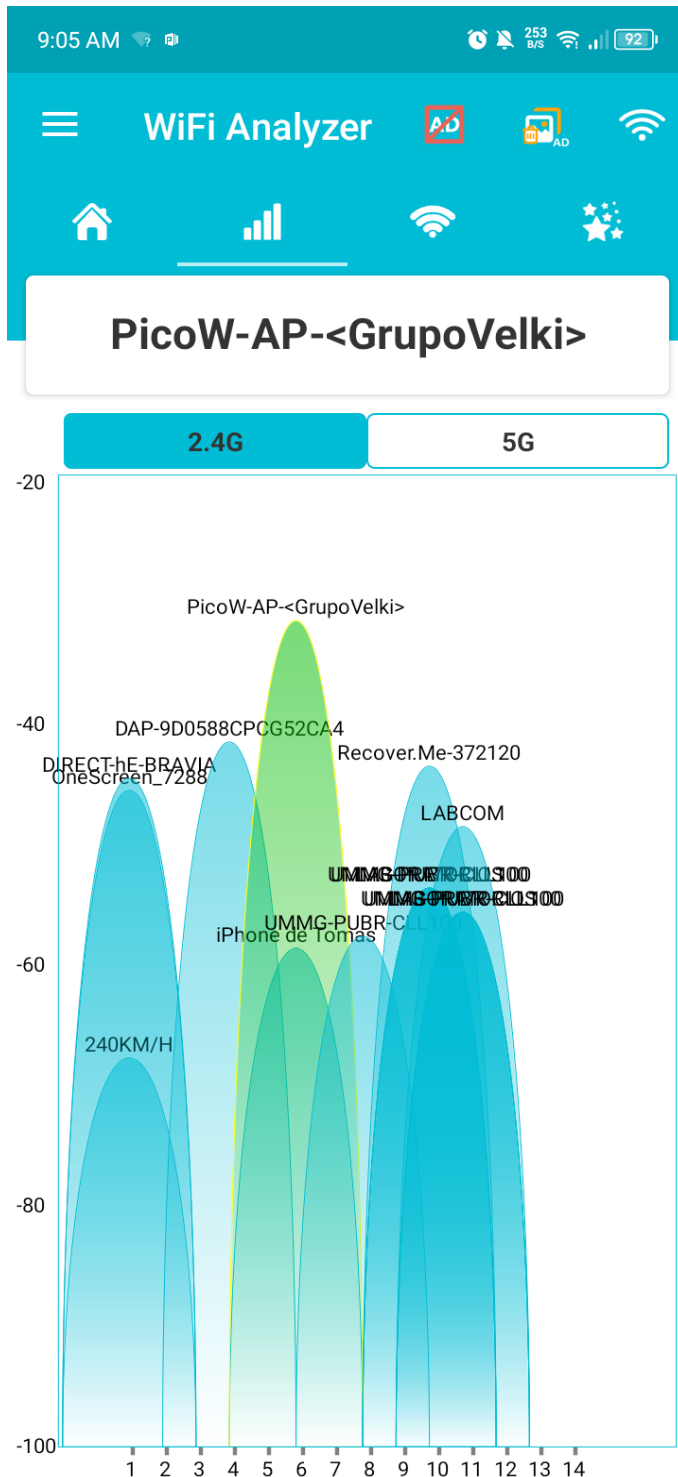
señal WiFi puede verse afectada por la distancia, los obstáculos, otras redes o aparatos electrónicos cercanos, e incluso por pequeños movimientos del dispositivo. Por eso, la intensidad de la señal no siempre se mantiene igual en cada escaneo.

### Creación de un punto de acceso inalámbrico



Al ejecutar el programa y conectarme al punto de acceso creado, todo funcionó correctamente. Desde el navegador pude acceder a la dirección, donde los botones LED ON/OFF respondieron de inmediato, encendiendo y apagando el LED del Pico 2W sin problema. Además, al ingresar a /state, el navegador mostró el JSON con el estado actual del LED, lo que confirma que la comunicación entre el microcontrolador y el dispositivo conectado fue exitosa y estable.

### Cambio de canal



```
>>> %Run -c $EDITOR_CONTENT
```

```
MPY: soft reboot
Scanning for Wi-Fi access points...
CH 6 | RSSI -52 dBm | BSSID 28:CD:C1:06:4C:4F | SSID: PicoW-AP-<GrupoVelki>
Waiting 10 seconds before next scan...
```

```
>>> %Run -c $EDITOR_CONTENT
```

```
MPY: soft reboot
Scanning for Wi-Fi access points...
CH 11 | RSSI -56 dBm | BSSID 28:CD:C1:06:4C:4F | SSID: PicoW-AP-<GrupoVelki>
Waiting 10 seconds before next scan...
```

- ¿Por que razón no hay solapamiento en los canales 1,6 y 11 en 2.4 GHz)?

En la imagen se puede ver que el punto de acceso PicoW-AP-<GrupoVelki> pasó del canal 6 al canal 11. El cambio se realizó correctamente, ya que el nombre de la red y la dirección se mantienen iguales, pero se nota una ligera diferencia en la intensidad de la señal, que pasó de -52 dBm a -56 dBm.

Esto confirma que el cambio de canal en el archivo APWifi.py funcionó bien y que la red se actualizó sin problemas. La pequeña variación en la señal es normal y puede deberse a factores como la distancia, los obstáculos o las interferencias del entorno.

Luego se detuvo la ejecución del programa, se modificó el canal en el archivo “APWifi.py” y se comprobó el cambio utilizando “scanner\_wifi.py” desde otro dispositivo.

```
>>> %Run -c $EDITOR_CONTENT
```

```
MPY: soft reboot
Scanning for Wi-Fi access points...
CH 6 | RSSI -52 dBm | BSSID 28:CD:C1:06:4C:4F | SSID: PicoW-AP-<GrupoVelki>
Waiting 10 seconds before next scan...
```

a)

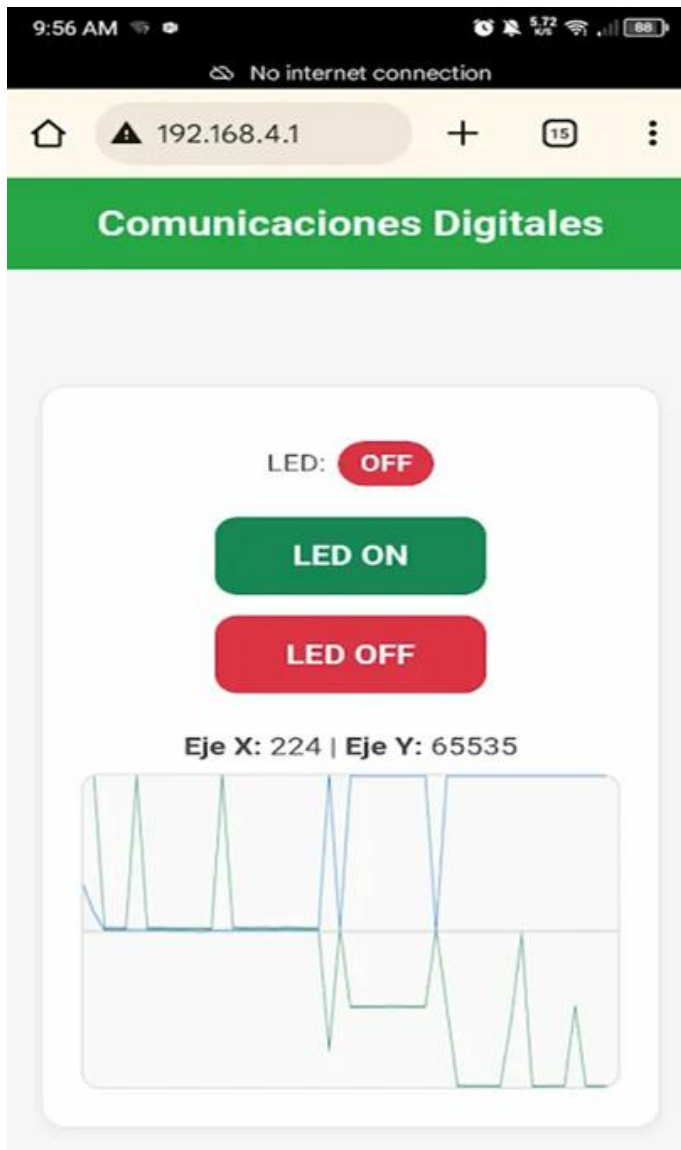
```
>>> %Run -c $EDITOR_CONTENT
```

```
MPY: soft reboot
Scanning for Wi-Fi access points...
CH 11 | RSSI -56 dBm | BSSID 28:CD:C1:06:4C:4F | SSID: PicoW-AP-<GrupoVelki>
Waiting 10 seconds before next scan...
```

No hay solapamiento entre los canales 1, 6 y 11 porque están lo suficientemente separados dentro del espectro de 2.4 GHz, de modo que sus señales no se cruzan ni interfieren entre sí. Estos tres canales se usan comúnmente porque permiten una conexión más estable y con menos interferencia, especialmente cuando hay varias redes WiFi cercanas funcionando al mismo tiempo.

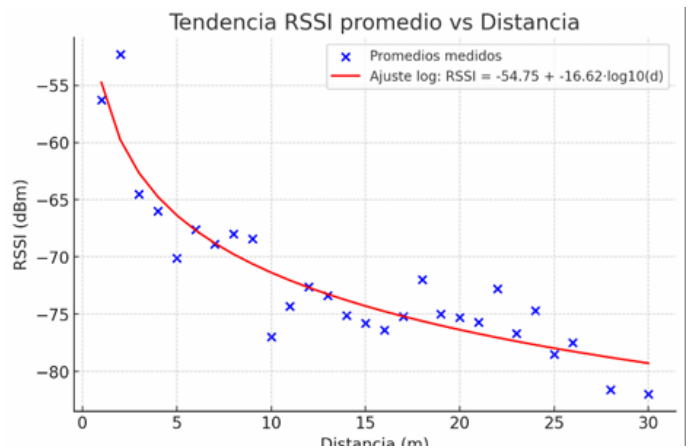
Modifique los programas APWifipico.py e index.html para que pueda leer el valor del conversor ADC 0.

Se modificó el programa “APWifipico.py” para leer los valores del conversor ADC y mostrar en tiempo real, de forma gráfica, el movimiento del joystick conectado al sistema.



Diseñe un código que le permita analizar la variación del RSSI

Se conectó la Pico 2W a un punto de acceso móvil y se midió el RSSI a distintas distancias. Los resultados mostraron una disminución logarítmica del RSSI al aumentar la distancia



La conexión se perdió a una distancia cercana a 30 metros, lo que se consideró como el límite máximo de alcance del enlace inalámbrico.

**Consulte acerca del funcionamiento de los protocolos http y UDP en WiFi. Analice las diferencias construyendo un diagrama UML y explique los procesos.**

El protocolo HTTP se usa principalmente para la comunicación entre un cliente (como un navegador web) y un servidor. Funciona mediante peticiones y respuestas, lo que permite intercambiar información de forma estructurada y confiable. Por ejemplo, cuando el cliente envía una solicitud para encender un LED, el servidor responde confirmando la acción. Aunque es más seguro y estable, puede ser un poco más lento debido al proceso de verificación y respuesta.

El protocolo UDP es mucho más rápido, ya que envía los datos directamente sin verificar si el receptor los recibió correctamente. Esto lo hace ideal para aplicaciones que requieren transmisiones continuas como audio, video o sensores en tiempo real, donde la velocidad es más importante que la confiabilidad absoluta.

El protocolo HTTP es el que se usa cuando un navegador o aplicación se comunica con un servidor, por ejemplo, al abrir una página web o enviar información. Funciona como una conversación entre dos partes: el cliente hace una solicitud (pide algo) y el servidor responde con lo que se le pidió. Este protocolo es seguro y confiable, ya que verifica que la información llegue completa antes de mostrar una respuesta.

## REFERENCIAS

[1] Universidad Militar Nueva Granada. (2025). Guía de laboratorio: Comunicación PC con Raspberry Pi Pico 2W y pantalla OLED SSD1306. Facultad de Ingeniería, Curso de Computadores II.

**Link repositorio GitHub:**

<https://github.com/belkyvalentina11/Comunicaci-n-digital-.git>