

# **COMUNICACIONES**

Laboratorio N°1



FACULTAD  
DE INGENIERÍA

Dolores Garro

Legajo: 03115/9

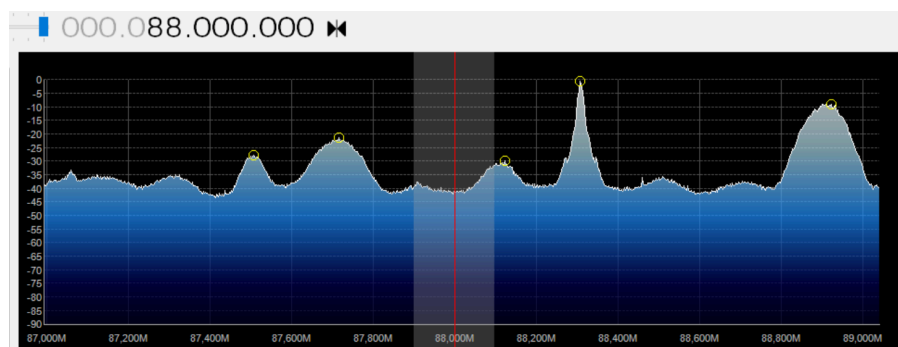
Ingeniería en Computación

## Introducción

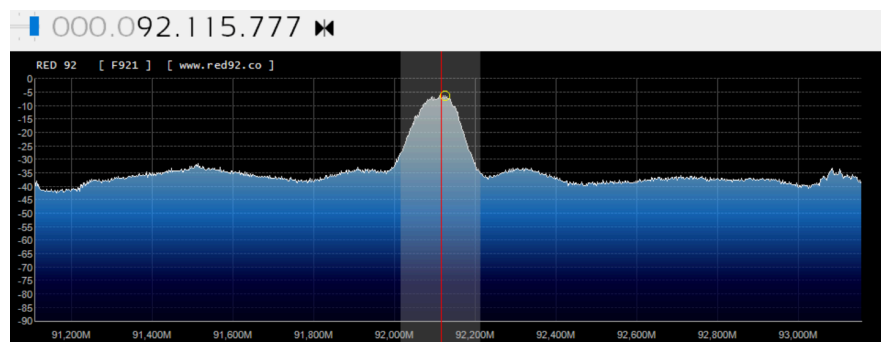
Para dar inicio al análisis de las señales de FM, se realizaron una serie de pasos utilizando el software *SDR#* y un *dongle RTL-SDR*. En primer lugar, fue necesario instalar los controladores del dispositivo mediante la herramienta *zadig.exe*, asegurando así su correcta conexión y configuración con la computadora. Una vez lista esta configuración, el dispositivo se conectó al software *SDR#*, permitiendo realizar un escaneo en tiempo real del espectro de FM comercial.

Posteriormente, se utilizó la función de barrido del espectro en *SDR#* para explorar la banda de FM comercial. Este proceso involucró identificar estaciones de radio con mayor potencia dentro del espectro, empleando pasos de 100 kHz para facilitar la visualización de las señales más destacadas. Al seleccionar la opción WFM en el software, fue posible escuchar las transmisiones de audio a través de auriculares.

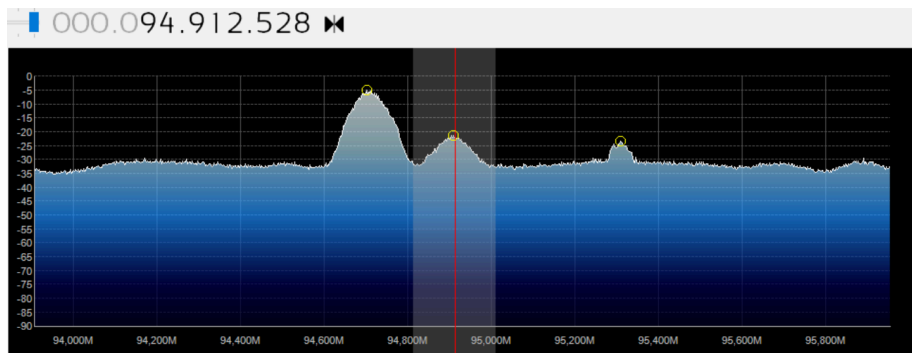
A continuación, en las siguientes figuras, se presentan algunos picos de frecuencia identificados en el análisis, los cuales se interpretaron como representativos de estaciones de radio activas en la banda de FM.



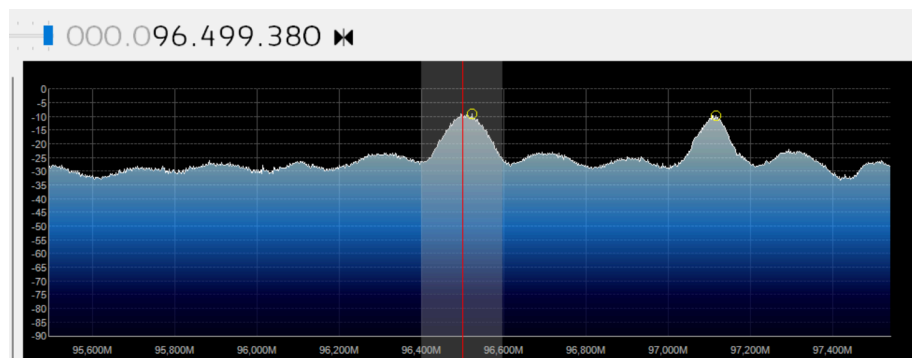
**Figura 1:** Espectro alrededor de 88 MHz



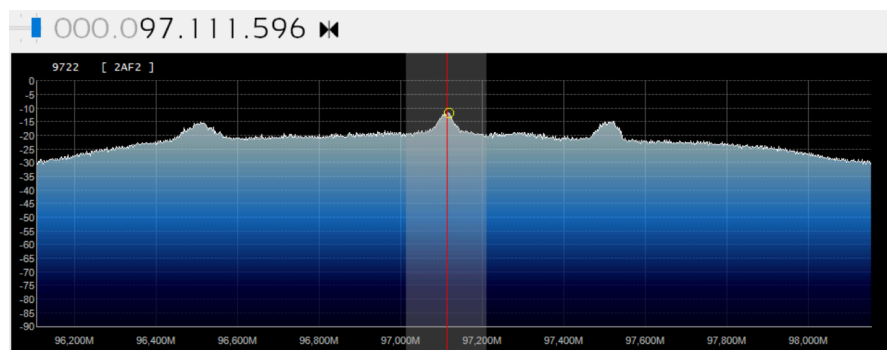
**Figura 2:** Espectro alrededor de 92 MHz



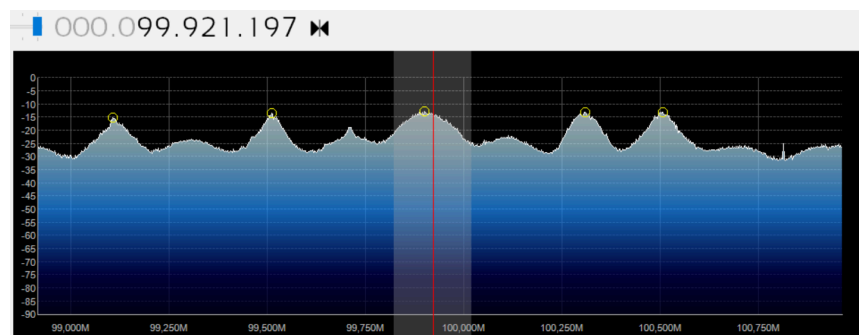
**Figura 3:** Espectro alrededor de 94 MHz



**Figura 4:** Espectro alrededor de 96.5 MHz



**Figura 5:** Espectro alrededor de 97 MHz



**Figura 6:** Espectro alrededor de 99 MHz

Se llevó a cabo un barrido completo del espectro hasta alcanzar los 108 MHz, límite superior de la banda comercial de FM.

La configuración del ancho de banda del filtro tiene un impacto directo en la calidad de la señal recibida y reproducida. Al reducir el ancho de banda, se observa una disminución en la capacidad del sistema para capturar componentes frecuenciales esenciales, lo que genera distorsiones en el audio. Este fenómeno demuestra la importancia de seleccionar un ancho de banda adecuado que equilibre la exclusión de ruido no deseado y la preservación de la información relevante para el mensaje.

Por otro lado, el control de ganancia, ya sea automático o manual, es importante en la recepción y representación de señales en el espectro. La activación del control automático de ganancia (AGC) asegura una adaptación dinámica a las condiciones cambiantes de la señal, lo que simplifica su manejo en la mayoría de los casos.

### **Uso del *Dongle* como proveedor de muestras**

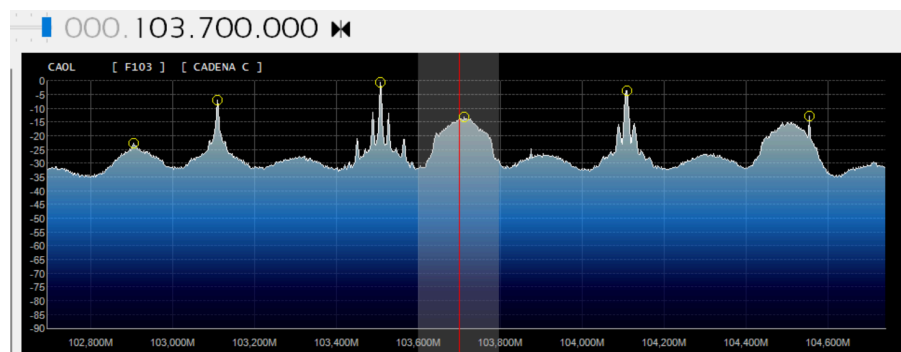
En este apartado, se empleará el dongle SDR como fuente de señales para su procesamiento digital, con el objetivo de explorar las capacidades de este dispositivo en la captura y análisis de señales de radiofrecuencia. El propósito principal es capturar un registro de muestras provenientes de una estación de radio conocida y, luego procesar estos datos para extraer y escuchar el audio correspondiente. Este proceso requiere un flujo de trabajo que incluye la configuración adecuada del dongle, la captura de las señales en formato digital, su visualización, y el desarrollo de funciones que permitan analizar y depurar los resultados.

Para llevar a cabo este ejercicio, se trabaja con muestras capturadas durante un periodo aproximado de 10 segundos, utilizando una frecuencia de muestreo de 2.048 MHz y centrando la recepción en una frecuencia específica, como 103.7 MHz o 96.5MHz (Ver Figuras 7 y 8). Se hace énfasis en la construcción de herramientas personalizadas que faciliten el análisis espectral y permitan la estimación de la densidad espectral de potencia (DEP).

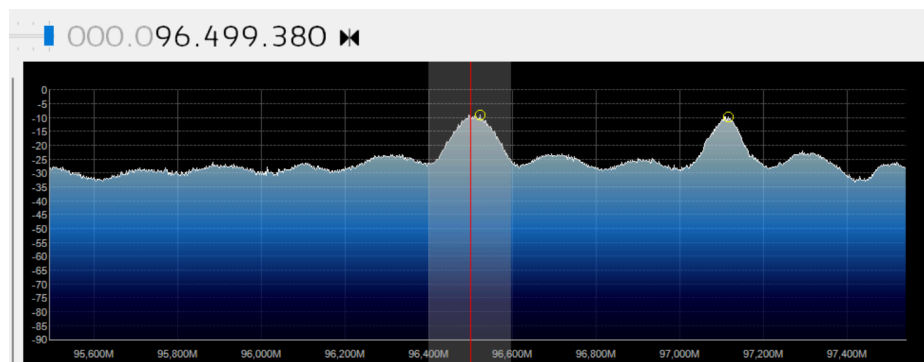
El procesamiento de las muestras puede realizarse directamente en Matlab, siguiendo las pautas establecidas por la cátedra, o mediante el uso de herramientas alternativas, como la librería *rtl-sdr* para lenguajes como Python. En este caso, se optó por utilizar Python debido a una mayor experiencia previa en dicho entorno, lo cual facilita la implementación y comprensión de los procedimientos requeridos. Este enfoque permitió capturar las señales de manera eficiente y almacenarlas en archivos para su posterior análisis y visualización.

Como parte del desarrollo, se incluirá la implementación de una función para la estimación de la DEP, utilizando conceptos aprendidos en ejercicios previos. Esta función

permitirá analizar el espectro de la señal capturada y evaluar su contenido, lo que constituye un paso fundamental para avanzar hacia el procesamiento y reproducción del audio.



**Figura 7:** Espectro alrededor de 103.7 MHz



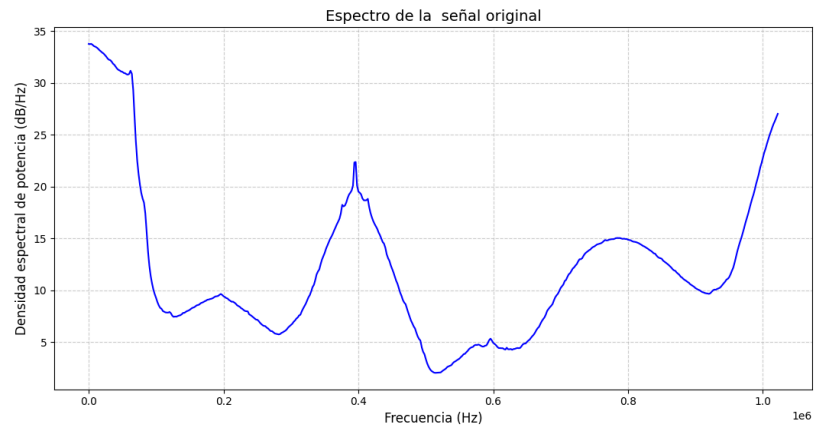
**Figura 8:** Espectro alrededor de 96.5 MHz

## Análisis espectral y estimación de la DEP

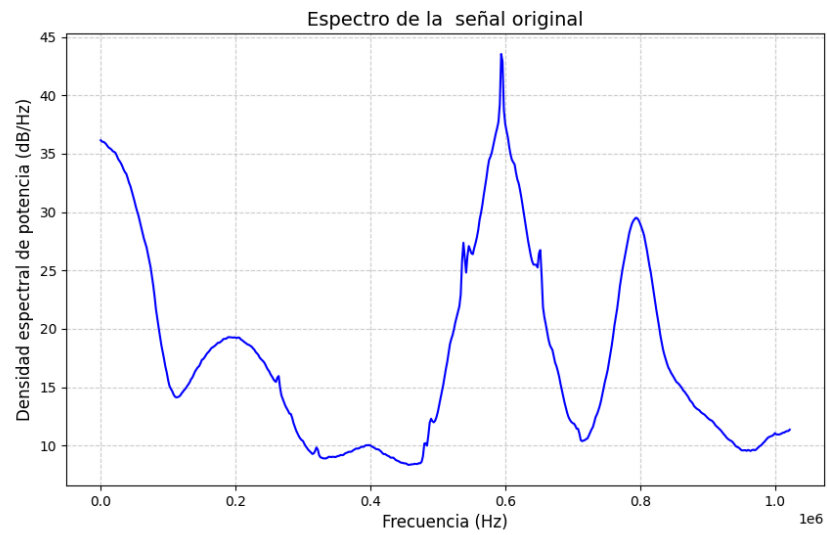
En esta etapa del trabajo, se utilizó el vector de muestras obtenido para realizar un análisis espectral detallado. Este procedimiento incluyó la implementación de una función en Python para estimar y graficar la *Densidad Espectral de Potencia* (DEP). El análisis espectral se realiza para observar las características de las señales capturadas, permitiendo identificar estaciones de radio y evaluar el comportamiento del espectro en función de las frecuencias seleccionadas.

La función desarrollada en Python para calcular la DEP y graficarla permite observar de manera detallada las variaciones de potencia en función de la frecuencia, facilitando la identificación de picos correspondientes a estaciones de radio.

En las figuras, se observan los espectros obtenidos para dos frecuencias de interés

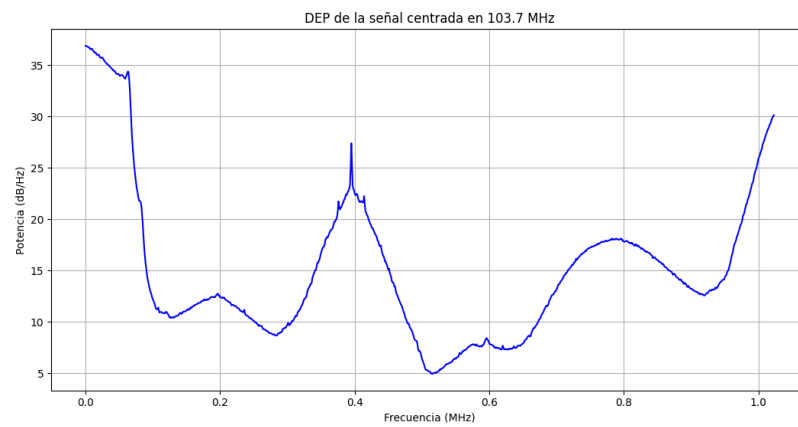


**Figura 9:** Espectro centrado en 103.7 MHz

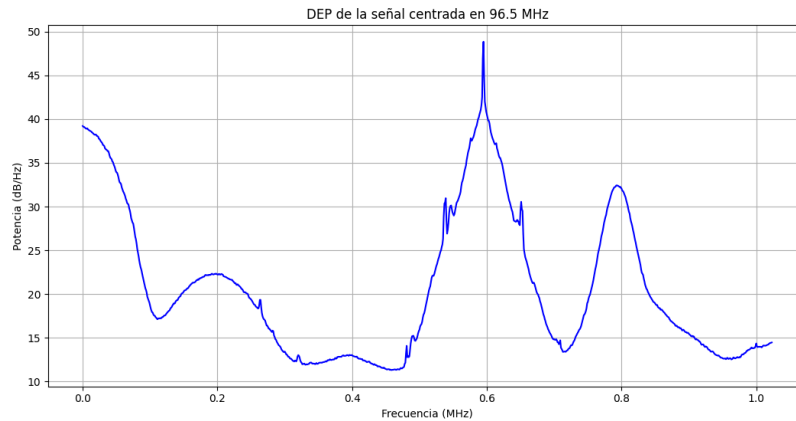


**Figura 10:** Espectro centrado en 96.5Mhz

A continuación, se observa la DEP calculada con la función desarrollada en Python



**Figura 11: DEP 103.7Mhz**



**Figura 12: DEP 96.5Mhz**

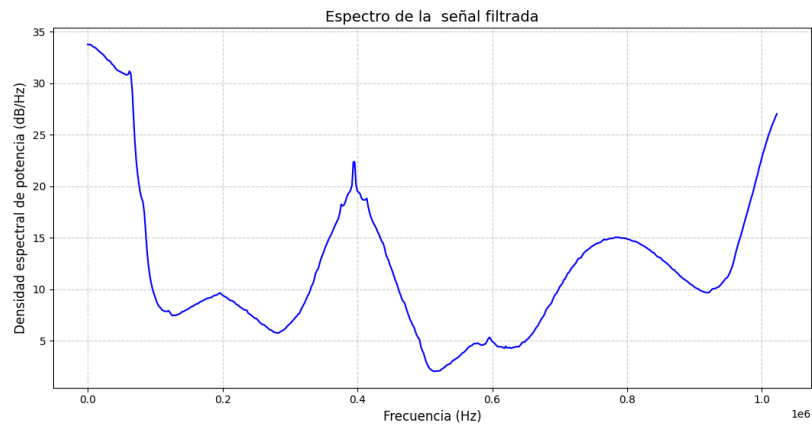
Ambos gráficos (Figura 11 y 12) representan la Densidad Espectral de Potencia (DEP) de las señales adquiridas, centradas en 103.7 MHz y 96.5 MHz, respectivamente. Estas representaciones permiten visualizar cómo se distribuye la potencia de las señales en el dominio de frecuencia, facilitando el análisis del espectro y proporcionando una herramienta para el diseño y ajuste de filtros y demoduladores en pasos posteriores.

## Filtrado y análisis espectral detallado

En este ejercicio, se desarrolló una función para filtrar la Densidad Espectral de Potencia obtenida en el ejercicio anterior. El filtro utilizado es un filtro Butterworth de orden 5, el cual nos permite eliminar señales no deseadas y aislar la estación de radio seleccionada. El filtrado se realizó considerando el ancho de banda de Carson para FM comercial, lo que asegura que solo la estación deseada permanezca en el espectro, mientras que las interferencias de otras señales fuera del rango de interés se atenúen.

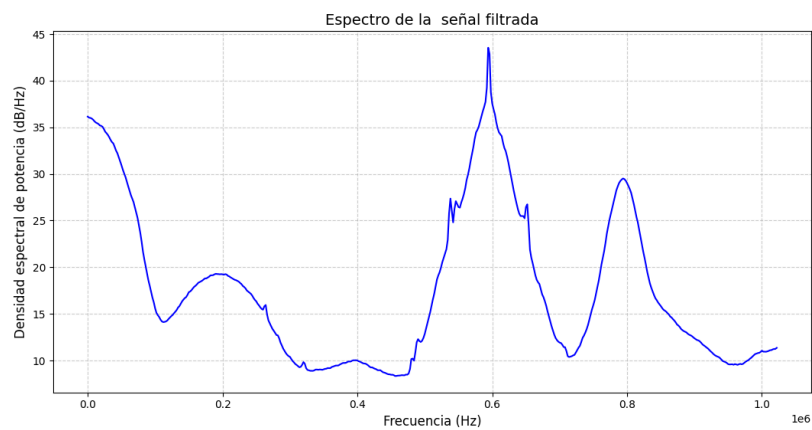
Luego del filtrado, se utilizó la función previamente diseñada para estimar y graficar la DEP de la señal filtrada. Esto permitió ver el comportamiento espectral de la estación de radio seleccionada y confirmar la efectividad del filtrado.

La Figura 13 muestra el espectro filtrado centrado en la frecuencia de 103.7 MHz. Se observa que la señal de la estación seleccionada se destaca claramente, mientras que las señales no deseadas han sido eliminadas.



**Figura 13:** Espectro filtrado centrado en 103.7Mhz

La Figura 14 muestra el espectro filtrado centrado en la frecuencia de 96.5 MHz.



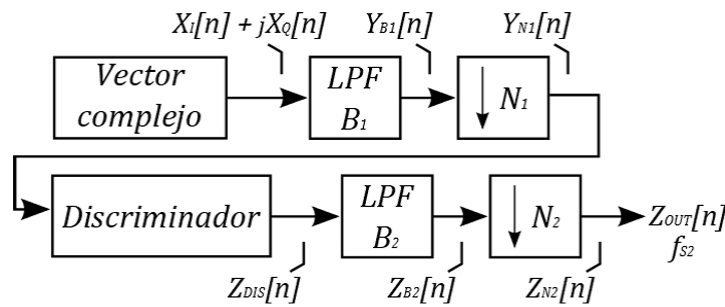
**Figura 14:** Espectro filtrado centrado en 96.5Mhz

Al igual que en la figura anterior, el filtrado ha aislado correctamente la estación de radio deseada.



## Demodulador de FM por procesamiento digital

El esquema de la Figura 15 representa un demodulador digital de señales de FM. Este diseño se utiliza para recuperar el mensaje modulado en la señal portadora adquirida previamente. La demodulación se realiza mediante el uso de bloques funcionales que permiten filtrar, reducir la tasa de muestreo y extraer la información del mensaje. A continuación, se describen los elementos del esquema y su implementación en el código.



**Figura 15:** Esquema de un demodulador de FM por procesamiento digital

### Etapas 1: Vector complejo

En principio, el vector complejo representa las muestras de la señal FM adquirida por el dongle SDR. Cada muestra contiene un componente en fase (parte real) y otro en cuadratura (parte imaginaria). Fueron tomadas utilizando un dongle SDR configurado para capturar señales en la frecuencia central de 96.5 MHz con una frecuencia de muestreo de 2.048 MHz. Para ello, se estableció una duración de 10 segundos, dividiendo los datos en *frames* de  $256 * 64$  muestras cada uno, lo que permitió gestionar eficientemente la captura de los datos. Los *frames* fueron capturados iterativamente y almacenados en un *buffer*, que posteriormente se concatenó para formar un *vector* continuo de muestras complejas. Estas muestras representan las componentes en fase y cuadratura (IQ) de la señal de FM adquirida.

Finalmente, las muestras fueron almacenadas en el archivo *muestras\_capturadas\_96\_5.npy* para su posterior procesamiento y análisis, asegurando así la preservación de los datos originales para replicar o extender el análisis en las próximas etapas.

## Etapa 2: Filtro pasabajos (LPF B1)

Este bloque aplica un filtro pasa bajos con un ancho de banda B1, que corresponde al rango de frecuencia de la estación de interés. Su objetivo es eliminar componentes de otras estaciones o ruido fuera del espectro de interés.

En el código, el filtro se implementa mediante la función *aplicar\_filtro\_butter*, que utiliza un filtro Butterworth de orden 5. La frecuencia de corte B1 es calculada según el ancho de banda de Carsson para FM comercial.

## Etapa 3: Diezmado NI

El objetivo de esta etapa es reducir la tasa de muestreo de la señal filtrada para disminuir la cantidad de datos procesados. El factor de diezmado NI se elige para que la señal diezmada conserve toda la información de la estación.

La elección de  $NI = 10$  como factor de diezmado inicial se basa en criterios de eficiencia y en el cumplimiento del teorema de muestreo de Nyquist. La frecuencia de muestreo original del sistema es  $f_s = 2.048 \text{ MHz}$ , significativamente mayor que el ancho de banda de interés de la señal, definido por el filtro paso bajo inicial ( $B1 = 200 \text{ kHz}$ ). Al aplicar el diezmado, se reduce la frecuencia de muestreo a  $f_{s1} = f_s / NI = 204.8 \text{ kHz}$ , lo cual sigue siendo suficiente para capturar adecuadamente la información relevante de la señal filtrada, dado que cumple con el criterio de Nyquist ( $f_{s1} \geq 2B1$ ). Esta reducción permite mejorar la eficiencia del procesamiento al disminuir la cantidad de muestras a procesar sin comprometer la calidad de la señal.

## Etapa 4: Discriminador

El discriminador permite extraer el mensaje original contenido en las variaciones de frecuencia de la señal. En la modulación FM, el mensaje original no está directamente presente en la amplitud ni en la frecuencia constante de la portadora. En lugar de eso, se codifica variando la frecuencia instantánea de la señal portadora en proporción al mensaje original. Por lo tanto, el objetivo principal de esta etapa es calcular la derivada de la fase para obtener el mensaje integrado.

Para lograrlo, se sigue el modelo descrito en el enunciado (Ver Figura 14)

$$m[n] \approx \frac{1}{2\pi f_d} \frac{(\phi[n] - \phi[n-1])}{T}$$

**Figura 14:** Relación entre fase y mensaje en FM

La implementación de este cálculo en el código se llevó a cabo a partir de las funciones de la librería *numpy*:

<code>phase = np.unwrap(np.angle(signal))</code>	Obtiene la fase de la señal diezmada
<code>delta_phase = np.diff(phase)</code>	Asegura que la fase sea continua, eliminando los saltos de $2\pi$
<code>demodulated_signal = delta_phase / (2 * np.pi * fd * T)</code>	Calcula las diferencias entre muestras consecutivas, aproximando la derivada

Finalmente, se normaliza el resultado dividiendo por  $2\pi f_d$ , lo que permite convertir las variaciones de fase en la señal banda base correspondiente al mensaje integrado.

Este cálculo es fundamental para traducir las variaciones de frecuencia, que representan los datos originales, en una señal interpretable para su procesamiento o reproducción posterior.

Más adelante en este informe se abordará un análisis detallado del espectro correspondiente a la señal discriminada, destacando sus principales características.

### **Etapas 5: Filtro pasabajos (LPF B2)**

La quinta etapa del proceso corresponde a la aplicación de un segundo filtro pasabajos (Low Pass Filter, LPF), denominado B2. Su propósito principal es eliminar las componentes de alta frecuencia que puedan haberse generado durante la etapa del discriminador de frecuencia. Estas componentes, generalmente introducidas por ruido o distorsiones derivadas del proceso de derivación y cálculo de la fase, no forman parte del mensaje de interés y, de no ser eliminadas, podrían degradar la calidad del mensaje final.

En el código, esta etapa se implementa, al igual que la etapa 2, mediante el uso de la función *aplicar\_filtro\_butter*, que emplea un filtro digital Butterworth tipo pasabajos, evitando que frecuencias no deseadas alteren las características del mensaje de audio.

El resultado de esta etapa es una señal filtrada que contiene únicamente el contenido útil del mensaje demodulado, libre de ruido y componentes de alta frecuencia.

### **Etapas 6: Diezmado $N_2$**

En esta etapa del demodulador FM, se realiza un *segundo diezmado* sobre la señal filtrada para ajustar su frecuencia de muestreo ( $f_s$ ) a un valor adecuado para la reproducción

como audio. Este proceso permite optimizar el manejo de datos, ya que la frecuencia de muestreo original ( $f_s=2.048\text{ MHz}$ ) es significativamente mayor de lo necesario para señales de audio.

El factor de diezmado  $N_2 = 5$  se seleccionó para reducir la frecuencia de muestreo de  $f_{s1}$ , a un valor dentro del ancho de banda de las señales de audio típicas que es aproximadamente 15 KHz.

Dado que

$$f_{s2} = \frac{f_{s1}}{N_2}$$

Y sabiendo que

$$f_{s1} = \frac{f_s}{N_1} = \frac{2.048\text{ MHz}}{10} = 204.8\text{ kHz}$$

tenemos

$$f_{s2} = \frac{204.8\text{ kHz}}{5} = 40.96\text{ kHz}$$

Esta frecuencia es adecuada para señales de audio, ya que cumple con los requerimientos del ancho de banda de audio (15 kHz) y se aproxima al estándar de 44.1 kHz utilizado por dispositivos de reproducción.

En el código, el diezmado se realizó utilizando la función `decimate`, la cual no solo reduce la tasa de muestreo, sino que también aplica un filtro antialiasing integrado para evitar el aliasing que puede ocurrir al reducir la frecuencia de muestreo sin un filtro adecuado.

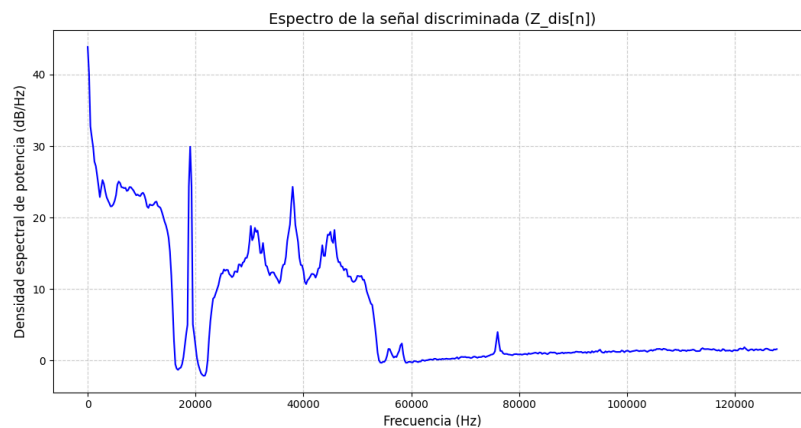
## Reproducción del mensaje demodulado

La etapa final del procesamiento de la señal discriminada se centra en que el mensaje demodulado sea apto para su reproducción audible. Para ello, luego del filtrado que elimina cualquier componente de alta frecuencia residual, se normaliza la señal para que su amplitud máxima sea 1, lo cual previene saturaciones y asegura un nivel de audio adecuado para la reproducción. La implementación de estas etapas no sólo valida el correcto funcionamiento del algoritmo desarrollado, sino que también permite que la señal demodulada pueda ser auditivamente verificada a través de la función de reproducción.

En el código desarrollado, la verificación del correcto funcionamiento del algoritmo se implementa utilizando la función `sd.play()` de la biblioteca `sounddevice`, que reproduce la señal de salida `z_out` a la frecuencia de muestreo ajustada  $f_{s2}$ .

## Análisis del espectro de la señal discriminada

El análisis del espectro de la señal discriminada muestra las principales componentes frecuenciales presentes después del proceso de discriminación (Ver Figura 16).



**Figura 16:** Espectro  $Z_{dis}$

En este espectro, es posible observar cómo se distribuye la energía de la señal discriminada en distintas frecuencias, destacando tanto las componentes relevantes como las no deseadas. La mayor parte de la información útil se encuentra concentrada en las frecuencias más bajas, reflejando el mensaje de audio que se transmite. Sin embargo, también se pueden identificar picos de alta frecuencia que probablemente correspondan a ruido o interferencias no deseadas, las cuales podrían haber sido introducidas durante el proceso de demodulación. Estas componentes fuera de banda serán eliminadas en etapas posteriores del procesamiento, asegurando que solo se conserve la información pertinente, lo que contribuirá a mejorar la calidad del audio final.

Por otro lado, se destaca en el espectro un pico en 19 kHz, correspondiente al *tono piloto* característico de las transmisiones FM estéreo. Un *tono piloto* es una señal de frecuencia fija que se transmite junto con la señal principal en un canal de comunicación. Este tono, aunque independiente de la información útil, desempeña un papel fundamental en la referencia, control y monitorización del sistema de transmisión. Al ser una señal constante y conocida, el tono piloto permite al receptor realizar diversas funciones críticas para el correcto funcionamiento del sistema.

En primer lugar, el tono piloto actúa como una referencia de funcionamiento. Su presencia en el canal confirma que el transmisor está activo y operando correctamente.

En segundo lugar, es empleado como herramienta de control. Por ejemplo, en sistemas de audio y telecomunicaciones, su presencia permite al receptor diferenciar entre estados operativos, como la transmisión de una señal o el silencio. En ausencia del tono piloto, el receptor puede activar automáticamente una función de *mute* para evitar la reproducción de ruido indeseado, mejorando la experiencia del usuario final.

Por último, el tono piloto facilita la monitorización del sistema de comunicación. Al ser una señal de frecuencia constante y predefinida, puede ser utilizada para calibrar equipos de recepción, medir parámetros de transmisión o verificar la calidad del canal. En el caso específico de las transmisiones de FM estéreo, el tono piloto a 19 kHz desempeña un rol adicional al sincronizar el proceso de decodificación estéreo, *garantizando que los canales izquierdo y derecho sean separados correctamente.*

## Conclusiones

En el laboratorio, se logró realizar un análisis de las señales de FM utilizando un dongle RTL-SDR y el software SDR#. A través del barrido del espectro, se identificaron diversas estaciones de radio en la banda comercial de FM. Además, se desarrollaron herramientas en Python para estimar y graficar la DEP de las señales para identificar las estaciones de radio y evaluar el comportamiento del espectro en función de las frecuencias seleccionadas. La implementación de un filtro Butterworth y técnicas de diezmado contribuyó a mejorar la calidad de la señal y facilitar la posterior demodulación de las señales FM.

En cuanto a la demodulación digital, se logró extraer el mensaje original de las señales FM utilizando discriminadores y filtros pasabajos. Al final, la reproducción del mensaje demodulado verificó que el sistema funciona como se esperaba, confirmando que las señales capturadas pueden ser procesadas y escuchadas sin inconvenientes.

En resumen, este laboratorio permite entender cómo se capturan, procesan y demodulan las señales FM, destacando lo importante que es configurar bien el sistema y usar las herramientas correctas para obtener buenos resultados en la transmisión digital de radio.

## Bibliografía

- Sierra Pérez, M., et al. (2003). *Electrónica de Comunicaciones*. Pearson. Prentice Hall.
- Scribd <https://es.scribd.com/>
- Wikipedia [https://es.wikipedia.org/wiki/Modulaci%C3%B3n\\_\(telecomunicaci%C3%B3n\)](https://es.wikipedia.org/wiki/Modulaci%C3%B3n_(telecomunicaci%C3%B3n))
- NumPy: Oliphant, T. E. (2006). A guide to NumPy. Trelgol Publishing.
- RTL-SDR: RTL-SDR Blog. (2023). RTL-SDR - The RTL-SDR Blog.  
<https://www.rtl-sdr.com/>
- Sounddevice: PySoundDevice. (2023). *Sounddevice: Play and record sound with Python*.  
<https://python-sounddevice.readthedocs.io/>