

# RADIO DEFINIDA POR SOFTWARE (GNURADIO)

David Josué Díaz Ortiz, 2204269, Estudiante Ing. Electrónica,  
Duban Yesid Cortes Tabares, 2214644, Estudiante Ing. Electrónica

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones  
Universidad Industrial de Santander

Agosto 30, 2025

[https://github.com/David2204269/CommII\\_LabB1\\_G4.git](https://github.com/David2204269/CommII_LabB1_G4.git)

## Abstract

This report presents the development of a Software Defined Radio (SDR) system using GNU Radio in the context of Digital Communications. The practice involved implementing and evaluating digital signal processing blocks—average time, accumulator, and differentiator—to improve signal-to-noise ratio and reduce noise effects. GitHub was integrated for collaborative version control, ensuring organized team progress. The results confirmed the effectiveness of the techniques and emphasized the flexibility of SDR in communication systems, strengthening skills in digital signal processing and collaborative project management.

**Keywords:** Software-defined radio (SDR), GNU radio, digital communications, signal-to-noise ratio (SNR), digital signal processing (DSP), GitHub.

## 1 Introducción

El desarrollo de las tecnologías de radio definida por software (SDR) ha impulsado la implementación flexible de sistemas de comunicación mediante herramientas como GNU Radio. En este laboratorio se trabajó en la programación y configuración de bloques de procesamiento, incluyendo el acumulador, el diferenciador y tiempo promedio, orientados a mejorar el tratamiento de señales afectadas por ruido. Además, se reforzaron competencias en el uso de comandos básicos de Git y Linux, así como en la gestión colaborativa de código a través de GitHub. Este informe presenta el procedimiento realizado, los resultados obtenidos y la importancia de estas prácticas en el ámbito del procesamiento digital de señales.

## 2 Metodología

Se inició con la familiarización de GNU Radio, abordando sus configuraciones básicas y la gestión de repositorios en GitHub. Posteriormente, se implementaron en Python bloques de procesamiento como el acumulador y el diferenciador, junto con un bloque estadístico para analizar el efecto del ruido y aplicar técnicas de mitigación. Finalmente, los procedimientos y resultados se documentaron en un informe estructurado, resaltando la aplicación práctica de GNU Radio y GitHub en el procesamiento digital de señales, como se muestra a continuación:

### 1. Creación del repositorio en GitHub y generación de tokens personales:

Se creó el repositorio “CommII\_LabB1\_G4” en GitHub para la gestión del código y la colaboración entre los integrantes del grupo (Figura 1). Cada estudiante generó un token personal de acceso, utilizado para la autenticación segura desde la terminal y en reemplazo de contraseñas. El repositorio permite el control de versiones, la creación de ramas y la integración de cambios, garantizando la seguridad y eficiencia en operaciones como git push, git pull y git commit.

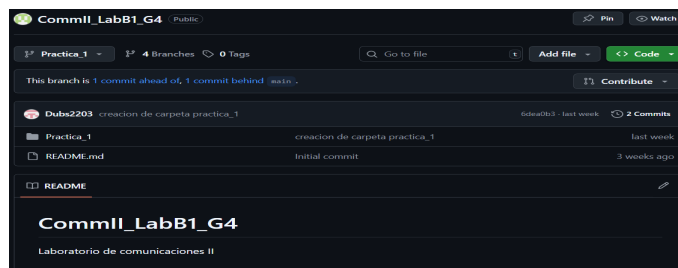


Fig. 1: Creación de repositorio.



## 2. Navegación y gestión del repositorio en GitHub:

Se exploró la integración de GitHub con la terminal de Linux, comprendiendo su aplicación en el control de versiones y la colaboración en proyectos. Tras clonar el repositorio en local, se creó el archivo "Readme.md" y se verificó su almacenamiento en la nube. Para la gestión de cambios se emplearon comandos fundamentales: "git pull" para actualizar el repositorio, "git add" y "git commit -m" para confirmar modificaciones, y "git push" para sincronizarlas en GitHub, asegurando así un flujo de trabajo ordenado y seguro.

## 3. Creación de ramas en GitHub:

Se creó la rama "Practica\_1" (Figura 2). como repositorio principal de evidencias y avances del grupo. A partir de esta, cada estudiante generó una rama personal identificada con su usuario, lo que permitió desarrollar, probar y modificar el código de forma independiente. Posteriormente, los cambios fueron integrados en la rama principal, garantizando un control de versiones eficiente y una adecuada gestión colaborativa.

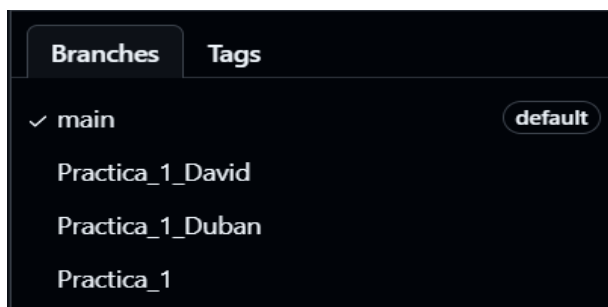


Fig. 2: Ramas de participantes.

## 4. Exploración y programación en GNU Radio:

En la práctica se emplearon bloques predefinidos y se desarrollaron otros personalizados, como un promedio temporal, un acumulador y un diferenciador, con el fin de analizar las señales con mayor precisión. Si bien es posible trabajar únicamente con bloques estándar, la programación ofrece la ventaja de adaptar y optimizar las herramientas, ampliando las capacidades de procesamiento y favoreciendo su reutilización en tiempo real.

Para alcanzar dicho objetivo, se implementaron y utilizaron bloques específicos dentro de la plataforma GNU Radio.

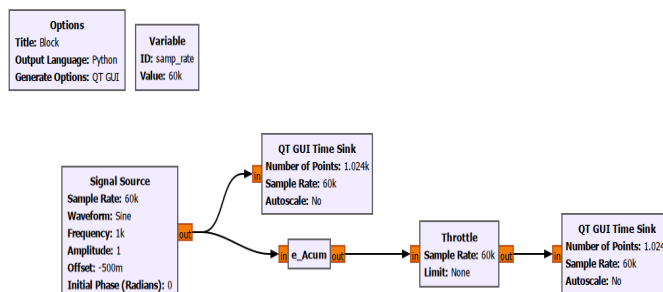


Fig. 3: Bloque acumulador.

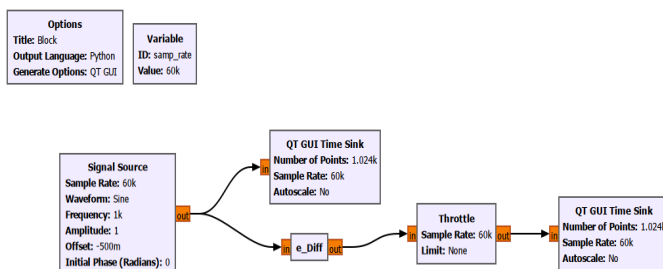


Fig. 4: Bloque diferenciador.

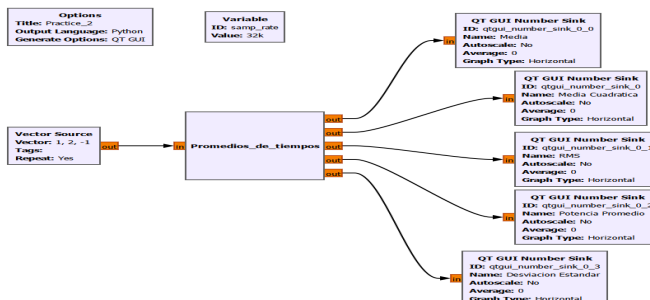


Fig. 5: Diagrama de bloques de tiempo promedio.

Los códigos empleados fueron tomados del texto guía de comunicaciones, el cual proporcionaba las bases para los bloques en Python, como el acumulador, el diferenciador y el promedio temporal. A dichos códigos se les realizaron varias correcciones de sintaxis y ajustes en el funcionamiento de algunas funciones para garantizar su correcto funcionamiento y así poder desarrollar satisfactoriamente la práctica. En el siguiente enlace se presentan los códigos corregidos: [Códigos](#).

### 3 Análisis de resultados

De este laboratorio obtuvimos estos laboratorios.

#### • BLOQUE ACUMULADOR

El bloque acumulador se encarga de ir sumando de manera progresiva los valores de una señal de entrada, generando como salida la acumulación de esas muestras a lo largo del tiempo. Este comportamiento resulta muy útil en distintas aplicaciones, como el análisis de la energía de las señales, la integración de datos en tiempo real o la detección de tendencias en mediciones. En el ámbito de la radio y la radio definida por software (SDR), puede aplicarse para calcular la potencia acumulada de una señal, mejorar la detección de pulsos débiles o realizar ajustes dentro de algoritmos de sincronización y corrección de errores.

Una vez diseñado el bloque, se procedió a observar sus resultados en GNU Radio.

Donde ya hecho los bloques procedemos a mostrar a los resultados de GNURadio.

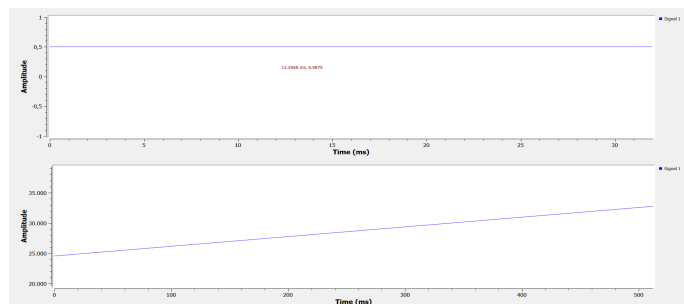


Fig. 6: Gráfica bloque acumulador.

#### • BLOQUE DIFERENCIADOR

El bloque diferenciador, por su parte, se fundamenta en la teoría de señales: al derivar una señal triangular, se obtiene como resultado una señal rectangular. Al visualizar los resultados en GNU Radio, se confirmó que el bloque funciona correctamente. Se notó además que la salida presenta una mayor amplitud, lo cual se debe a que la diferenciación resalta los cambios bruscos de la señal. En efecto, la amplitud de la salida es proporcional a la pendiente de la entrada, lo que explica el incremento más notable cuando el período es más pequeño.

De la misma forma que con el acumulador, una vez implementado el bloque se visualizaron los resultados en GNU Radio.

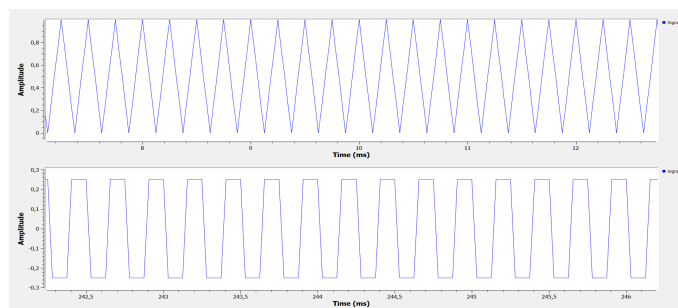


Fig. 7: Gráfica bloque diferenciador.

#### • BLOQUE DE TIEMPO PROMEDIO

Durante la clase se analizó el bloque diseñado en GNU Radio, el cual tiene como función calcular el promedio de los tiempos de una señal. En la imagen se observa que el bloque recibe una señal de entrada, la procesa internamente y entrega varias salidas: una correspondiente al valor promedio calculado y otra relacionada con el número de muestras empleadas en dicho cálculo. Este procedimiento consiste en tomar varias muestras consecutivas y promediarlas, lo que reduce las fluctuaciones instantáneas y atenúa el ruido aleatorio presente en la señal. En aplicaciones de radio digital, este tipo de bloque es fundamental para mejorar la calidad de la recepción: el promedio temporal disminuye la variabilidad introducida por el ruido, permitiendo obtener una señal más estable y con mejor relación señal-ruido (SNR).

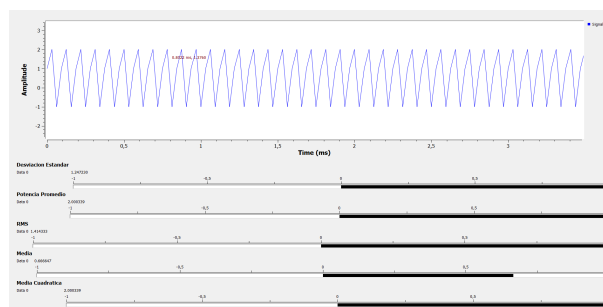


Fig. 8: Resultados de tiempo promedio.

## 4 Aplicación sugerida

### Bloque acumulador:

- Aplicación: El bloque acumulador suma progresivamente las muestras de la señal, lo que permite analizar la tendencia acumulativa. Esto es útil, por ejemplo, en sistemas de monitoreo de espectro o en la detección de pulsos débiles, debido a la integración de muestras a lo largo del tiempo que ayuda a realzar señales de baja potencia y a compensar el ruido.
- Vista de los resultados: Si se tiene una señal de entrada (con variaciones o ruido) se transforma en una curva creciente que representa la suma acumulada de sus valores.

### Bloque diferenciador:

- Aplicación; Este bloque es ideal para resaltar cambios abruptos en la señal. Por ejemplo, en modulaciones digitales que puede utilizarse para detectar transiciones rápidas o bordes de pulsos, lo que facilita la sincronización y la identificación de símbolos en la transmisión. También se puede aplicar en sistemas de radar para detectar la aparición de picos o variaciones repentinas.
- Vista de los resultados: Si se tiene una gráfica donde la señal de entrada es una onda triangular. Tras pasar por el bloque diferenciador, la salida se convierte en una señal rectangular o “de caja”, donde los saltos en la señal original se destacan notablemente.

### Bloque de tiempo promedio:

- Aplicación: Este bloque promedia múltiples muestras de la señal a lo largo del tiempo para reducir las fluctuaciones y atenuar el ruido aleatorio. Es fundamental en la mejora de la relación señal-ruido (SNR) en sistemas de SDR, debido a que permite obtener una versión “suavizada” de la señal original, facilitando su demodulación y análisis.
- Vista de los resultados: A una señal de entrada con ruido "Con fluctuaciones de picos", da cómo resultado tras el promediado una curva más suave y estable, con la reducción de los picos de ruido.

## 5 Conclusiones

- El bloque acumulador permitió observar cómo la integración de las muestras a lo largo del tiempo ayuda a evaluar la energía de la señal y a detectar tendencias en su comportamiento, algo fundamental cuando se trabaja con procesamiento en tiempo real.
- El bloque de promedio demostró que al combinar varias muestras se logra suavizar la señal, reduciendo las fluctuaciones generadas por el ruido aleatorio. Esta función resulta especialmente valiosa en entornos de radio definida por software (SDR), donde las interferencias pueden deteriorar la señal y complicar su análisis.
- La implementación de los bloques digitales (promedio de tiempo, acumulador y diferenciador) en GNU Radio resultó ser una herramienta eficaz para reducir el impacto del ruido, lo que se traduce en una mejor relación señal-ruido (SNR) y, por lo tanto, en una transmisión de mayor calidad.
- se evidenció que la forma en que se muestrea la señal de entrada es clave para evitar problemas de sincronización al trabajar con el acumulador y el diferenciador. Si las frecuencias se desfasan, el sistema puede tomar muestras en puntos incorrectos, lo que provoca errores y una representación deficiente de la señal.

## References

- [1] G. R. Wiki, “Creating your first block, online, 4 may 2024. [online].” [Online]. Available: [https://wiki.gnuradio.org/index.php?title=Creating\\_Your\\_First\\_Block](https://wiki.gnuradio.org/index.php?title=Creating_Your_First_Block)
- [2] H. O. y Oscar M. Reyes Torres, “Comunicaciones digitales basadas en radio definida por software. 2022. [online].” [Online]. Available: [https://drive.google.com/file/d/1fd9M4\\_bJwLOajQdkdN9ex2pLYRoXomz/view](https://drive.google.com/file/d/1fd9M4_bJwLOajQdkdN9ex2pLYRoXomz/view)