

Apéndices

A. Guía de explicación para la validación instrumental

Flujograma utilizado en GNU Radio, algoritmo utilizado en Google Colab y archivos obtenidos.

INTRODUCCIÓN

En este apartado se describe el desarrollo de la validación instrumental en condiciones controladas de laboratorio, con el propósito de estudiar la cadena completa de transmisión y recepción basada en SDR y sentar las bases para la caracterización del entorno rural de interés. Para ello, se configuró un generador de señales RF y un analizador de espectros trabajando sobre el rango de TVWS, seleccionando tres frecuencias de prueba que pertenecieran al intervalo de estudio. La señal generada se midió directamente con el analizador de espectros, obteniendo para cada frecuencia un archivo CSV con la traza espectral, frecuencia en Hz y magnitud en dBm, que posteriormente se documentó y almacenó en repositorio de GitHub junto con las evidencias gráficas.

En paralelo, la misma señal se conectó a la entrada del USRP-2920 configurado como receptor SDR, implementando en GNU Radio un flujograma que replica el análisis espectral: adquisición I/Q, agrupación en vectores, cálculo de la FFT, conversión a potencia y escala en dB, visualización en tiempo real y almacenamiento en archivos de texto. La parametrización de bloques como la tasa de muestreo en 12.5 MS/s, la longitud de FFT de 1024 puntos y el uso de ventanas adecuadas garantizó una resolución en frecuencia comparable a la del analizador de espectros. Finalmente, se desarrolló un script en Python que lee y procesa los archivos CSV de ambos sistemas, calcula la correlación entre las trazas, y aplica un algoritmo de detección de ocupación espectral mediante

un umbral adaptativo, permitiendo cuantificar el grado de concordancia entre ambas mediciones y evaluar la capacidad del SDR para caracterizar el espectro en el rango TVWS.

DESARROLLO

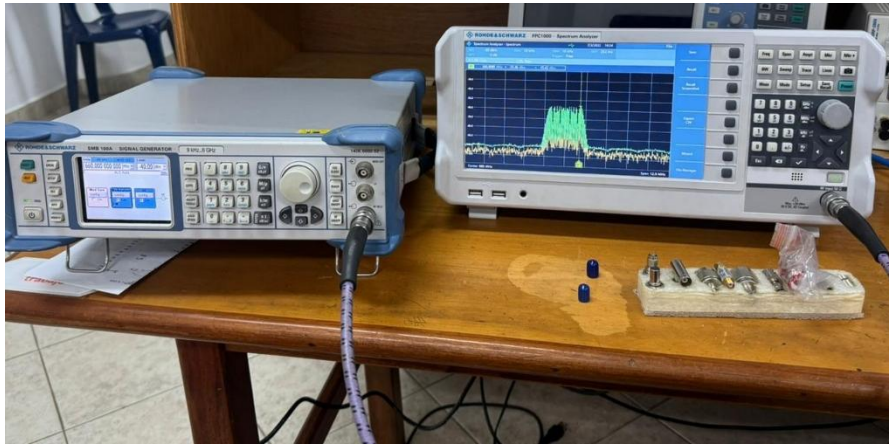
En el laboratorio de comunicaciones se configuró un generador de señales RF para emitir una portadora de prueba a distintas frecuencias dentro del espectro TVWS. En concreto, se trabajó con tres frecuencias de interés: 500 MHz, 590 MHz y 660 MHz, que fueron escogidas aleatoriamente, pertenecientes al rango de estudio.

La señal generada se midió de forma directa con un analizador de espectros, configurado con los mismos parámetros de referencia como frecuencia central, ancho de banda y nivel de referencia que se utilizarían posteriormente en la cadena SDR. El analizador permitió registrar la traza espectral y exportarla en un archivo CSV con dos columnas principales: frecuencia (Hz) y magnitud (dBm). Como se observa en la Figura 1.

Figura 1. Generador de señales y analizador de espectro.



a) Generador de señales con 660MHz.



b) Generador de señales y analizador de espectro.



c) Generador de señales y analizador de espectro.

Para cada frecuencia se obtuvo un archivo CSV con los datos mencionados, la cantidad de muestras es grande, por lo que se mostrará nada más una foto del archivo, para verlo completo se puede visitar el repositorio de Github del proyecto: (Rodríguez y Díaz, 2025)

Figura 2. Archivos CSV para diferentes frecuencias.

| | | | | | | | | | | | | | | | |
|--------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Frequency: 500000000.000000 | | | | | | | | | | | | | | | |
| Sample Rate: 12500000.000000 | | | | | | | | | | | | | | | |
| Vector Length: 1024 | | | | | | | | | | | | | | | |
| Notes: | | | | | | | | | | | | | | | |
| Run time: 30.000585 seconds | | | | | | | | | | | | | | | |
| Timestamp: 09/09/2025 19:04:02 | | | | | | | | | | | | | | | |
| -102.08 | -100.2 | -98.92 | -98.86 | -98.75 | -98.72 | -99.73 | -101.2 | -102.14 | -102.34 | -102.04 | -101.56 | -101.19 | -100.82 | -100.12 | -99.34 |
| Run time: 60.001713 seconds | | | | | | | | | | | | | | | |
| Timestamp: 09/09/2025 19:04:32 | | | | | | | | | | | | | | | |
| -121.51 | -128.59 | -121.02 | -123.31 | -117.76 | -112.17 | -112.92 | -119.52 | -121.92 | -119.49 | -116.16 | -115.3 | -119.9 | -117.6 | -113.92 | -113.58 |
| Run time: 90.001765 seconds | | | | | | | | | | | | | | | |
| Timestamp: 09/09/2025 19:05:02 | | | | | | | | | | | | | | | |
| -116.4 | -120.01 | -123.06 | -117.09 | -119.45 | -132.75 | -119.35 | -115.23 | -112.05 | -112.89 | -127.07 | -117.76 | -114.12 | -116.57 | -119.99 | -113.02 |

a) Primer archivo CSV para la frecuencia de 500MHz.

| | | | | | | | | | | | | | | | |
|--------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Frequency: 590000000.000000 | | | | | | | | | | | | | | | |
| Sample Rate: 12500000.000000 | | | | | | | | | | | | | | | |
| Vector Length: 1024 | | | | | | | | | | | | | | | |
| Notes: | | | | | | | | | | | | | | | |
| Frequency: 590000000.000000 | | | | | | | | | | | | | | | |
| Sample Rate: 12500000.000000 | | | | | | | | | | | | | | | |
| Vector Length: 1024 | | | | | | | | | | | | | | | |
| Notes: | | | | | | | | | | | | | | | |
| Run time: 30.000191 seconds | | | | | | | | | | | | | | | |
| Timestamp: 09/09/2025 19:24:00 | | | | | | | | | | | | | | | |
| -112.06 | -108.99 | -111.77 | -116.57 | -117.03 | -117.26 | -127.34 | -116.99 | -113.69 | -115.51 | -118.91 | -123.58 | -123 | -116.24 | -114.15 | -111.87 |
| Run time: 60.000714 seconds | | | | | | | | | | | | | | | |
| Timestamp: 09/09/2025 19:24:30 | | | | | | | | | | | | | | | |
| -116.82 | -121.92 | -132.73 | -120.87 | -117.88 | -117.21 | -126.16 | -113.46 | -111.38 | -115.49 | -116.76 | -113.37 | -112.04 | -113.17 | -119.15 | -124.7 |
| Run time: 90.001520 seconds | | | | | | | | | | | | | | | |
| Timestamp: 09/09/2025 19:25:00 | | | | | | | | | | | | | | | |
| -115.92 | -116.26 | -118.86 | -120.68 | -120.43 | -116.24 | -121.36 | -115.13 | -115.3 | -119.29 | -112.34 | -109.66 | -113.31 | -122.24 | -112.77 | -117.89 |

b) Primer archivo CSV para la frecuencia de 590MHz.

| | | | | | | | | | | | | | | | |
|--------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Frequency: 660000000.000000 | | | | | | | | | | | | | | | |
| Sample Rate: 12500000.000000 | | | | | | | | | | | | | | | |
| Vector Length: 1024 | | | | | | | | | | | | | | | |
| Notes: | | | | | | | | | | | | | | | |
| Frequency: 660000000.000000 | | | | | | | | | | | | | | | |
| Sample Rate: 12500000.000000 | | | | | | | | | | | | | | | |
| Vector Length: 1024 | | | | | | | | | | | | | | | |
| Notes: | | | | | | | | | | | | | | | |
| Run time: 30.000156 seconds | | | | | | | | | | | | | | | |
| Timestamp: 09/09/2025 18:52:22 | | | | | | | | | | | | | | | |
| -123.39 | -118.12 | -113.92 | -109.53 | -109.32 | -111.95 | -115.2 | -114.46 | -110.65 | -107.49 | -106.88 | -109.83 | -112.4 | -113.44 | -117.48 | -112.4 |
| Run time: 60.000301 seconds | | | | | | | | | | | | | | | |
| Timestamp: 09/09/2025 18:52:52 | | | | | | | | | | | | | | | |
| -114.54 | -112.43 | -115.17 | -120.6 | -114.94 | -119.79 | -120.64 | -116.88 | -113.48 | -110.16 | -112.01 | -115.47 | -113.58 | -114.13 | -117.45 | -118.31 |
| Run time: 90.000436 seconds | | | | | | | | | | | | | | | |
| Timestamp: 09/09/2025 18:53:22 | | | | | | | | | | | | | | | |
| -116.31 | -117.01 | -117.57 | -114.18 | -113.19 | -112.8 | -111.94 | -111.54 | -112.75 | -114.19 | -115.36 | -123.18 | -115.17 | -110.49 | -111.03 | -115.11 |

c) Primer archivo CSV para la frecuencia de 660MHz.

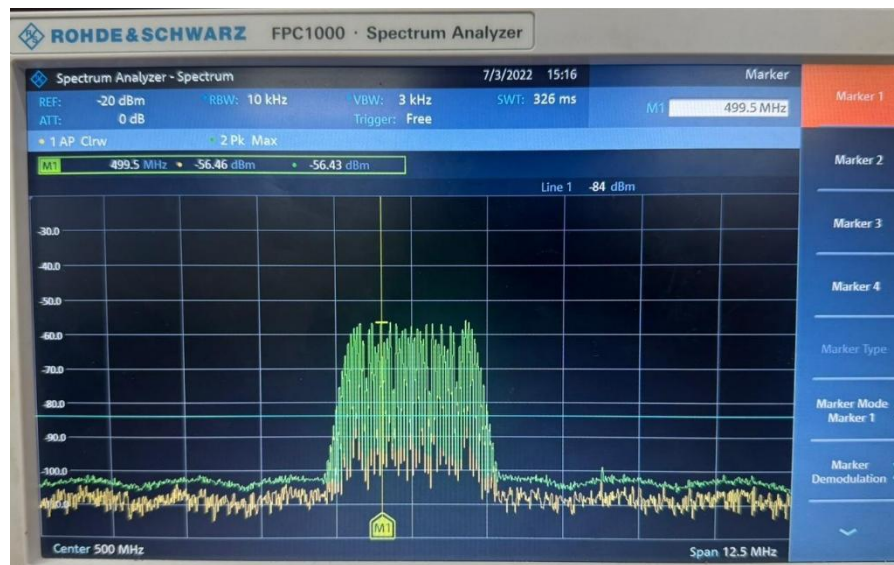


Figura 2. Analizador de espectro con 500MHz.

La misma salida del generador de señales se conectó a la entrada del USRP-2920, que se configuró como receptor SDR. La señal fue procesada mediante un diagrama de bloques en GNU Radio, en el que se encadenó la adquisición, el cálculo de la FFT y un vector to text file para almacenar los datos en un CSV. El flujograma empleado se puede entender mejor a través del siguiente diagrama de flujo que se muestra en la Figura 4.

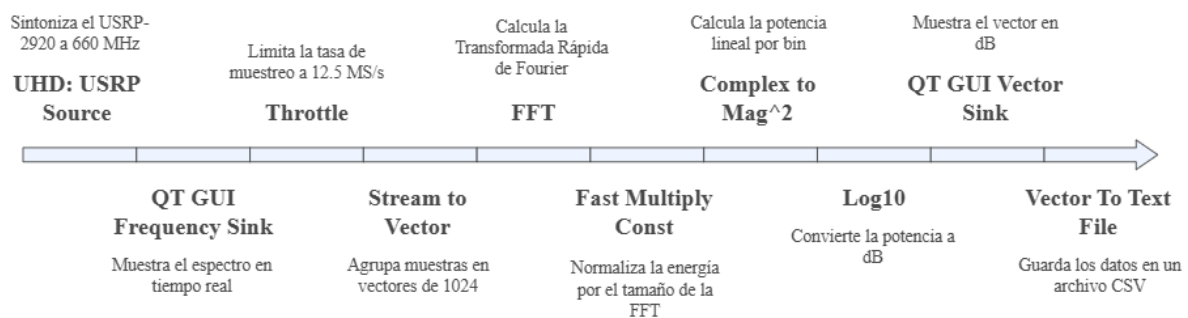


Figura 4. Diagrama de flujo empleado en GNU Radio.

The screenshot shows a GNU Radio flow graph for spectrum analysis. The flow starts with a 'UHD: USRP Source' block, which is connected to a 'QT GUI Frequency Sink' and a 'Throttle' block. The signal then goes through an 'FFT' block, a 'Fast Multiply Const' block, and a 'Complex to Mag^2' block. Finally, it branches into a 'Log10' block and a 'Vector To Text File' block. A 'QT GUI Vector Sink' block is also present. The flow graph is connected to a 'command' terminal.

Options:
 Title: Not titled yet
 Output Language: Python
 Generate Options: QT GUI

Variable:
 ID: samp_rate
 Value: 12.5M

Variable:
 ID: vec_len
 Value: 1.024k

QT GUI Range:
 ID: freq
 Default Value: 660M
 Start: 470M
 Stop: 710M
 Steps: 10M

QT GUI Range:
 ID: amp
 Default Value: 1
 Start: -11
 Stop: 11
 Steps: 500m

UHD: USRP Source:
 Sync: PC Clock
 Samp rate (Sps): 12.5M
 Ch0: Center Freq (Hz): 660M
 Ch0: AGC: Default
 Ch0: Gain Value: 0
 Ch0: Antenna: RX2

QT GUI Frequency Sink:
 FFT Size: 1024
 Center Frequency (Hz): 660M
 Bandwidth (Hz): 12.5M

Throttle:
 Sample Rate: 12.5M
 Limits: None

Stream to Vector:

FFT:
 FFT Size: 1.024k
 Forward/Reverse: Forward
 Window: window.blackmanharris
 Shifts: Yes
 Num. Threads: 1

Fast Multiply Const:
 Constant: 976.562u
 Vector Length: 1.024k

Complex to Mag^2:
 Vector Length: 1.024k

Log10:
 m: 10
 k: 0
 Vector Length: 1.024k

Vector To Text File:
 Files: ...adsipruebaHackRF.csv
 Vector Size: 1.024k
 Frequency: 660M
 Sample Rate: 12.5M
 Notes:
 Update Rate (sec): 30
 Output Precision: 2
 Append To File: Yes
 Include Runtime: Yes

QT GUI Vector Sink:
 Vector Size: 1.024k
 X-Axis Start Value: 0
 X-Axis Step Value: 1
 Ref Level: 0

A continuación, se explicará cada bloque utilizado, en la línea superior tenemos Options, Variables y QT GUI Range. Define el nombre del flujo, el tipo de GUI (QT GUI), esta opción sirve para la visualización, y el lenguaje de programación de salida, Python. Todo esto es, básicamente, la configuración del proyecto.

Variable `samp_rate`: almacena la tasa de muestreo del sistema la cual se fijó en 12.5 MS/s. Además, de acuerdo con el datasheet del USRP2920 (National Instruments, 2013) y lo mostrado en la Figura 6, el equipo soporta un sample rate máximo de 25 MS/s en cuadratura con una resolución de 16 bits, tanto para transmisión como para recepción. Esta variable se reutiliza en los bloques USRP Source, en la FFT y en los sinks para mantener coherencia en toda la cadena de procesamiento.

| Maximum I/Q sample rate ^[3] | |
|--|------------------------------|
| 16-bit sample width | 25 MS/s |
| 8-bit sample width | 50 MS/s |
| Digital-to-analog converter (DAC) | 2 channels, 400 MS/s, 16 bit |
| DAC spurious-free dynamic range (sFDR) | 80 dB |

Figura 6. Datasheet de máximo sample rate en cuadratura y fase.

Variable `vec_len`: define la longitud de vector de la FFT en este caso 1024 puntos. Se usa tanto en el bloque Stream to Vector como en el FFT y en los bloques posteriores, que esperan vectores de ese tamaño. Todo esto se relaciona también con la resolución de frecuencia, y esta se define por la Ecuación 2. Esa ecuación define la resolución utilizada tanto en el USRP2920 como en el analizador de espectros, con la finalidad de que ambas mediciones tuvieran concordancia y se lograran comparar.

QT GUI Range (freq, amp): son controles deslizantes que permiten modificar en tiempo real parámetros del flujo, como: freq: frecuencia central de trabajo. amp: amplitud de señal o factor de escala. Se emplean para hacer barridos de frecuencia y ajustes interactivos sin tener que modificar el diagrama ni reiniciar la ejecución.

UHD: USRP Source: es el bloque que se comunica con el USRP-2920: configura reloj, frecuencia central, tasa de muestreo, ganancia y antena. Produce una señal compleja en banda base (I/Q) a la tasa de muestreo especificada. En este flujo, la salida del USRP se divide, primero: Una rama va al QT GUI Frequency Sink, para ver el espectro en tiempo real. La otra rama se procesa numéricamente, FFT, magnitud, log10 y almacenamiento en archivo. Se utiliza porque es el punto de entrada de la señal real medida en el laboratorio, capturada por el SDR.

QT GUI Frequency Sink: muestra en pantalla el espectro en frecuencia de la señal compleja que viene del USRP, aplicando internamente una FFT, ventana y cálculo de magnitud. Permite verificar visualmente que la portadora generada, en este caso en 500, 590 o 660 MHz aparece donde se espera y con el nivel configurado. Se emplea como herramienta de diagnóstico inmediato, para confirmar que el montaje de transmisión/recepción está funcionando correctamente antes de procesar y guardar los datos.

Throttle: limita la velocidad de flujo de muestras a la tasa indicada, como se utilizó en este caso 12.5 MS/s. En diagramas sin hardware es esencial para que GNU Radio no consuma toda la CPU; con USRP no siempre es estrictamente necesario, pero aquí se mantiene para evitar que la parte de procesamiento y GUI sobrecargue el sistema. Se usa para controlar el ritmo de procesamiento y mantener estable la ejecución del flujo, especialmente cuando se añaden bloques de visualización y escritura a disco.

Stream to Vector: agrupa la secuencia de muestras complejas de entrada, stream, en vectores de longitud `vec_len`, en este caso 1024. La FFT de GNU Radio trabaja sobre vectores; por ello es necesario convertir el flujo continuo en bloques de tamaño fijo. Se emplea para preparar los datos en la forma adecuada para el bloque FFT, de modo que cada vector represente una instantánea espectral.

Bloque FFT: implementa la Transformada Rápida de Fourier (FFT) sobre cada vector de entrada. Convierte el conjunto de muestras de tiempo en sus componentes de frecuencia, entregando un vector complejo que contiene amplitud y fase por bin de frecuencia. Se puede configurar como Forward que es transformada directa y seleccionar el tipo de ventana (Hann, Blackman–Harris, etc.), que reduce el efecto de fugas espectrales. Se utiliza porque es el núcleo

del análisis espectral: permite pasar de la señal temporal medida por el USRP a la representación en frecuencia con resolución controlada por `vec_len`.

Fast Multiply Const: este bloque realiza la multiplicación de cada muestra del flujo de datos equivalente a cada bin espectral en el dominio de la frecuencia por una constante escalar. En el contexto del procesamiento espectral, dicha constante se selecciona como $\frac{1}{N}$, donde N corresponde a la longitud del vector de la FFT, con el fin de normalizar su salida. Esta normalización permite ajustar la escala del espectro, garantizando que los valores de amplitud o potencia obtenidos con posterioridad, al calcular la magnitud y su representación en decibelios, tengan una interpretación física coherente.

Complex to Mag²: este bloque transforma la salida compleja de la FFT normalizada en una magnitud al cuadrado, equivalente a una medida proporcional de la potencia espectral en cada bin de frecuencia. Su uso responde a que, para la caracterización del canal, el interés se centra en la distribución de potencia en frecuencia, prescindiendo de la información de fase, y permite obtener una representación en unidades lineales adecuada para el análisis posterior.

Log10: se usa para expresar la potencia en escala logarítmica (Ecuación 1). Esta representación es la forma estándar para niveles de señal en RF (por ejemplo, en dBm) y permite la comparación directa con la traza del analizador de espectro.

$$\text{Salida} = 10\text{Log}_{10}(P) \quad (1)$$

QT GUI Vector Sink: permite observar la forma exacta del espectro calculado después de la FFT, la normalización, el cálculo de potencia y la conversión a dB. Se utiliza como herramienta

visual complementaria al Frequency Sink y al archivo de texto: ayuda a verificar en tiempo real que el procesamiento numérico está produciendo la curva esperada.

Vector to Text File: este recibe cada vector de 1024 puntos en dB y lo escribe en un archivo de texto/CSV junto con la información de frecuencia central, sample rate, longitud de vector, dependiendo de la configuración. Se emplea para guardar los datos de la FFT y poder analizarlos posteriormente en Python: lectura de CSV, cálculo de correlación con la traza del analizador de espectros, análisis de ocupación espectral, etc. Es el puente entre GNU Radio y el procesamiento descrito a continuación.

Una vez obtenido cada archivo CSV, tanto del analizador de espectros como del flujograma de GNU Radio por medio del bloque Vector to Text File se implementó el siguiente algoritmo de Python, la lógica se puede ver en Figura 7:

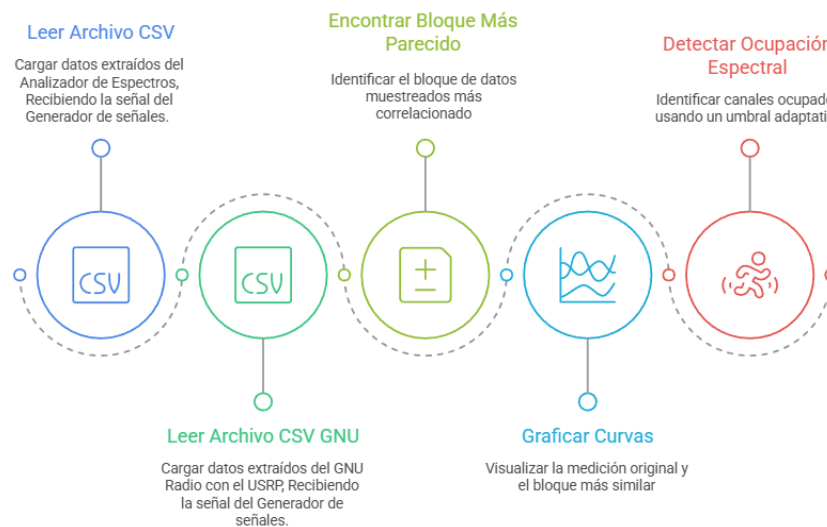


Figura 7. Lógica del código de Python para la comparación de archivos CSV.

A. 0.1 Algoritmo comparación de archivos CSV

El algoritmo implementado en Python para la lectura y comparación de los archivos CSV, también se puede encontrar en el repositorio de Github, (Rodríguez y Díaz, 2025) , se llama Generador vs Analizador.

1) Carga de librerías y definición de archivos

En esta sección se importan las bibliotecas principales: pandas para manejar datos tabulares, NumPy para operaciones numéricas vectorizadas, Matplotlib para las gráficas y re para expresiones regulares. Después se definen dos rutas de archivo: el CSV exportado por el analizador de espectros y el archivo de texto generado por GNU Radio que contiene los vectores con las muestras espectrales.

2) Lectura y normalización del archivo CSV del analizador

Primero se lee el CSV completo sin encabezados. Luego se busca la fila donde aparece la palabra “Frequency” en la primera columna; la fila siguiente marca el inicio de las mediciones. A partir de esa posición se extraen únicamente las dos primeras columnas, que corresponden a la frecuencia en Hz y a la magnitud en dBm, eliminando filas vacías. Ambas columnas se convierten explícitamente a tipo numérico y se descartan los valores no válidos. Finalmente, la columna de magnitudes se pasa a un arreglo NumPy y se normaliza a media cero y desviación estándar uno. Esta normalización hace que la comparación con los datos del SDR dependa de la forma del espectro y no del nivel absoluto.

3) Lectura del archivo de GNU Radio y construcción del eje de frecuencias

En este fragmento se abre el archivo de texto generado por GNU Radio y se almacenan únicamente las líneas no vacías. A partir de las líneas descriptivas se extraen tres parámetros clave: la frecuencia central de la medida, la tasa de muestreo y la longitud del vector FFT. Después se recorren de nuevo las líneas buscando aquellas que contienen solo números, espacios, puntos y comas; estas líneas corresponden a los bloques de datos numéricos. Cada una se separa en valores individuales, se convierte a flotante y se guarda en la lista `datos_txt`, donde cada elemento representa un vector de muestras espectrales. Finalmente, utilizando la frecuencia central y la tasa de muestreo se construye un eje de frecuencias equiespaciado que va desde: $f_c - F_s/2$ a $f_c + F_s/2$, donde f_c es frecuencia central, F_s es frecuencia de muestreo, con un número de puntos igual a la longitud del vector.

4) Búsqueda del bloque más parecido mediante correlación de Pearson

Aquí se importa la función Pearson de SciPy [1] para calcular el coeficiente de correlación de Pearson. Primero se verifica si la longitud de la referencia (`mag_ref`) coincide con la del vector FFT; en caso contrario, la referencia se re-muestrea mediante interpolación lineal para que ambas señales tengan el mismo número de puntos. Luego se recorre cada bloque de datos obtenido del SDR. Si el bloque tiene la longitud adecuada, se normaliza y se calcula la correlación de Pearson frente a la referencia re-muestreada. El código va guardando el índice y el valor de la correlación máxima; al final se imprime qué bloque es el más parecido al espectro original del analizador de espectros.

5) Generación de la gráfica comparativa entre analizador y SDR

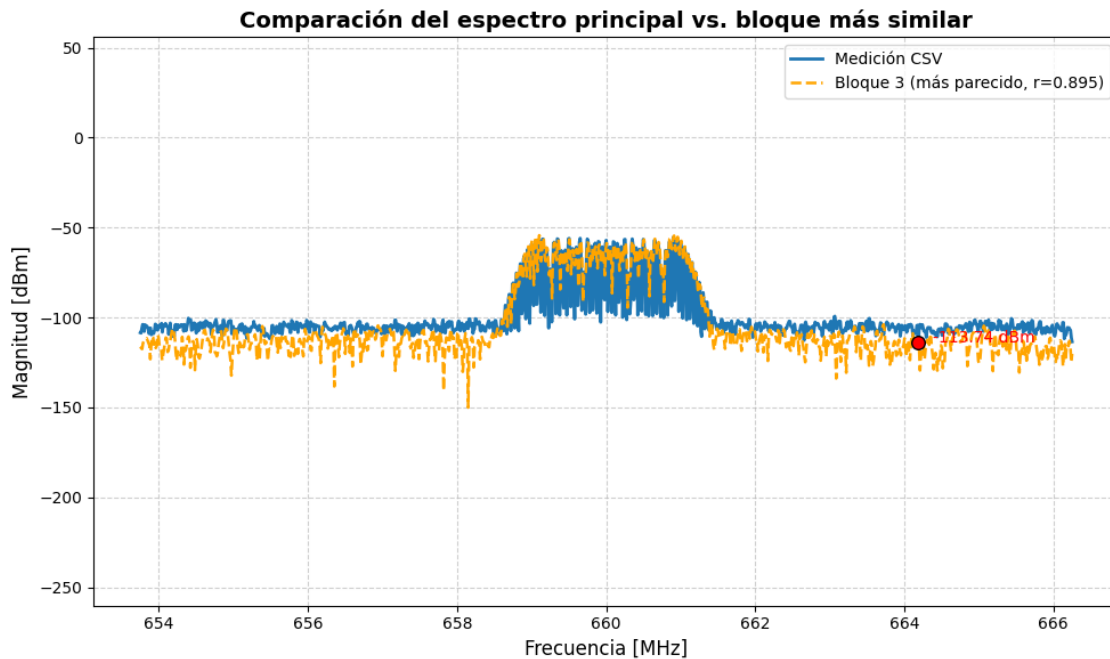
Este fragmento genera la figura de comparación entre ambas mediciones. Se crea una figura y se dibuja primero la curva procedente del analizador de espectros, usando la frecuencia en MHz y la magnitud en dBm. Si se encontró un bloque más parecido, se dibuja sobre la misma gráfica la curva correspondiente al SDR utilizando el eje de frecuencias calculado anteriormente con una línea discontinua. Finalmente se añaden título, etiquetas de ejes, rejilla y leyenda, y se ajustan los límites del eje vertical para dar un margen extra alrededor de la señal.

6) Detección de ocupación espectral y reporte de canales

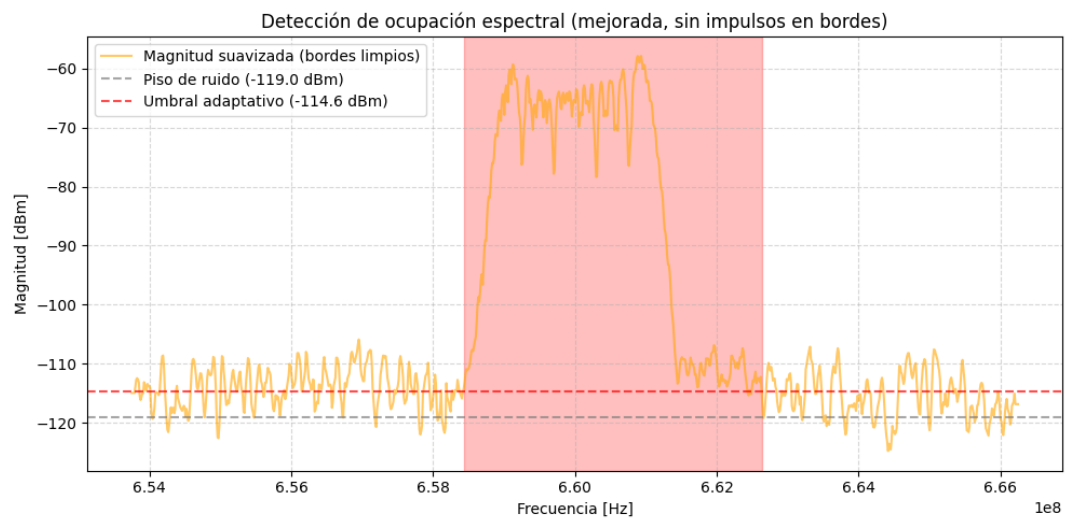
En el último bloque se toma el vector correspondiente al bloque más parecido y se analiza la ocupación espectral. Se definen varios parámetros ajustables, como el tamaño de la ventana de suavizado, el factor k para el umbral, el ancho mínimo de canal y las tolerancias en puntos y huecos, suavizando las muestras con un promedio móvil. A partir de las muestras de menor potencia se estima el piso de ruido y su desviación estándar, y se calcula un umbral adaptativo tipo piso. Luego se construye una máscara que marca los puntos por encima del umbral y se agrupan en tramos continuos, permitiendo pequeños huecos internos. Cada tramo suficientemente ancho se interpreta como un canal ocupado, y sus rangos de frecuencia se almacenan en la lista canales_ocupados. Finalmente se genera una gráfica con la magnitud suavizada, el piso de ruido, el umbral y las bandas ocupadas sombreadas, y se imprime en consola un resumen del número de canales detectados y sus rangos en MHz. Se tomo como referencia el paper (Gorcin & Arslan, 2010).

RESULTADOS Y HALLAZGOS

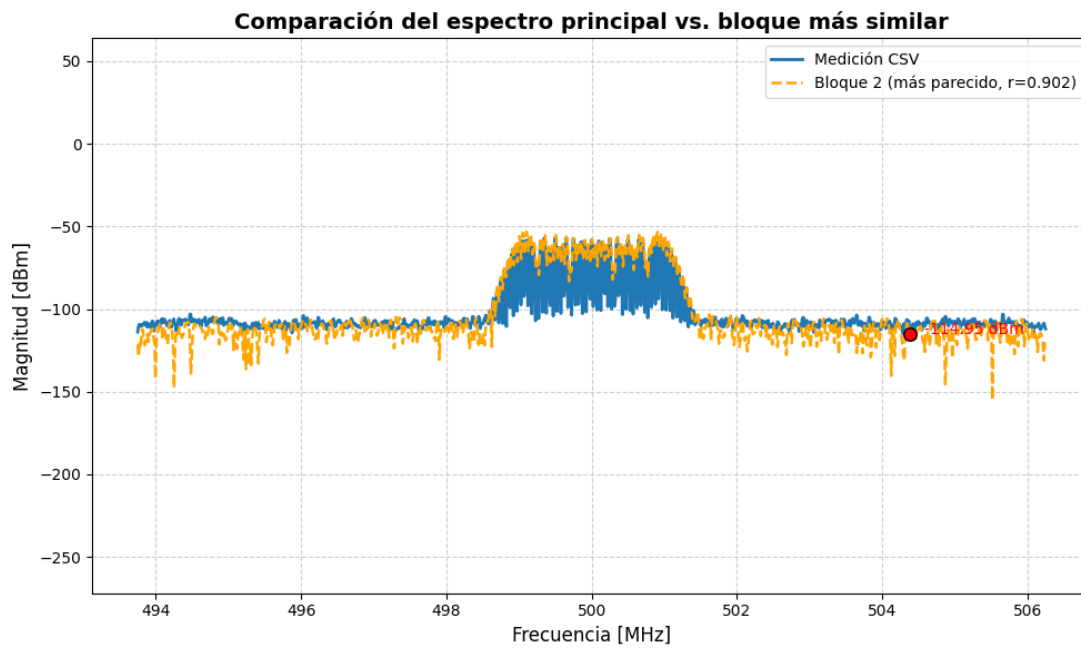
Figura 8. Analizador de espectros vs USRP2920.



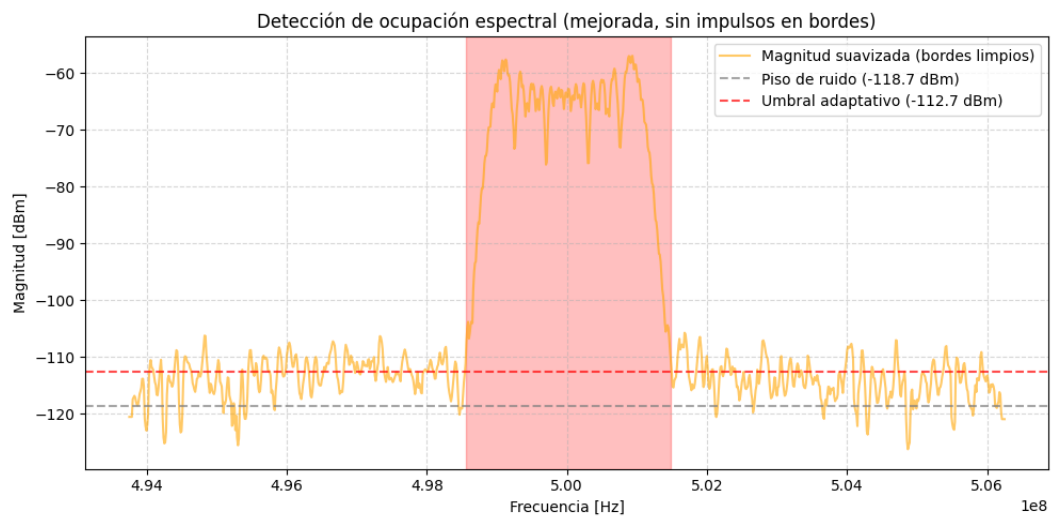
a) Analizador de espectros vs USRP2920 660MHz.



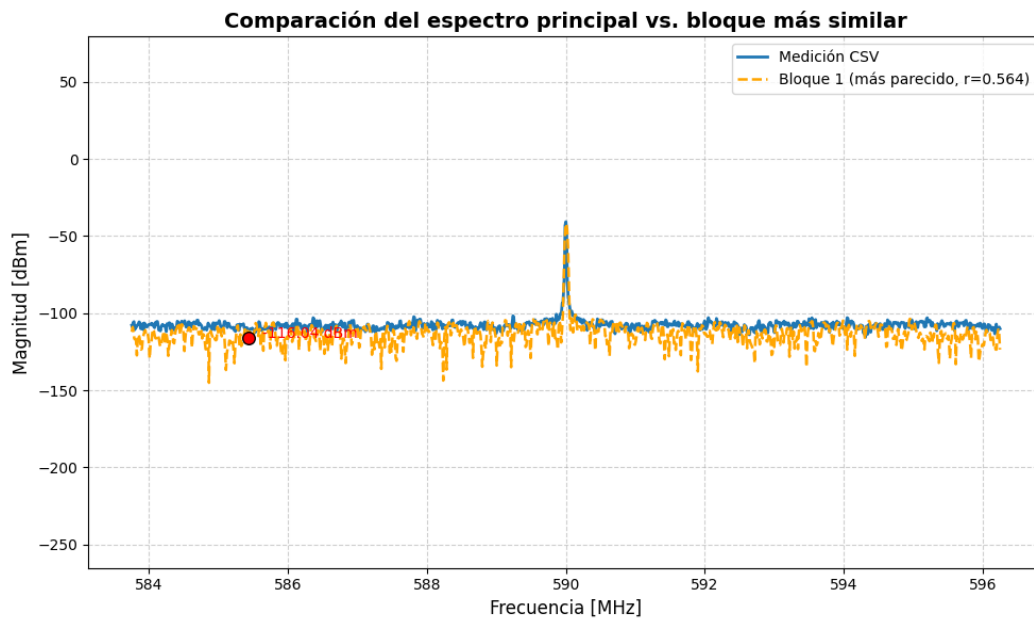
b) Detección de ocupación frecuencia 660 MHz.



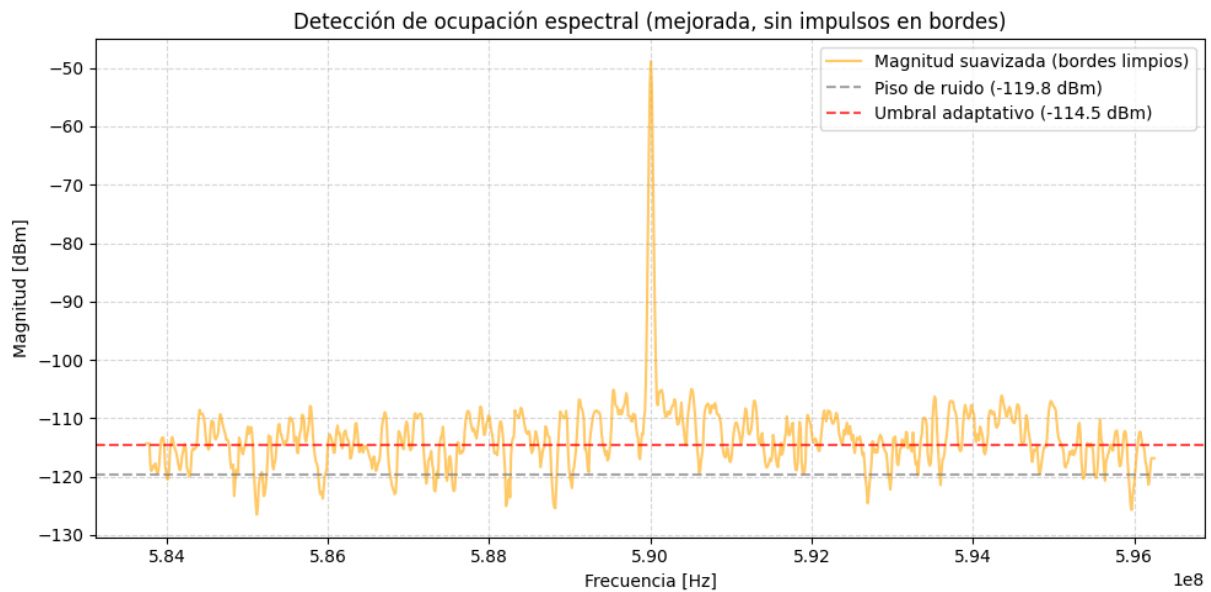
c) Analizador de espectro vs USRP2920 500MHz.



d) Detección de ocupación frecuencia 500MHz.



e) Analizador de espectro vs USRP2920 590 MHz.



f) Detección de ocupación frecuencia 590MHz.

| Frecuencia [MHz] | Coefficiente de correlación (r) | Interpretación cualitativa |
|---------------------|------------------------------------|--------------------------------|
| 660 | 0.902 | Correlación muy alta |
| 500 | 0.895 | Correlación alta / muy alta |
| 590 | 0.564 | Correlación moderada |

Tabla 1. Correlación de cada frecuencia

CONCLUSIONES Y RECOMENDACIONES

A partir del procedimiento experimental y de procesamiento descrito en este apartado, se pueden extraer las siguientes conclusiones:

1. Equivalencia práctica entre el analizador de espectros y la cadena SDR

Los coeficientes de correlación obtenidos entre la traza del analizador de espectros y la generada por el USRP-2920 con GNU Radio muestran una concordancia elevada para dos de las tres frecuencias analizadas: $r = 0,902$ para 660 MHz y $r = 0,895$ para 500 MHz, lo que se interpreta como una correlación alta. Estos resultados indican que, con una configuración adecuada de frecuencia central, tasa de muestreo, longitud de FFT y ventana, la cadena SDR reproduce de manera consistente el espectro medido por el instrumento de laboratorio.

2. Comportamiento particular a 590 MHz

En el caso de 590 MHz, la correlación obtenida ($r = 0,564$) es moderada, lo cual sugiere la presencia de discrepancias pequeñas entre ambas mediciones, ya sea por ruido y estabilidad de la señal durante la captura. Lejos de invalidar el método, este resultado resalta la importancia de

considerar la influencia del entorno espectral y de la configuración fina de parámetros en la calidad de la medición.

3. Estabilidad y coherencia del montaje de laboratorio

Las mediciones realizadas en el laboratorio evidencian que, para las tres frecuencias, la cadena completa generador–analizador–SDR se comportó de manera estable y coherente: la portadora apareció en la posición esperada, con el nivel configurado y sin distorsiones significativas en el entorno inmediato de la frecuencia central. Esto indica que, en condiciones controladas, el hardware utilizado (generador de RF, analizador de espectros y USRP-2920) es capaz de reproducir de forma fiable el escenario de transmisión y recepción que posteriormente se desea extrapolar a campo.

[1] SciPy Developers. (2026). `scipy.stats.pearsonr` — SciPy v1.17.0 Manual. SciPy. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html>