

PSD DE SEÑALES ALEATORIAS (GNURADIO)

Santiago Andrés Ibañez Ramírez
2211258

santiago2211258@correo.uis.edu.co

Nicolás David Martínez Cristancho
2212269

nicolas2212269@correo.uis.edu.co

Jonathan Mauricio Rojas Rodriguez
2212280

jonathan2212280@correo.uis.edu.co

Escuela de Ingeniería Electrónica
Universidad Industrial De Santander
Bucaramanga, Colombia

Repositorio Github

Resumen—This report details the third laboratory practice focused on Power Spectral Density (PSD) analysis of random signals using GNU Radio. The study explores the spectral characteristics of bipolar binary signals, white noise, and real-world data from cameras and microphones. By varying parameters such as samples per symbol (Sps), configuring GNU Radio blocks (e.g., interpolation filters, virtual sources), and implementing custom functions, the PSD behavior under different scenarios is evaluated. The methodology emphasizes the use of GitHub for collaborative development, signal processing workflows, and spectral resolution calculations. Results highlight the relationship between signal bandwidth, modulation schemes, and spectral efficiency, demonstrating the practical relevance of PSD in communication systems.

I. INTRODUCCIÓN

La Densidad Espectral de Potencia (PSD) es una herramienta clave en el análisis de señales, al cuantificar cómo se distribuye la energía de una señal en el dominio frecuencial. Su relevancia en telecomunicaciones radica en optimizar el uso del ancho de banda, identificar interferencias y diseñar esquemas de modulación eficientes. En sistemas modernos, como redes 5G o radio definida por software, el análisis de la PSD permite caracterizar canales de comunicación, mejorar la calidad de transmisión y garantizar la integridad de la información, incluso en entornos con ruido o limitaciones espectrales.

Esta práctica se centra en explorar la PSD de señales aleatorias, contrastando señales binarias sintéticas con datos reales (imágenes y audio). Mediante la variación de parámetros como Samples per Symbol (Sps) y la adaptación de flujogramas para emular codificaciones (RZ, Manchester) y modulaciones (OOK, BPSK), se analiza cómo la configuración de pulsos y filtros impacta el espectro. Además, se aborda el desafío de procesar fuentes heterogéneas, donde factores como la variabilidad estadística en imágenes o audio exigen ajustes metodológicos, reforzando la conexión entre teoría y aplicación en el diseño de sistemas robustos.

II. METODOLOGÍA

En cuanto al procedimiento realizado para la ejecución de la segunda práctica de laboratorio se procedió como sigue:

II-A. Configuración inicial y gestión colaborativa

La primera parte del laboratorio constó en crear directorios donde se guardaron los archivos y los avances del informe,

luego de esto se crearon las ramas de cada uno de los integrantes del grupo para subir sus avances individuales.

II-B. Generación y análisis de señales sintéticas

Se implementó un flujograma base para crear señales binarias bipolares, variando parámetros como la cantidad de muestras por símbolo (Sps). Esto permitió observar cambios en la forma de onda temporal y su impacto en la distribución espectral, utilizando bloques predefinidos y personalizados.

II-C. Integración de fuentes reales

Se reemplazaron generadores aleatorios por datos del mundo real, como archivos de imagen y audio. Este proceso incluyó la extracción de bits y el ajuste de bloques para adaptar las señales, comparando sus características espectrales con las de fuentes sintéticas.

II-D. Adaptación para esquemas de comunicación

Mediante modificaciones en pulsos y filtros, se emularon codificaciones de línea (ej. RZ, Manchester) y técnicas de modulación (ej. OOK, BPSK). Estos ajustes se orientaron a estudiar cómo la forma de la señal influye en su huella espectral y eficiencia energética.

II-E. Evaluación teórico-práctica

Se resolvieron interrogantes sobre el rol de bloques específicos, la interpretación de gráficas PSD y las diferencias entre señales unipolares y bipolares. Esto incluyó análisis comparativos y validación de resultados mediante herramientas integradas en GNU Radio.

III. ANÁLISIS DE RESULTADOS

Como parte fundamental del desarrollo del laboratorio, se solicitó el análisis detallado de bloques específicos dentro de la cadena de procesamiento de señales, con el objetivo de comprender su funcionamiento interno y evaluar su influencia en las etapas subsiguientes. Este estudio incluyó la identificación de parámetros críticos (como tasas de interpolación, respuestas al impulso y escalados de amplitud), así como la correlación entre su configuración y los efectos observables en los dominios temporal y espectral, permitiendo vincular teoría, implementación práctica y resultados experimentales.

III-A. Análisis del Ruido Blanco en Tiempo y Frecuencia

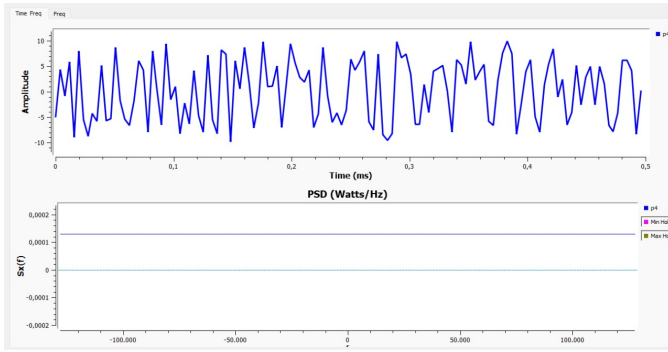


Figura 1. Ruido Blanco en Tiempo y Frecuencia.

El ruido blanco en el dominio temporal se manifiesta como una secuencia aleatoria sin patrones repetitivos, con amplitudes que varían siguiendo una distribución estadística uniforme o gaussiana. Esta aleatoriedad garantiza que no exista correlación entre muestras consecutivas, lo que lo convierte en un modelo ideal para emular interferencias en sistemas de comunicación. En el dominio frecuencial, su densidad espectral de potencia (PSD) teóricamente es plana, lo que indica una distribución uniforme de energía en todas las frecuencias. Sin embargo, en implementaciones prácticas como GNU Radio, la PSD está limitada por la frecuencia de muestreo $\pm \frac{f_s}{2}$, y su amplitud depende de parámetros como la desviación estándar de la fuente gaussiana utilizada. Esta característica lo hace útil para calibrar sistemas o simular canales con ruido aditivo.

III-B. Análisis de Señales Provenientes de una Cámara Fotográfica

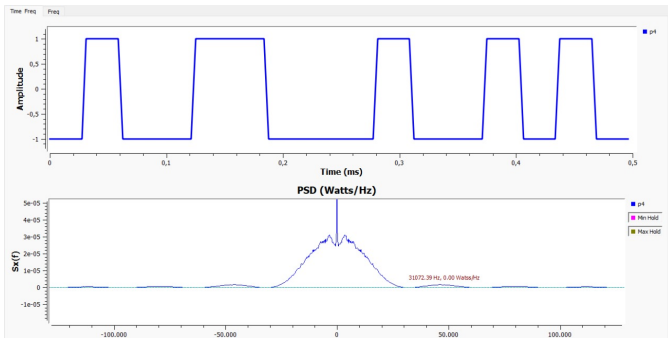


Figura 2. Imagen de Rana en Tiempo y Frecuencia.

Al procesar una imagen (por ejemplo, rana.jpg), la señal en el dominio temporal refleja la secuencia de bits generada a partir de los valores de los píxeles. Si la imagen contiene áreas homogéneas (como un fondo verde predominante), se producen tramos extensos de bits idénticos, reduciendo la variabilidad temporal. Esto se traduce en una PSD con un impulso prominente en frecuencia cero (DC), originado por

el valor promedio no nulo de la señal, y una energía mínima en altas frecuencias debido a la falta de transiciones abruptas (bordes). Al aumentar el parámetro Samples per Symbol (Sps), la amplitud de la PSD disminuye, ya que la energía total se distribuye en un mayor número de muestras, diluyendo la potencia por unidad de frecuencia. Este comportamiento subraya la relación entre la complejidad visual de una imagen y su huella espectral.

III-C. Análisis de Señales Provenientes de un Micrófono

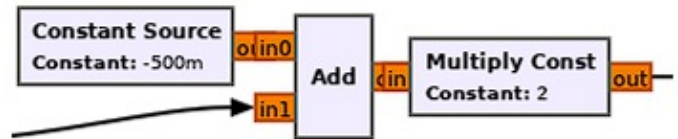


Figura 3. Audio en Tiempo y Frecuencia.

La señal de audio (por ejemplo, sonido.wav) preserva su forma original en el dominio temporal, mostrando oscilaciones similares a una función sinc en modulaciones digitales. Sin embargo, la presencia de ruido ambiental o interferencias introduce componentes espectrales no deseadas, como pulsos en frecuencias superiores a 20 kHz, inaudibles para humanos pero detectables en la PSD. Un aspecto crítico surge cuando Sps excede la longitud del vector de interpolación: esto induce la inserción de ceros entre muestras, generando discontinuidades que se reflejan como artefactos de aliasing y lóbulos espurios en la PSD. Estos efectos degradan la calidad de la señal, evidenciando la importancia de ajustar parámetros como la longitud del filtro y la tasa de muestreo para preservar la integridad espectral en aplicaciones prácticas.

IV. PREGUNTAS Y RESPUESTAS

IV-A. ¿Qué papel juega la siguiente combinación de bloques de la Figura 4?

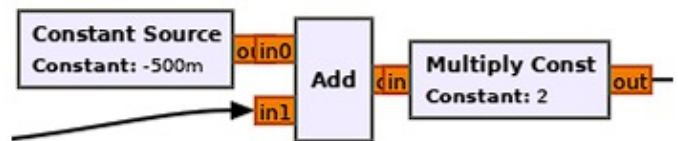


Figura 4. Combinación de Bloques GNURadio.

El arreglo de bloques "Add", "Constant Source", agregan un nivel DC a la señal, transformándola en bipolar (valores positivos y negativos). El bloque "Multiply Const" escala la amplitud para mantener los niveles originales (ej., de $\pm A/2$ a $\pm A$), evitando que la señal se atenúe tras el desplazamiento DC.

IV-B. ¿Qué papel juega el bloque “Interpolation FIR Filter”, ¿cómo funciona?

El bloque Interpolation FIR Filter interpola la señal insertando muestras adicionales según el vector h . Si se cambia el valor de Sps sin ajustar h , pueden surgir discontinuidades (muestras en cero) o solapamientos entre símbolos. Según el teorema de Nyquist, el ancho de banda mínimo requerido es $B = \frac{R_b Sps}{2}$. La frecuencia de muestreo en p3 se relaciona con p4 como $f_{sp3} = \frac{f_{sp4}}{Sps}$, ya que Sps define la interpolación.

IV-C. ¿Por qué razón la PSD de las señales binarias que provienen de una señal de audio es diferente a la que proviene de una foto siendo ellas igualmente señales binarias bipolares de forma rectangular?

Las señales de audio suelen tener PSD con energía concentrada en bandas específicas (ej., voces en 300 Hz - 3.4 kHz). En imágenes, la energía se distribuye espacialmente, generando PSD más planas debido a la aleatoriedad de los píxeles. Esto explica diferencias espectrales incluso si ambas son binarias.

IV-D. ¿Qué papel juega el bloque “Throttle”?

El bloque Throttle regula la tasa de muestreo para sincronizar el flujo de datos con la capacidad de procesamiento del sistema. Esto evita sobrecargas en tiempo real, asegurando estabilidad en simulaciones.

IV-E. ¿Qué pasaría con la PSD si no se hace la conversión a señal bipolar, sino que la señal binaria en p4 solo tiene valores de 0 ó 1 en lugar de -1 ó 1?

Una señal unipolar (0 y 1) introduce un componente DC en la PSD (pico en $f = 0$), ya que el valor promedio no es cero. En contraste, una señal bipolar (-1 y +1) elimina este pico, distribuyendo la energía simétricamente. El resto del espectro mantiene una forma similar, pero sin distorsión por DC.

IV-F. Se supone que el ruido blanco tiene un ancho de banda infinito, ¿coincide esto con lo observado en GNU Radio?, ¿por qué?

No coincide completamente. El ruido blanco teórico tiene PSD plana y ancho de banda infinito, pero en GNU Radio, el ancho de banda está limitado por la frecuencia de muestreo. Por ello, la PSD observada es plana solo dentro del rango $\pm \frac{f_s}{2}$.

IV-G. Se supone que una señal binaria aleatoria de forma rectangular tiene un ancho de banda infinito, ¿coincide esto con lo observado en GNU Radio y por qué?

Al aumentar Sps, los lóbulos de la PSD (asociados a la función sinc) se acercan entre sí, ya que el ancho de banda se reduce. Esto refleja la relación inversa entre duración del símbolo y dispersión espectral.

IV-H. ¿Qué fórmula podría ayudar a calcular el número de lóbulos de la PSD de señal binaria aleatoria de forma rectangular cuando se conoce la frecuencia de muestreo y Sps? Nota: el lóbulo de la mitad se cuenta como dos porque tiene el doble de ancho que los demás.

El espectro de una señal binaria tiene lóbulos principales y secundarios. Para hacer una estimación del número de lóbulos que tiene una señal, es necesario saber cuántas veces cabe la frecuencia de la señal en el ancho de banda $N_{Lobulos} = \frac{B}{f_{seal}} = \frac{Sps}{2}$ y teniendo en cuenta que el lóbulo principal ocupa el espacio equivalente a dos lóbulos normales en el espectro de frecuencias, se multiplica el valor obtenido por un factor de 2, por ende el número de lóbulos de la señal va a estar dado por $N_{Lobulos} = Sps$.

IV-I. ¿Cómo se calcula todo el rango de frecuencias que ocupa el espectro cuando se conoce Rb y Sps?

El ancho de banda aproximado es $B = \frac{R_b}{Sps}$ es la tasa de bits. Al aumentar Sps, el tiempo por símbolo crece, reduciendo el ancho de banda necesario.

IV-J. ¿Cómo se calcula la resolución espectral del analizador de espectros, cuando se conoce N y la frecuencia de muestreo?

La resolución espectral se calcula como $\Delta f = \frac{f_s}{N}$, donde N es el número de puntos de la FFT. Mayor N mejora la resolución, permitiendo distinguir componentes cercanas en frecuencia.

IV-K. ¿Qué pasaría si en el bloque “Unpack K Bits” se configura el parámetro K como 16?

Al aumentar K en Unpack K Bits, se descomponen más bits por muestra, generando secuencias más largas de ceros o unos. Esto induce periodicidades en la señal, creando armónicos espurios en la PSD y aumentando la potencia en DC.

IV-L. ¿Cómo calcularía la frecuencia de muestreo a la entrada del bloque “Unpack K Bits” si conoce el número de lóbulos de la PSD y el ancho de banda de la señal?

Para calcular la frecuencia de muestreo en la entrada del bloque Unpack K Bits se debe relacionar el número de lóbulos de la PSD y el ancho de banda de la señal. Asumiendo que cada lóbulo de la PSD se asocia a la tasa de símbolos (según la relación de Nyquist), se puede estimar la frecuencia de muestreo mediante la fórmula $f_s = N_{lob} * R_{symbols}$ garantizando que la frecuencia de muestreo sea mayor a dos veces el ancho de banda.

IV-M. ¿Cómo calcularía la frecuencia de muestreo a la salida del bloque “Unpack K Bits” si conoce la frecuencia de muestreo a la entrada?

El bloque Unpack K Bits convierte cada byte en K bits individuales. La frecuencia de muestreo a la salida es $f_{salida} = K * f_{entrada}$, ya que se generan K muestras por byte.

IV-N. ¿Cómo calcularía la frecuencia de muestreo a la salida del bloque “Char to Float” si conoce la frecuencia de muestreo a la entrada?

El bloque “Char to Float” no altera la frecuencia de muestreo. Solo cambia el tipo de dato (de entero a flotante), manteniendo f_s igual en entrada y salida.

IV-Ñ. ¿Para qué caso de Sps la PSD de una señal binaria aleatoria bipolar es similar a la PSD de ruido blanco?

Con $Sps = 1$, la PSD de una señal binaria bipolar se asemeja al ruido blanco, ya que los símbolos son independientes y equiprobables, generando un espectro plano (sin lóbulos dominantes).

IV-O. ¿Qué cambios mínimos haría al flujograma, manipulando principalmente h, si desea que los bits en la señal binaria aleatoria tomen la forma de dientes de sierra?

Para generar pulsos en dientes de sierra, el vector h debe ser de longitud 2 con valores [1, -1], creando una transición lineal entre símbolos.

IV-P. ¿Qué cambios mínimos haría al flujograma, manipulando principalmente h, si desea que la señal binaria aleatoria tenga codificación de línea Unipolar RZ.

Para Unipolar RZ, h debe ser [1, 1, 0, 0] (mitad activo, mitad cero), generando pulsos que regresan a cero en cada símbolo.

IV-Q. ¿Qué cambios mínimos haría al flujograma, manipulando principalmente h, si desea que la señal binaria aleatoria tenga codificación de línea Manchester NRZ?

Para Manchester NRZ, h debe ser de mitad del valor 1 y la otra mitad de -1, alternando la polaridad en mitad del símbolo para sincronización.

IV-R. ¿Qué cambios mínimos haría en el flujograma, aprovechando h y el FIR Interpolating Filter para que la señal binaria tenga la forma de señal OOK?

Para OOK, el vector h contiene una senoide $\sin(15n)$, y la señal se multiplica por esta portadora. El bloque Constant Source se configura en 0 para eliminar el offset.

IV-S. ¿Qué cambios mínimos haría en el flujograma, aprovechando h y el FIR Interpolating Filter para que la señal binaria tenga la forma de señal BPSK?

Para BPSK, se usa el mismo h sinusoidal que en OOK, pero la señal se desplaza a bipolar ($\pm A$) antes de la multiplicación, generando cambios de fase de 180° .

IV-T. ¿Qué cambios mínimos haría en el flujograma, aprovechando h y el FIR Interpolating Filter para que la señal binaria tenga la forma de señal ASK?

Para ASK, el vector h es sinusoidal, y la señal se escala a un cuarto de su amplitud original, creando múltiples niveles de amplitud modulada.

IV-U. ¿Qué cambios mínimos haría en el flujograma, aprovechando h y el FIR Interpolating Filter para que la señal binaria tenga la forma de los latidos del corazón como se muestra en la Figura 5?

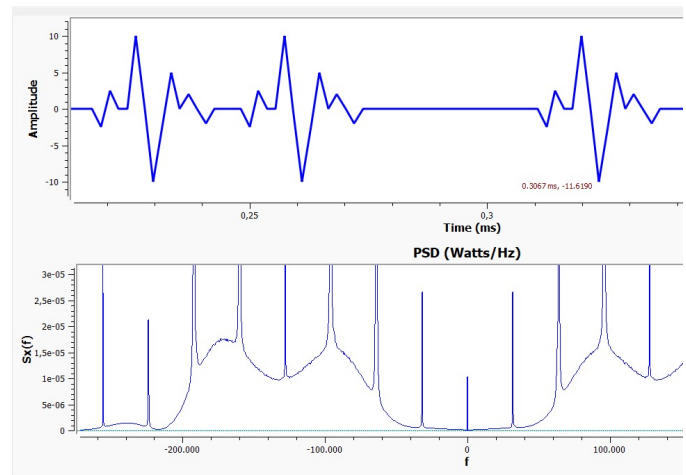


Figura 5. Señal Obtenida de ECG.

Para pulsos cardíacos, h se define con valores no lineales [0, 0, -0,25, 0, 0,25, 0, 0, 1, 0, -1, -0,25, 0,5, 0, 0,25, 0, -0,25, 0, 0, 0], emulando la forma característica de un latido.

IV-V. ¿Qué cambios mínimos haría en el flujograma, aprovechando h y el FIR Interpolating Filter para que la señal binaria tenga la forma que se muestra en la Figura 6?

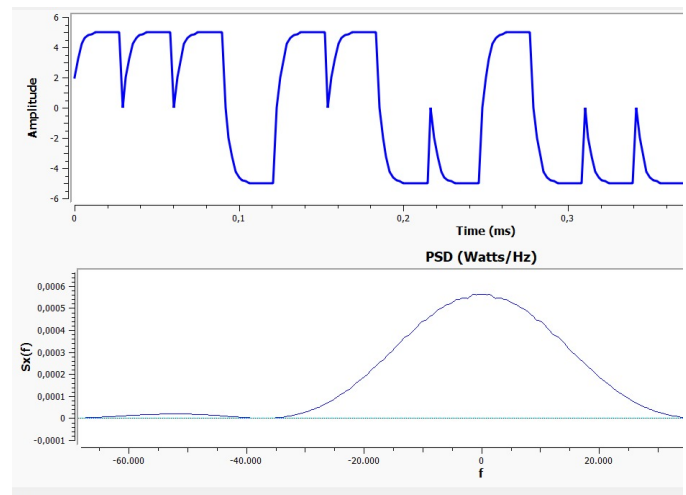


Figura 6. Señal Obtenida Pulsos Rizados.

Para pulsos rizados, h se genera con una función logarítmica $\ln(100x)$, variando x para crear una envolvente suave.

IV-W. Explique usando gráficas de PSD la diferencia que existe entre la PSD de una señal binaria bipolar y una unipolar.

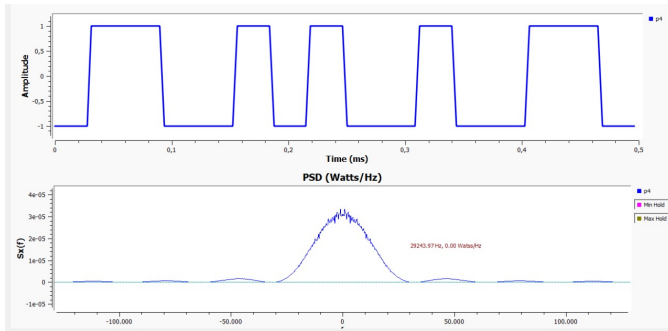


Figura 7. PSD Señal Binaria Unipolar.

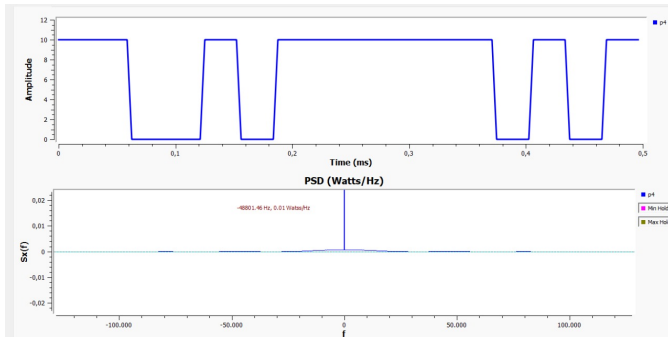


Figura 8. PSD Señal Binaria Bipolar.

La PSD unipolar tiene un pico en DC y lóbulos sinc asimétricos. La bipolar elimina el pico en DC, mostrando lóbulos simétricos, lo que mejora la eficiencia espectral en comunicaciones.

V. CONCLUSIONES

- El ruido blanco en GNU Radio, aunque teóricamente de ancho de banda infinito, presenta una PSD limitada por la frecuencia de muestreo, lo que lo hace idóneo para simular interferencias en entornos controlados. Esta característica es esencial en aplicaciones como calibración de receptores y evaluación de la tolerancia a ruido en sistemas de comunicación inalámbricos.
- Las señales derivadas de imágenes (ej., rana.jpg) muestran un pico dominante en DC debido a la homogeneidad de los píxeles, lo que evidencia la relación entre complejidad visual y distribución espectral. Este hallazgo es relevante en procesamiento de imágenes médicas o satelitales, donde la supresión de componentes DC optimiza el análisis de bordes y texturas.

- Las señales de audio (sonido.wav) revelan que una interpolación mal configurada (Sps elevado) introduce artefactos de aliasing, degradando la calidad espectral. Esto subraya la importancia de ajustar parámetros en sistemas de voz sobre IP (VoIP) o radiodifusión digital para preservar la fidelidad acústica.
- La emulación de codificaciones (RZ, Manchester) y modulaciones (OOK, BPSK) demostró que la forma del pulso incide directamente en la eficiencia espectral. Por ejemplo, la codificación Manchester es útil en redes de alta velocidad donde la sincronización temporal es crítica, mientras que BPSK se emplea en enlaces satelitales por su robustez ante ruido.
- La variación del parámetro Sps permitió corroborar que a mayor Sps, menor ancho de banda ocupado, facilitando la adaptación de sistemas a canales con restricciones espectrales. Esto es aplicable en tecnologías como LoRa o 5G, donde la escalabilidad del ancho de banda es clave para gestionar múltiples dispositivos.

REFERENCIAS

- [1] Homero Ortega Boda, Oscar Mauricio Reyes Torres. *Comunicaciones Digitales basadas en radio definida por software*. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones.
- [2] GNU Radio. *Python Module - GNU Radio*. Disponible en: https://wiki.gnuradio.org/index.php?title=Python_Module
- [3] Repositorio en GitHub. Disponible en: https://github.com/Jomao03/com2_B1_G3