



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE TELECOMUNICACIONES**

**Tema:**

---

**GUÍAS DE PRÁCTICAS DE LABORATORIO INTERDISCIPLINARIAS  
PARA SISTEMAS DE COMUNICACIONES UTILIZANDO RADIO  
DEFINIDO POR SOFTWARE.**

---

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la  
obtención del título de Ingeniero en Telecomunicaciones

**ÁREA:** Comunicaciones

**LÍNEA DE INVESTIGACIÓN:** Tecnologías de la información y sistemas de control

**AUTOR:** Jonathan David Palán Murillo

**TUTOR:** Ing. Mario Geovanny García Carrillo, Mg

**Ambato - Ecuador**

**febrero – 2024**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del trabajo de titulación con el tema: **GUÍAS DE PRÁCTICAS DE LABORATORIO INTERDISCIPLINARIAS PARA SISTEMAS DE COMUNICACIONES UTILIZANDO RADIO DEFINIDO POR SOFTWARE**, desarrollado bajo la modalidad Proyecto de Investigación por el señor Jonathan David Palan Murillo, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024.

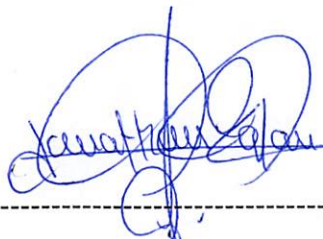
-----  
Ing. Mario Geovanny García Carillo, Mg.

**TUTOR**

## AUTORÍA

El presente trabajo de titulación con el tema: GUÍAS DE PRÁCTICAS DE LABORATORIO INTERDISCIPLINARIAS PARA SISTEMAS DE COMUNICACIONES UTILIZANDO RADIO DEFINIDO POR SOFTWARE es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024.



---

Jonathan David Palan Murillo

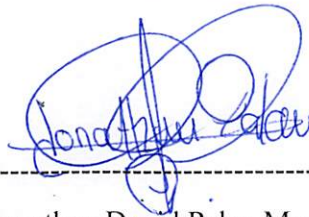
C.C. 1804204988

AUTOR

## DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2024.



---

Jonathan David Palan Murillo

C.C. 1804204988

AUTOR

## **APROBACIÓN DEL TRIBUNAL DE GRADO**

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Jonathan David Palan Murillo, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado: GUÍAS DE PRÁCTICAS DE LABORATORIO INTERDISCIPLINARIAS PARA SISTEMAS DE COMUNICACIONES UTILIZANDO RADIO DEFINIDO POR SOFTWARE, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

-----  
Ing. Elsa Pilar Urrutia Urrutia, Mg.  
PRESIDENTE DEL TRIBUNAL

-----  
Ing. Santiago Manzano Villafuerte, Mg  
PROFESOR CALIFICADOR

-----  
Ing. Juan Pallo Noroña, Mg  
PROFESOR CALIFICADOR

## DEDICATORIA

*Dedico este trabajo especialmente a mis padres Xavier y Mónica por el apoyo y enseñanzas que me han dado para poder cumplir mi meta. A mi hermano Jeremy por su compañía incondicional. A mi familia por los deseos de superación y éxitos y a todas las personas y amigos que están y que conocí en esta etapa de mi vida.*

***Palan Jonathan***

## AGRADECIMIENTO

*Agradezco a Dios por darme la sabiduría para cumplir mis objetivos.*

*A mis padres por enseñarme el significado de esfuerzo y superación.*

*A la Universidad por brindarme un lugar y la oportunidad de formarme profesionalmente.*

*A mi tutor el ingeniero Mario García por el tiempo y consejos para cumplir este proyecto.*

*Y a los ingenieros Santiago Manzano, Juan Pablo Pallo y Julio Cuji por la colaboración en este trabajo.*

## ÍNDICE GENERAL DE CONTENIDOS

<b>PORTADA</b> .....	<b>i</b>
<b>APROBACIÓN DEL TUTOR</b> .....	<b>ii</b>
<b>AUTORÍA</b> .....	<b>iii</b>
<b>DERECHOS DE AUTOR</b> .....	<b>iv</b>
<b>APROBACIÓN DEL TRIBUNAL DE GRADO</b> .....	<b>v</b>
<b>DEDICATORIA</b> .....	<b>vi</b>
<b>AGRADECIMIENTO</b> .....	<b>vii</b>
<b>ÍNDICE GENERAL DE CONTENIDOS</b> .....	<b>viii</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>xi</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>xii</b>
<b>ÍNDICE DE ANEXOS</b> .....	<b>xvi</b>
<b>RESUMEN EJECUTIVO</b> .....	<b>xvii</b>
<b>ABSTRACT</b> .....	<b>xviii</b>
<b>CAPÍTULO I. MARCO TEÓRICO</b> .....	<b>1</b>
1.1 Tema de investigación .....	1
1.1.1 Planteamiento del problema .....	1
1.2 Antecedentes investigativos .....	2
1.3 Fundamentación teórica .....	4



1.3.1 Radio Definida por Software .....	4
1.3.2 Arquitectura de SDR .....	5
1.3.3 Hardware SDR .....	6
1.3.4 Equipos USRP .....	7
1.3.5 USRP N210 .....	13
1.3.6 Arquitectura del equipo USRP N210 .....	15
1.3.7 Software y controladores.....	17
1.3.8 Sistema de comunicaciones.....	22
1.4 Objetivos .....	23
1.4.1 Objetivo general.....	23
1.4.2 Objetivos específicos .....	23
<b>CAPÍTULO II.- METODOLOGÍA.....</b>	<b>25</b>
2.1 Materiales.....	25
2.2 Métodos.....	26
2.2.1 Modalidad de la investigación .....	26
2.2.2 Recolección de información.....	26
2.2.3 Procesamiento y análisis de datos .....	26
<b>CAPÍTULO III.- RESULTADOS Y DISCUSIÓN .....</b>	<b>28</b>
3.1 Desarrollo de la propuesta.....	28
3.2 Desarrollo de prácticas .....	29

3.2.1 Práctica 1: Introducción a GNU Radio y USRP .....	30
3.2.2 Práctica 2: Análisis de señales .....	35
3.2.3 Práctica 3: Operaciones con señales continuas .....	39
3.2.4 Práctica 4: Operaciones con señales singulares .....	44
3.2.5 Práctica 5: Osciladores de radiofrecuencia .....	47
3.2.6 Práctica 6: Filtros .....	51
3.2.7 Práctica 7: Modulador AM.....	56
3.2.8 Práctica 8: Demodulador AM .....	61
3.2.9 Práctica 9: Modulador FM .....	64
3.2.10 Práctica 10: Demodulador FM.....	69
3.2.11 Práctica 11: Receptor de Frecuencias FM.....	73
3.2.12 Práctica 12: Transmisión y recepción de señales FM .....	76
3.2.13 Práctica 13: Red GSM con USRP.....	81
<b>CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>87</b>
4.1 Conclusiones .....	87
4.2 Recomendaciones.....	88
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>90</b>
<b>ANEXOS .....</b>	<b>94</b>

## ÍNDICE DE TABLAS

Tabla 1. Dispositivos SDR.....	6
Tabla 2. Parámetros técnicos USRP B200/B210 .....	8
Tabla 3. Parámetros técnicos USRP N200/N210 .....	8
Tabla 4. Parámetros técnicos USRP X300/X310 .....	9
Tabla 5. Parámetros técnicos USRP E313 .....	10
Tabla 6. Parámetros técnicos USRP RIO .....	10
Tabla 7. Parámetros técnicos USRP E320 .....	11
Tabla 8. Parámetros técnicos USRP N310 .....	12
Tabla 9. Parámetros técnicos USRP N320 .....	12
Tabla 10. Características técnicas del equipo USRP N210 .....	13
Tabla 11. Descripción del panel frontal del dispositivo .....	14
Tabla 12. Descripción general del controlador NI-USRP y del paquete UHD de Ettus Research .....	17
Tabla 13. Distribución ventana general de GNU Radio .....	34
Tabla 14. Parámetros para generador de señales .....	36
Tabla 15. Parámetros para tomar de bloques Signal Source .....	40

## ÍNDICE DE FIGURAS

Figura 1. Diagrama de bloques funcionales de SDR .....	5
Figura 2. Estructura de SDR .....	5
Figura 3. USRP N210 Series.....	14
Figura 4. Arquitectura USRP N210 SERIES .....	15
Figura 5. Pilar de Software RFNoC .....	18
Figura 6. Conexión usuario-USRP a través de GNU Radio .....	19
Figura 7. Arquitectura de GNU Radio .....	19
Figura 8. USRP API de LabView .....	20
Figura 9. Comunicación entre USRP y NI LabVIEW .....	21
Figura 10. Diagrama de bloques de componentes SDR y cómo se conectan con MATLAB .....	21
Figura 11. Áreas académicas necesarias para la comprensión de los sistemas de comunicaciones .....	23
Figura 12. Diagrama del proceso del desarrollo de la propuesta .....	29
Figura 13. Diagrama de conexión e instalación necesarias.....	30
Figura 14. Ventana de inicio Virtual Box .....	31
Figura 15. Ubuntu 22.04 .....	32
Figura 16. Asignación de IP estática.....	33
Figura 17. Ventana de inicio de GNU Radio .....	35
Figura 18. Diagrama de práctica 2 .....	36

Figura 19. Señal senoidal generada.....	37
Figura 20. Señal Cuadrada generada.....	38
Figura 21. Diagrama de variables práctica 3.....	39
Figura 22. Diagrama de práctica 3 .....	40
Figura 23. Interfaz y operación entre dos señales .....	42
Figura 24. Respuesta en frecuencia de operación entre señales.....	42
Figura 25. Operación entre bloques Signal Source y Vector Source .....	43
Figura 26. Respuesta en Frecuencia de operación .....	43
Figura 27. Diagrama de práctica 4 .....	45
Figura 28. Interfaz y operación entre bloques personalizados .....	46
Figura 29. Diagrama de práctica 5 .....	48
Figura 30. Interfaz Señal de oscilador AM .....	50
Figura 31. Grafica en tiempo y frecuencia del oscilador Sinusoidal .....	50
Figura 32. Diagrama de práctica 6 .....	52
Figura 33. Gráficas de señal de audio ante filtro pasa bajos .....	53
Figura 34. Gráficas de señales de audio ante filtro pasa altos .....	54
Figura 35. Gráficas de señal de audio ante filtro pasa banda.....	54
Figura 36. Gráficas de señal de audio ante filtro de banda relejada .....	55
Figura 37. Diagrama de práctica 7 .....	56
Figura 38. Sistema de modulación AM con archivo de audio .....	57
Figura 39. Modulación AM con $m=50\%$ .....	59

Figura 40. Modulación AM con $m=100\%$ .....	59
Figura 41. Modulación AM con $m=150\%$ .....	60
Figura 42. Modulación AM de un archivo de audio .....	60
Figura 43. Bloques para demodular una señal AM.....	62
Figura 44. Sistema Modulador y Demodulador AM .....	62
Figura 45. Modulación y demodulación de señal AM.....	63
Figura 46. Modulación y demodulación AM de archivo de audio .....	64
Figura 47. Diagrama de práctica 9 .....	65
Figura 48. Habilitar bloques.....	66
Figura 49. Señal original.....	67
Figura 50. Frecuencia Original .....	67
Figura 51. Señal resultante a la salida del modulador.....	68
Figura 52. Frecuencia resultante a la salida del modulador .....	68
Figura 53. Diagrama de práctica 10 .....	69
Figura 54. Señal modulada recibida a través del archivo de datos .....	71
Figura 55. Frecuencia de la señal modulada .....	71
Figura 56. Salida del sistema de demodulación GFM Señal recuperada.....	72
Figura 57. Frecuencia de la señal recuperada .....	72
Figura 58. Diagrama de práctica 11 .....	74
Figura 59. Receptor FM .....	76
Figura 60. Diagrama de práctica 12 .....	77

Figura 61. Frecuencia de la señal enviada a través del trasmisor .....	79
Figura 62. Modulación FM de la señal .....	80
Figura 63. Diagrama de constelación de la señal .....	80
Figura 64. Señal FM captada en el recetor FM .....	81
Figura 65. Diagrama implementación red GSM .....	82
Figura 66. Envío y respuestas de mensajería entre suscriptores Poco-Redmi .....	85
Figura 67. Envío y respuestas de mensajería entre suscriptores Redmi-Poco .....	85
Figura 68. Llamada entrante entre suscriptores Poco-Redmi .....	86

## ÍNDICE DE ANEXOS

Anexo A. Datasheet USRP N210.....	94
Anexo B. Datasheet antenas VERT900 .....	95



## RESUMEN EJECUTIVO

El avance continuo de las tecnologías de comunicación ha presenciado desarrollos significativos en los últimos años debido al aumento de dispositivos inalámbricos y la creciente demanda de usuarios. En respuesta a estas tendencias, la búsqueda de soluciones innovadoras y sólidos conocimientos en este ámbito se ha vuelto imprescindible. Los equipos USRP han destacado en el campo del desarrollo e investigación en entornos académicos, gracias a su versatilidad y flexibilidad en la implementación de sistemas de comunicación mediante el uso de software.

El presente trabajo de investigación tiene como objetivo la implementación de un conjunto de prácticas de laboratorio empleando software libre y dispositivos USRP. Su propósito es complementar el estudio teórico de áreas académicas contempladas en la carrera de Telecomunicaciones. Se busca proporcionar a los estudiantes la oportunidad de adquirir conocimientos en un entorno práctico y experimental. La ejecución de estas prácticas facilita a los estudiantes la comprensión de la relación y uso conjunto de software y hardware, sirviendo como una introducción esencial al ámbito de dispositivos de radio definidos por software.

Este proyecto busca facilitar la comprensión del uso de esta tecnología a través de un documento que detalla la elaboración y procedimiento de las prácticas. Este documento servirá como guía para los instructores de laboratorios, permitiéndoles manejar los dispositivos de manera eficiente. Las áreas académicas abordadas en este trabajo, fundamentales para el manejo y configuración de los sistemas a implementar, incluyen módulos como Sistemas Lineales, Procesamiento de Señales, Comunicaciones Analógicas y Digitales, Sistemas Inalámbricos y Comunicaciones Avanzadas.

**Palabras clave:** USRP N210, SDR, UHD, GNU radio, radio definido por software.

## ABSTRACT

The continuous advancement of communication technologies has witnessed significant developments in recent years due to the increase in wireless devices and the growing demand from users. In response to these trends, the search for innovative solutions and solid knowledge in this field has become essential. In this context, USRP teams have stood out in the field of development and research in academic environments, thanks to their versatility and flexibility in implementing communication systems using software.

The aim of this research project is the implementation of a set of laboratory practices using free software and USRP devices. Its purpose is to complement the theoretical study of academic areas covered in the Telecommunications Engineering program. The goal is to provide students with the opportunity to acquire knowledge in a practical and experimental environment. The execution of these practices facilitates students' understanding of the relationship and joint use of software and hardware, serving as an essential introduction to the field of software-defined radio devices.

In this sense, the project seeks to facilitate the understanding of the use of this technology through a document detailing the development and procedure of the practices. This document will serve as a guide for laboratory instructors, allowing them to efficiently manage the devices. The academic areas addressed in this work, essential for the management and configuration of the systems to be implemented, include modules such as Linear Systems, Signal Processing, Analog and Digital Communications, Wireless Systems, and Advanced Communications.

**Keywords:** USRP N210, SDR, UHD, GNU radio, software defined radio.

# CAPÍTULO I. MARCO TEÓRICO

## 1.1 Tema de investigación

GUÍAS DE PRÁCTICAS DE LABORATORIO INTERDISCIPLINARIAS PARA SISTEMAS DE COMUNICACIONES UTILIZANDO RADIO DEFINIDA POR SOFTWARE

### 1.1.1 Planteamiento del problema

En la evolución de las telecomunicaciones inalámbricas han aparecido diversas tecnologías para el intercambio de información entre dos puntos distantes, con requisitos cada vez más exigentes. Las incompatibilidades entre las diferentes tecnologías han supuesto un problema a la hora de reutilizar equipos o prestar determinados servicios, como por ejemplo los terminales de telefonía móvil [1].

En el mundo de las comunicaciones, el concepto de Radio Definido por Software ha ganado un gran reconocimiento y aceptación. La falta de estándares y directrices a nivel mundial dificulta la implementación coherente y eficiente de los sistemas SDR, lo que limita su potencial de adaptación y evolución rápida en diferentes sectores industriales. El uso de técnicas digitales en lugar de hardware tradicional, como mezcladores, moduladores y demoduladores, permite a los SDR adaptarse y evolucionar de manera más rápida y eficiente. El SDR ha encontrado aplicaciones en diversas industrias, incluyendo las comunicaciones móviles, la radiodifusión, la defensa y la seguridad, entre otras.

A pesar de los avances en la adopción de la tecnología de Radio Definida por Software (SDR) en las universidades de Ecuador, aún existen desafíos y problemáticas que deben ser abordados para garantizar una implementación efectiva y maximizar los beneficios de esta tecnología en el ámbito de la educación superior. Si bien se puede observar cómo algunas universidades en Ecuador, como la Universidad de las Fuerzas Armadas ESPE y la Universidad Politécnica Salesiana, han comenzado a incorporar prácticas multidisciplinarias y proyectos de emulación de sistemas de comunicaciones

utilizando SDR, es necesario evaluar la extensión y el impacto real de estas iniciativas [2].

Uno de los desafíos en la implementación de la tecnología SDR en la educación superior en Ecuador es la disponibilidad y accesibilidad de los recursos necesarios. Es necesario evaluar la infraestructura requerida, como equipos de SDR, software especializado y laboratorios adecuados, así como la capacitación docente necesaria para utilizar eficazmente esta tecnología en aulas y laboratorios. También es importante considerar aspectos como el costo asociado y las limitaciones presupuestarias que puedan afectar la implementación generalizada de SDR en todas las instituciones educativas.

En la facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, se dispone de equipos funcionales USRP N210, dispositivos de radio definida por software (SDR), que permiten la implementación y el desarrollo de sistemas de comunicaciones personalizados.

Sin embargo, a pesar de las ventajas y beneficios asociados con esta tecnología, existe una falta de guías específicas que utilicen SDR para la enseñanza y sus aplicaciones. Esta carencia limita la experiencia de los estudiantes en la aplicación práctica de los conceptos teóricos de distintas asignaturas utilizando SDR, así como su preparación para las demandas tecnológicas de la industria de las comunicaciones.

## **1.2 Antecedentes investigativos**

La investigación fue realizada partiendo de la información procesada de los distintos repositorios de las entidades educativas de nivel superior, tanto nacionales como internacionales, de la misma manera se ha procesado información proveniente de artículos científicos y papers.

Yessenia Liliana Bermeo Guerrero, en la Universidad de las Fuerzas Armadas ESPE, Sangolquí, en el año de 2022, presentó la investigación con el tema: “Diseño y desarrollo de guías de laboratorios utilizando la herramienta GNU Radio con base a la tecnología de Radio Definido por Software para el aprendizaje y comprensión del componente Sistemas de Comunicaciones”, utilizando el dispositivo ADALM PLUTO

de la empresa Analog Devices y el software de simulación de sistemas dedicados a SDR GNU Radio, creo guías de laboratorios con enfoque a la ayuda de estudiantes para la comprensión de modulaciones analógicas, digitalización de señales, modulación digital y transmisión de señales de radiofrecuencia [3].

Vanessa Aracely Balseca Acuña, en la Universidad Técnica de Ambato, presentó, en el año de 2021, la investigación con el tema: “Sistema de comunicaciones empleando SDR para prácticas multidisciplinarias de la carrera de Ingeniería en Electrónica y Comunicaciones de la Universidad Técnica de Ambato”, en el cual desarrolla un gran número de aplicaciones funcionales usando sistemas de comunicación inalámbrica haciendo uso de distintas tecnologías como Wifi, LTE o radiodifusión. Haciendo uso de equipos SDR como hardware para el sistema pretende reducir costos de equipos, proponiendo una serie de prácticas empleando esta tecnología con el fin de permitir que el estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones pueda conocer las aplicaciones y beneficios que tiene SDR [4].

José Luis Sanz Ruiz, en el año 2020, desarrolló en Escuela Técnica Superior de Ingeniería de Telecomunicación en la ciudad de Murcia el trabajo de investigación titulado: “Implementación de modulaciones analógicas y digitales en GNU Radio”, El proyecto hace uso del software GNU Radio, para proponer unas implementaciones con fin educativo de las modulaciones analógicas y digitales básicas, que se estudian en los planes de ingeniería de telecomunicación. Además, implementa un modulador AM comercial y otro de FM estéreo, utilizando el hardware HackRF One de Great Scott Gadget. Logrando tanto el aprendizaje y manejo de la herramienta para futuros usos, como el refuerzo de los conceptos teóricos vistos en teoría de la comunicación [5].

Mario Daniel Carchi Sañay, en el año 2020, en la Escuela Politécnica Nacional en la ciudad de Quito, presenta el trabajo de investigación titulado: “Implementación de prácticas de laboratorio para la materia de comunicación digital utilizando radio definida por software y GNU Radio”, mediante el uso de dispositivos USRP NI 2990 de la empresa National Instruments y GNU Radio como software de configuración para dispositivos SDR, obtuvo un conjunto de prácticas laboratoristas enfocadas a la transmisión de señales basadas en modulaciones digitales PSK, análisis de canales y técnicas de modulación de espectro ensanchado DSSS [6].

Álvaro Ignacio Monteros Túquerres, en la Escuela Politécnica Nacional, presentó en el año 2019, la investigación con el tema: “Diseño y elaboración de prácticas de laboratorio para la materia de fundamentos de comunicaciones usando radio definida por software”, diseño una guía de prácticas de laboratorio enfocadas al análisis espectral, modulación analógicas AM, FM y PM. Haciendo uso del dispositivo USRP NI 2990 de la empresa National Instruments como hardware y GNU Radio para la configuración y simulación de dispositivos radio definidas por software, presenta diversas propuestas de prácticas en sistemas de comunicaciones trabajando mediante la interfaz gráfica que presenta el software [7].

Rogelio Jaimes Rico y Saul Lazcano Salas, en el año de 2019, publicaron en la Universidad de Guadalajara, México, el artículo con el tema: “SDR y GNU Radio como plataforma para un laboratorio de comunicaciones digitales”, se analizan algunos temas básicos como muestreo, filtrado, modulación y la factibilidad de implementación de los mismos en un esquema SDR, mediante un radio modelo HackRf One en un entorno de GNU Radio, dirigido a laboratorios de comunicaciones digitales en licenciaturas vinculadas a las telecomunicaciones, tales como Ingeniería en Computación o Ingeniería en Sistemas y Comunicaciones [8].

### **1.3 Fundamentación teórica**

Al explicar el concepto de SDR (Software Defined Radio), es necesario tener como base algunos conceptos de transmisión, recepción de sistemas de comunicaciones, dispositivos que operan utilizando una parte del espectro electromagnético. De la misma manera los dispositivos que funcionan con esta tecnología.

#### **1.3.1 Radio Definida por Software**

Un Radio Definido por Software (SDR), es un sistema de comunicación inalámbrica en el cual la mayoría de los componentes esenciales son implementados mediante software en lugar de hardware. Con esta tecnología, se emplea un receptor configurable Zero-IF o low-IF que permite diseñar diversos elementos como mezcladores, filtros, amplificadores, moduladores/demoduladores, detectores, y también sistemas completos como transmisores, receptores, transceptores,

osciloscopios, analizadores de espectros o analizadores vectoriales de redes. Estos parámetros son configurables de manera dinámica, lo que aporta una gran flexibilidad en la creación de sistemas de radiocomunicación [1].

La Figura 1 muestra la distribución en bloques de un equipo USRP

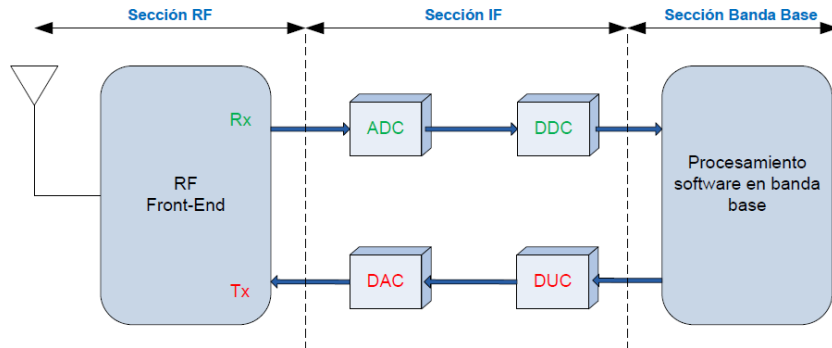


Figura 1. Diagrama de bloques funcionales de SDR [1]

### 1.3.2 Arquitectura de SDR

Un sistema de radio definido por software consta de los componentes mostrados en la Figura 2.

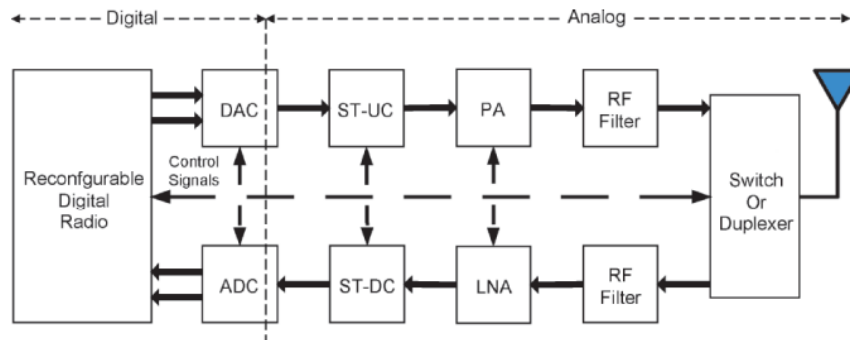


Figura 2. Estructura de SDR [4]

- Una antena que recibe y transmite las señales de RF (Radiofrecuencia).
- Un convertidor de frecuencias que convierte las señales de radiofrecuencias a señales de frecuencia intermedia o viceversa.
- Un convertidor analógico digital (ADC), la cual convierte las señales de frecuencia intermedia IF en señales digitales y viceversa. Y un procesador

digital de señales que realiza el procesamiento de señales digitales tales como la modulación, demodulación, filtrado y codificación entre otros.

### 1.3.3 Hardware SDR

Las características dependerán del equipo y del fabricante, añadiendo capacidades de infraestructura permite a investigadores y estudiantes implementar cualquier sistema de comunicaciones.

En el mercado actual existen distintos equipos basados en SDR, suelen estar desarrollados para aplicaciones en general, pero también existen los que son para aplicaciones específicas siendo estas más comunes para recepción de señales ADS-B o señales de radio FM.

En la Tabla 1. se muestra los dispositivos SDR más comunes disponibles en el mercado

Tabla 1. Dispositivos SDR [9], [10], [11], [12]

Dispositivo	Descripción	Rango de frecuencia
USRP	Dispositivo de RF reconfigurable con procesadores basados en host, FPGA y terminales frontales.	Amplio rango de frecuencias. Depende del modelo.
HackRF One	Periférico SDR capaz de transmitir o recibir señales de radio desde 1 MHz hasta 6 GHz.	1 MHz - 6 GHz
BladeRF	SDR asequible diseñado para estudiantes y entusiastas de RF.	300 MHz - 3.8 GHz
LimeSDR	SDR de bajo costo y código abierto compatible con diversos estándares de comunicación inalámbrica.	100 kHz - 3.8 GHz
RTL-SDR	Radio definida por software RX económica con un ancho de banda de 2.4 MHz [13].	24 MHz



### 1.3.4 Equipos USRP

El NI USRP (Periférico Universal de Radio Definida por Software) es un conjunto de radios definidas por software (SDR) que son completamente programables por el usuario. Estas radios combinan procesadores de propósito general, matrices de puertas programables por campo (FPGA) y frontales de RF. Esto permite a los usuarios diseñar, prototipar e implementar sistemas inalámbricos de manera rápida y eficiente. Con opciones que abarcan desde bajo costo hasta alto rendimiento, así como despliegues robustos y multicanal, el NI USRP destaca como el SDR más popular y versátil disponible en la actualidad [14].

El núcleo de los SDR se encuentra en el software, y la base de la línea de productos NI USRP proviene de los diseños de Ettus Research, una marca de NI, desde el año 2010. Ya sea que tenga experiencia como programador en LabVIEW o prefiera herramientas de desarrollo de código abierto, puede utilizar la línea de productos NI USRP para iniciar rápidamente y desarrollar su aplicación inalámbrica personalizada. El NI USRP es compatible con el conocido marco GNU Radio, gracias a su controlador de código abierto denominado USRP Hardware Driver (UHD) [14].

Además de la versatilidad del software, mediante la línea de productos NI USRP, tiene la capacidad de llevar el código simulado del procesador digital de señales (DSP) en línea y dirigirlo hacia la FPGA para realizar un procesamiento en tiempo real. Puede lograr esto utilizando una infraestructura digital preconstruida, aprovechando tanto LabVIEW FPGA como el robusto marco RF Network on Chip (RFNoC), el cual es compatible con flujos de trabajo comunes de código abierto. Este avance se encuentra incorporado en el equipo USRP N210, desarrollado por la empresa Ettus Research [14].

Las siguientes Tablas 2-9 muestran los productos más populares de la empresa:

Tabla 2. Parámetros técnicos USRP B200/B210 [15]



Modelo	Parámetros
 <p data-bbox="395 994 596 1025">USRP B200/B210</p>	<p data-bbox="703 389 1390 450">Frecuencia de operación: 70 MHz a 6 GHz (B200), 70 MHz a 3.8 GHz (B210)</p> <p data-bbox="703 454 1158 483">Ancho de banda máximo: 56 MHz (B200)</p> <p data-bbox="703 488 911 517">61.44 MHz (B210)</p> <p data-bbox="703 521 1390 582">Velocidad de transmisión de datos: Hasta 25 MS/s (B200), hasta 40 MS/s (B210)</p> <p data-bbox="703 586 1390 647">Resolución de conversión analógica a digital: 12 bits (B200/B210)</p> <p data-bbox="703 651 1390 712">Resolución de conversión digital a analógica: 12 bits (B200/B210)</p> <p data-bbox="703 716 1015 745">Potencia de salida: &gt;10 dBm</p> <p data-bbox="703 750 1086 779">Número de canales: 1 (B200/B210)</p> <p data-bbox="703 784 1238 813">Conectores de antena: 2 conectores SMA hembra</p> <p data-bbox="703 817 1114 846">Interfaz de host: USB 3.0 SuperSpeed</p> <p data-bbox="703 851 1118 880">FPGA: Xilinx Spartan-6 XC6SLX150</p> <p data-bbox="703 884 1390 945">Alimentación: A través del puerto USB o fuente de alimentación externa 6 V</p> <p data-bbox="703 949 975 978">Empresa: Ettus Research</p> <p data-bbox="703 983 1198 1012">Software: GNU Radio, C++ and Python APIs</p> <p data-bbox="703 1016 1038 1046">Dimensiones: 9.7x15.5x1.5 cm</p> <p data-bbox="703 1050 831 1079">Peso: 350 g</p> <p data-bbox="703 1084 951 1113">Costo: \$2,101.00 USD</p>
<b>Funcionalidad</b>	
<p data-bbox="316 1173 1390 1382">Los USRP B200 y B210 son plataformas versátiles que permiten la experimentación con una amplia variedad de señales, incluyendo FM, TV, celular y Wi-Fi. El B200 ofrece un canal de recepción y transmisión, mientras que el B210 amplía esto a dos canales de recepción y transmisión, incluyendo un FPGA y GPIO. Ambos son económicamente accesibles y pueden transmitir hasta 56 MHz de ancho de banda a través de una conexión USB 3.0 de alta velocidad. Son controladores de código abierto que funcionan en diversas plataformas y son compatibles con marcos de software como GNU Radio.</p>	

Tabla 3. Parámetros técnicos USRP N200/N210 [16]

Modelo	Parámetros
 <p data-bbox="443 1839 584 1868">USRP N200</p>	<p data-bbox="738 1570 1206 1599">Frecuencia de operación: 10 MHz a 6 GHz</p> <p data-bbox="738 1603 1198 1632">Ancho de banda máximo: 50 MHz (N200)</p> <p data-bbox="738 1637 1366 1666">Velocidad de transmisión de datos: Hasta 50 MS/s (N200)</p> <p data-bbox="738 1671 1310 1700">Resolución de conversión analógica a digital: 16 bits</p> <p data-bbox="738 1704 1310 1733">Resolución de conversión digital a analógica: 14 bits</p> <p data-bbox="738 1738 1094 1767">Precisión de frecuencia: 2.5 ppm</p> <p data-bbox="738 1771 975 1800">Número de canales: 2</p> <p data-bbox="738 1805 1382 1834">Conectores de antena: 2 conectores SMA hembra por canal</p> <p data-bbox="738 1839 1102 1868">Interfaz de host: Gigabit Ethernet</p> <p data-bbox="738 1872 1246 1901">FPGA: Xilinx Spartan-3A DSP XC3SD3400A</p> <p data-bbox="738 1906 1390 1966">Alimentación: A través del puerto Ethernet o fuente de alimentación externa 6 V</p> <p data-bbox="738 1971 1015 2000">Empresa: Ettus Research</p> <p data-bbox="738 2004 1310 2033">Software: GNU Radio, LabVIEW™ and Simulink™</p> <p data-bbox="738 2038 1054 2067">Dimensiones: 22 x 16 x 5 cm</p>

Modelo	Parámetros
	Peso: 1.2 kg Costo: \$3,354.00 USD
Funcionalidad	
<p>El hardware N200 es ideal para aplicaciones que requieren alta Rendimiento de RF y gran ancho de banda. Tales aplicaciones incluyen prototipos de capa física, espectro dinámico acceso y radio cognitiva, monitoreo de espectro, grabación y reproducción, e incluso despliegue de sensores en red. Los productos de la serie en red ofrecen capacidad MIMO con alto ancho de banda y rango dinámico. La interfaz Gigabit Ethernet sirve como conexión entre el N200/N210 y la computadora central. Esto permite al usuario utilizar 50 MS/s de ancho de banda en tiempo real en las direcciones de recepción y transmisión, simultáneamente (dúplex completo).</p>	


Tabla 4. Parámetros técnicos USRP X300/X310 [17]

Modelo	Parámetros
 <p>USRP X300/X310</p>	<p>Frecuencia de operación: 10 MHz a 6 GHz (X300), 10 MHz a 3.8 GHz (X310)  Ancho de banda máximo: Hasta 160 MHz de ancho de banda por canal (X300/X310)  Velocidad de transmisión de datos: Hasta 160 MS/s (X300), hasta 200 MS/s (X310)  Resolución de conversión analógica a digital: 14 bits (X300/X310)  Resolución de conversión digital a analógica: 16 bits (X300/X310)  Número de canales: 2 (X300/X310)  Conectores de antena: 2 conectores SMA hembra por canal  Interfaz de host: PCIe (X300), 10 Gigabit Ethernet (X310)  FPGA: Xilinx Virtex-6 XC6VLX240T o Virtex-7 XC7V2000T  Alimentación: Fuente de alimentación externa 12 V  Empresa: Ettus Research  Software: GNU Radio, C++, Python, Other third-party frameworks &amp; applications  Consumo de energía: 45 W  Dimensiones: 27.7 x 21.8 x 3.9 cm  Peso: 1.7 kg  Costo: \$9,353.00 USD</p>
Funcionalidad	
<p>Las plataformas de radio definidas por software (SDR) USRP X300 y X310 ofrecen un alto rendimiento y escalabilidad para sistemas de comunicaciones inalámbricas de próxima generación. Con amplio ancho de banda, múltiples interfaces de alta velocidad y programación personalizada de FPGA, estas plataformas son ideales para aplicaciones avanzadas. Además, su arquitectura de software de código abierto es compatible con una variedad de marcos de desarrollo y proyectos de código abierto, lo que las hace altamente versátiles y adaptables para diversas aplicaciones de SDR.</p>	

Tabla 5. Parámetros técnicos USRP E313 [18]

Modelo	Parámetros
 <p data-bbox="448 824 584 853">USRP E313</p>	<p data-bbox="743 365 1390 394">Frecuencia de operación: 70 MHz a 6 GHz, 1 MHz a 6 GHz</p> <p data-bbox="743 396 1114 425">Ancho de banda máximo: 56 MHz</p> <p data-bbox="743 427 1278 456">Velocidad de transmisión de datos: hasta 56 MS/s</p> <p data-bbox="743 459 1310 488">Resolución de conversión analógica a digital: 12 bits</p> <p data-bbox="743 490 1267 519">Resolución de conversión digital a analógica: 12</p> <p data-bbox="743 521 979 551">Número de canales: 2</p> <p data-bbox="743 553 1382 582">Conectores de antena: 2 conectores SMA hembra por canal</p> <p data-bbox="743 584 1390 665">Interfaz de host: • 10/100/1000 BASE-T Ethernet, Integrated GPS receiver, Stereo audio out, mono mic in, Host USB, 9-axis IMU</p> <p data-bbox="743 667 1027 696">FPGA: Zynq-7030 (E313)</p> <p data-bbox="743 698 1390 759">Alimentación: A través del puerto Ethernet o fuente de alimentación externa 5 – 19 V</p> <p data-bbox="743 761 1390 822">Software: UHD version 3.9.2 or later, RFNoC, GNU Radio, C/C++, Python</p> <p data-bbox="743 824 1123 853">Dimensiones: 186 x 280 x 106 mm</p> <p data-bbox="743 855 879 884">Peso: 2.5 kg</p> <p data-bbox="743 887 991 916">Costo: \$7,094.00 USD</p>
<b>Funcionalidad</b>	
<p data-bbox="316 1016 1390 1167">El USRP E313 es un SDR resistente y diseñado para uso en exteriores, con protección IP67 contra polvo y agua. Incorpora filtros de RF, un procesador Xilinx Zynq 7020, GPS para geolocalización, y conectividad USB. Viene con software UHD y es compatible con herramientas como GNU Radio. Ofrece procesamiento en tiempo real de señales de banda ancha mediante el marco RFNoC™, lo que lo hace ideal para aplicaciones al aire libre exigentes.</p>	

Tabla 6. Parámetros técnicos USRP RIO [19]

Modelo	Parámetros
 <p data-bbox="470 1594 595 1624">USRP RIO</p>	<p data-bbox="775 1384 1238 1413">Frecuencia de operación: 10 MHz a 6 GHz</p> <p data-bbox="775 1415 1382 1444">Ancho de banda máximo: 40 MHz, 120 MHz, 160 MHz</p> <p data-bbox="775 1447 1318 1476">Velocidad de transmisión de datos: Hasta 25 MS/s</p> <p data-bbox="775 1478 1342 1507">Resolución de conversión analógica a digital: 12 bits</p> <p data-bbox="775 1509 1299 1538">Resolución de conversión digital a analógica: 14</p> <p data-bbox="775 1541 1010 1570">Número de canales: 2</p> <p data-bbox="775 1572 1390 1632">Conectores de antena: 2 conectores SMA hembra por canal</p> <p data-bbox="775 1635 1174 1664">Interfaz de host: Cabled PCI Express</p> <p data-bbox="775 1666 1267 1695">Alimentación: fuente de alimentación externa</p> <p data-bbox="775 1697 1007 1727">Software: LabVIEW</p>
<b>Funcionalidad</b>	

La plataforma de radio definida por software (SDR) USRP RIO proporciona una solución integrada de hardware y software para crear rápidamente prototipos de sistemas de comunicación inalámbrica de alto rendimiento. Cada dispositivo USRP RIO tiene dos transceptores de RF y un FPGA Xilinx Kintex-7 (410T) grande en un formato de montaje en rack de medio 1U. El Kintex-7 FPGA es un objetivo LabVIEW FPGA configurable que incorpora coprocesamiento DSP48 para aplicaciones de alta velocidad y baja latencia. Con la arquitectura de hardware flexible y el flujo de diseño unificado de LabVIEW, los investigadores pueden crear prototipos más rápido y acortar significativamente el tiempo de obtención de resultados, para crear prototipos de una amplia gama de aplicaciones de investigación avanzada que incluyen entrada múltiple, salida múltiple (MIMO); sincronización de redes heterogéneas; retransmisión LTE; Muestreo por compresión de RF; detección de espectro; radio cognitiva; formación de haces; y búsqueda de dirección.

Tabla 7. Parámetros técnicos USRP E320 [20]


Modelo	Parámetros
 <p data-bbox="464 1216 596 1245">USRP E320</p>	<p data-bbox="770 801 1390 1384">           Frecuencia de operación: 70 MHz a 6 GHz            Ancho de banda máximo: 56 MHz            Velocidad de transmisión de datos: Hasta 25 MS/s            Resolución de conversión analógica a digital: 16 bits            Resolución de conversión digital a analógica: 14            Número de canales: 2 RX, 2TX            Conectores de antena: 2 conectores para antena de tipo SMA (para ambos canales)            Interfaz de host: SFP+ (1/10 GbE, Aurora), RJ45 (1 GbE), Type A USB Host, Micro-USB (serial console, JTAG), GPIO            Alimentación: fuente de alimentación externa 110 – 14, 3 V, A            FPGA: Xilinx Zynq 7045            Software: UHD version 3.14.0 or later, RFNoC, GNU Radio, C/C++, Python            Dimensiones: 173x100x36 mm            Peso: 0.86 kg            Costo: \$8,332.00 USD         </p>
<b>Funcionalidad</b>	
<p data-bbox="316 1480 1390 1809">           El USRP E320 es un avanzado dispositivo de radio definida por software (SDR) que ofrece un rendimiento mejorado en comparación con sus predecesores. Cuenta con cuatro veces más recursos FPGA, lo que mejora la transmisión, sincronización, integración y recuperación de fallos. Utiliza un transceptor MIMO 2x2 de Analog Devices con un amplio rango de frecuencias (70 MHz a 6 GHz y 56 MHz de ancho de banda instantáneo). Además, incorpora un potente procesador de banda base Xilinx Zynq 7045 para cálculos acelerados de FPGA. Los usuarios pueden implementar aplicaciones en el sistema operativo Linux preinstalado o transmitir muestras a computadoras host a través de interfaces de alta velocidad. Esto lo hace ideal para aplicaciones SDR con requisitos de alto rendimiento, como el procesamiento de espectro en tiempo real y la emulación de estaciones base/UE. Además, su tamaño compacto lo hace adecuado para despliegues en manpacks o vehículos aéreos no tripulados.         </p>	

Tabla 8. Parámetros técnicos USRP N310 [21]



Modelo	Parámetros
 <p data-bbox="464 763 600 792">USRP N310</p>	<p data-bbox="772 365 1235 394">Frecuencia de operación: 10 MHz a 6 GHz</p> <p data-bbox="772 396 1158 425">Ancho de banda máximo: 100 MHz</p> <p data-bbox="772 427 1315 456">Velocidad de transmisión de datos: Hasta 25 MS/s</p> <p data-bbox="772 459 1342 488">Resolución de conversión analógica a digital: 16 bits</p> <p data-bbox="772 490 1294 519">Resolución de conversión digital a analógica: 14</p> <p data-bbox="772 521 1118 551">Número de canales: 4 TX, 4 RX</p> <p data-bbox="772 553 1390 611">Conectores de antena: 2 conectores para antena de tipo SMA (para ambos canales)</p> <p data-bbox="772 613 1390 701">Interfaz de host: 2 SFP+ (1/10 GbE, Aurora), RJ45 (1 GbE), 1 Type A USB Host, 1 Micro-USB (serial console, JTAG)</p> <p data-bbox="772 703 1390 761">Alimentación: fuente de alimentación externa 12 V, 7 A DC</p> <p data-bbox="772 763 1289 792">FPGA: Xilinx Kintex-7 XC7K325T-2FFG900C</p> <p data-bbox="772 795 1390 853">Software: UHD version 3.11.0.0 or later, RFNoC, GNU Radio, C/C++, Python</p> <p data-bbox="772 855 1198 884">Dimensiones: 357.1 x 211.1 x 43.7 mm</p> <p data-bbox="772 887 919 916">Peso: 3.13 kg</p> <p data-bbox="772 918 1031 947">Costo: \$17,247.00 USD</p>
<b>Funcionalidad</b>	
<p data-bbox="316 1055 1390 1323">El USRP N310 es una radio definida por software en red que proporciona confiabilidad y tolerancia a fallas para la implementación en sistemas inalámbricos distribuidos y de gran escala. Este DEG es uno de los dispositivos de mayor densidad de canales disponibles, utilizando transceptores RFIC AD9371 duales Analog Devices. Compatibilidad con 1 GbE, 10 GbE y Aurora las interfaces sobre puertos duales SPF+ permiten un alto rendimiento Transmisión de IQ a una PC anfitriona o coprocesador FPGA. El USRP N310 también presenta una nueva generación de software USRP que simplifica el control y gestión de múltiples dispositivos a través de la red con la capacidad única de administrar remotamente tareas como como depuración, actualización de software, reinicio, reinicio a estado de fábrica, depuración y monitoreo de PC host/ARM.</p>	

Tabla 9. Parámetros técnicos USRP N320 [22]

Modelo	Parámetros
 <p data-bbox="464 1850 600 1879">USRP N320</p>	<p data-bbox="772 1541 1222 1570">Frecuencia de operación: 3 MHz a 6 GHz</p> <p data-bbox="772 1572 1227 1601">Ancho de banda máximo: Hasta 200 MHz</p> <p data-bbox="772 1603 1315 1632">Velocidad de transmisión de datos: Hasta 25 MS/s</p> <p data-bbox="772 1635 1342 1664">Resolución de conversión analógica a digital: 12 bits</p> <p data-bbox="772 1666 1294 1695">Resolución de conversión digital a analógica: 14</p> <p data-bbox="772 1697 1118 1727">Número de canales: 2 TX, 2 RX</p> <p data-bbox="772 1729 1390 1787">Conectores de antena: 2 conectores para antena de tipo SMA (para ambos canales)</p> <p data-bbox="772 1789 1278 1818">Interfaz de host: 1 Gigabit Ethernet, 1 USB 3.0</p> <p data-bbox="772 1821 1390 1879">Alimentación: fuente de alimentación externa 12 V, 7 A DC</p> <p data-bbox="772 1881 1050 1910">FPGA: Xilinx Zynq 7100</p> <p data-bbox="772 1912 1390 1971">Software: UHD version 3.14.0 or later, RFNoC, GNU Radio, C/C++, Python</p> <p data-bbox="772 1973 1198 2002">Dimensiones: 357.1 x 211.1 x 43.7 mm</p> <p data-bbox="772 2004 919 2033">Peso: 3.13 kg</p>

<b>Modelo</b>	<b>Parámetros</b>
	Costo: \$19,642.00 USD
<b>Funcionalidad</b>	
El USRP N320 es una radio definida por software en red que proporciona confiabilidad y tolerancia a fallas para la implementación en sistemas inalámbricos distribuidos y de gran escala. Esto es un SDR de alto rendimiento que utiliza un diseño RF único por Ettus Research para proporcionar 2 canales RX y 2 TX en un factor de forma RU de medio ancho. Cada canal proporciona a 200 MHz de ancho de banda instantáneo, y cubre un rango de frecuencia ampliado de 3 MHz a 6 GHz. El USRP N320 aprovecha los desarrollos de software recientes en UHD para simplificar el control y la gestión de múltiples dispositivos a través de la red con la capacidad única de administrar de forma remota tareas como depuración, actualización software, reinicio, restablecimiento al estado de fábrica y monitorear la salud del sistema.	

### 1.3.5 USRP N210

Los modelos Ettus Research USRP N200 y N210 representan la categoría de hardware de mayor capacidad dentro de la familia de productos USRP™ (Periférico Universal de Radio Definida por Software), lo que permite a los ingenieros diseñar e implementar de manera eficiente sistemas de radio flexibles mediante software. Estos dispositivos N200 y N210 son especialmente adecuados para aplicaciones que demandan un alto rendimiento en RF y un amplio ancho de banda. Dichas aplicaciones abarcan desde prototipos de capa física, acceso dinámico al espectro y radio cognitiva, hasta monitoreo de espectro, grabación y reproducción, e incluso implementaciones de sensores en red [23].

La serie USRP N210 presenta una capacidad de procesamiento con un amplio ancho de banda y un elevado rango dinámico. A través de la interfaz Gigabit Ethernet, la serie USRP N210 facilita una velocidad de transmisión de hasta 50 MS/s en ambas direcciones (con muestras de 8 bits). Estas características, junto con la capacidad de MIMO plug-and-play, convierten a la serie USRP en una opción óptima para sistemas de radio definidos por software que demandan un rendimiento exigente [24].

La Tabla 10 describe al equipo USRP N210:

Tabla 10. Características técnicas del equipo USRP N210 [16], [24]

<b>CARACTERÍSTICAS</b>	<b>USRP N210</b>
<b>Interfaz</b>	Gigabit Ethernet
<b>Rango de Frecuencia</b>	DC - 6 GHz con placa secundaria adecuada
<b>Ancho de banda</b>	50 MHz de ancho de banda de RF con muestras de 8 bits 25 MHz de ancho de banda de RF con muestras

CARACTERÍSTICAS	USRP N210
	de 16 bits
Convertidor análogo-digital	14 bits
Convertidor digital-análogo	16 bits
Modo de Transmisión	Full dúplex
Máxima velocidad	hasta 50 MS/s
Número de canales de salida	2
Número de canales de entrada	2
Clock	Posibilidad de bloqueo a referencia de reloj externa de 5 o 10 MHz
FPGA	Xilinx Spartan-3A DSP XC3SD3400A
Software compatible	GNU Radio, LabVIEW™ Simulink™
Sistema Operativo	Windows Linux Mac OS X

La Figura 3 muestra el plano frontal del equipo.



Figura 3. USRP N210 Series [16]

La descripción de cada una de las interfaces disponibles se puede observar en la Tabla 11

Tabla 11. Descripción del panel frontal del dispositivo [4]

CONECTOR	DESCRIPCIÓN
1 REF CLOCK	Entrada de señal de referencia para oscilador local; señal máxima de 10 MHz con potencias de 0 dBm hasta 15 dBm.
2 PPS IN	Entrada de referencia de temporización de pulso.
3 RX1	Transmisor de señales de radio frecuencia, tiene un conector SMA (hembra) con una impedancia de 50 Ω.



<b>4 RX2</b>	Receptor de señales de radio frecuencia, tiene un conector SMA (hembra) con una impedancia de 50 $\Omega$ .
<b>5 MIMO EXPANSION</b>	Ranura de expansión mimo empleado para realizar la conexión de USRP en paralelo.
<b>6 GB ETHERNET</b>	Puerto Gigabit Ethernet permite la comunicación con la PC.
<b>7 POWER</b>	Entrada de la alimentación del dispositivo, trabaja con 6 voltios y 1.3 amperios

### 1.3.6 Arquitectura del equipo USRP N210

Los módulos del equipo USRP tienen una función específica y se comunican entre los demás a través de las interfaces definidas, a continuación, la Figura 4 presenta la estructura interna que conforma al equipo.

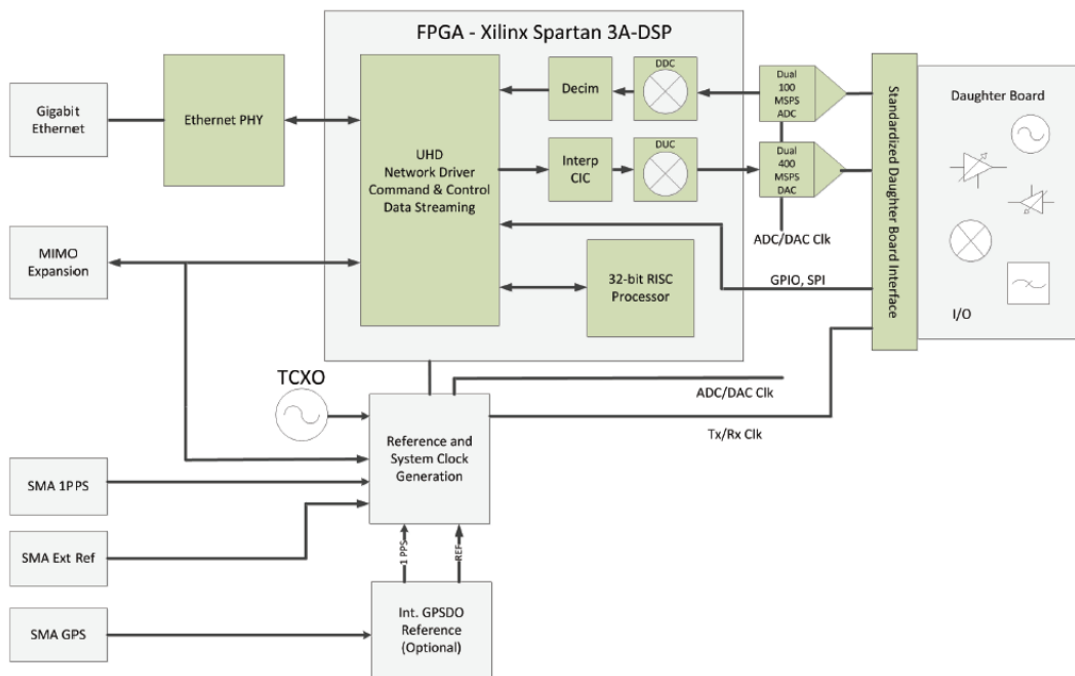


Figura 4. Arquitectura USRP N210 SERIES [16]

- **FPGA – Xilinx Spartan 3A-DSP**

Es el componente principal del USRP N210, que se encarga de procesar las señales digitales que provienen o van hacia el ADC, el DAC y el host. La FPGA tiene recursos

lógicos, de memoria y DSP suficientes para implementar funciones personalizadas, como filtros, moduladores, demoduladores, codificadores, decodificadores, etc [25].

- ***Daughter Board***

Es un bloque que contiene una tarjeta hija, que es una placa de circuito impreso que se conecta al sistema principal para ampliar sus funcionalidades o adaptarse a diferentes bandas de frecuencia [25].

- ***Referencia de frecuencia TCXO***

Es el oscilador controlado por temperatura que genera la señal de reloj para el USRP N210.

- ***Módulo GPSDO opcional***

Es el oscilador disciplinado por GPS que se puede utilizar para mejorar la precisión del reloj del USRP N210.

- ***E/S analógicas y digitales auxiliares***

Son las entradas y salidas adicionales que se pueden utilizar para comunicarse con otros dispositivos o sensores externos.

- ***Reference Generation & System Clock***

Es el módulo que se encarga de generar la señal de reloj para el USRP N210. El módulo puede utilizar una referencia interna o externa. La referencia interna es un oscilador controlado por temperatura (TCXO) que tiene una precisión de 2.5 ppm. La referencia externa puede ser un oscilador disciplinado por GPS (GPSDO) opcional que tiene una precisión de 0.01 ppm [25].

- ***GPSDO Reference***

Es el oscilador disciplinado por GPS opcional que se puede utilizar para mejorar la precisión del reloj del USRP N210. El GPSDO tiene una precisión de 0.01 ppm, lo que significa que la frecuencia puede variar hasta 0.01 partes por millón respecto al estándar GPS mundial [25].

### 1.3.7 Software y controladores

Independientemente de las preferencias de herramientas de desarrollo, los SDR de USRP funcionan con la gama más amplia de flujos de trabajo de software del mercado.

Existen dos opciones de controladores para los dispositivos NI-USRP, siendo una de ellas el USRP Hardware Driver (UHD), y la elección depende de las preferencias en cuanto a las herramientas de desarrollo. Al optar por NI-USRP, es posible llevar a cabo la programación de la Radio Definida por Software (SDR) de USRP mediante LabVIEW. La Figura 8 ilustra el entorno de desarrollo distintivo y fácil de utilizar proporcionado por NI, que facilita la creación rápida de sistemas de medición. Esto permite la programación tanto del host como de la FPGA en un solo entorno de desarrollo, utilizando el estilo de programación basado en flujo de datos característico de LabVIEW [14].

UHD es compatible con todos los SDR de USRP, y la división de Investigación y Desarrollo de NI lo distribuye bajo licencias de código abierto. Este software simplifica el desarrollo de aplicaciones en los SDR de USRP mediante C/C++ y proporciona soporte multiplataforma para diversos entornos y marcos de desarrollo estándar en la industria, como RFNoC, GNU Radio, Python, y las herramientas MathWorks MATLAB y Simulink. Como un participante activo en comunidades de código abierto, incluida la Fundación GNU Radio, NI trabaja continuamente para impulsar la investigación en comunicaciones inalámbricas a través del ecosistema de código abierto. La Tabla 12 detalla los dos controladores utilizados para la programación en tarjetas SDR [14].

Tabla 12. Descripción general del controlador NI-USRP y del paquete UHD de Ettus Research [26]

	<b>NI-USRP</b>	<b>UHD</b>
Sistema Operativo	Windows NI Linux Real-Time	Windows Linux Mac OS
Host	LabVIEW 20XX	GNU Radio C/C++ Matlab/Simulink Python
FPGA	LabVIEW FPGA Module	VHDL Verilog

	<b>NI-USRP</b>	<b>UHD</b>
		HDL Coder RFNoC (Open-Source FPGA Framework)

*a. GNU Radio*

En caso de preferir los lenguajes tradicionales de diseño para FPGA, UHD proporciona código de FPGA gratuito y de código abierto para todos los dispositivos USRP. Una opción adicional dentro de UHD es un marco de FPGA llamado RF Network-on-Chip (RFNoC), que ofrece un enfoque uniforme para implementar código de DSP de FPGA en un entorno de código abierto. Un software ampliamente reconocido y utilizado por desarrolladores es GNU Radio; la Figura 5 ilustra cómo se integran el software RFNoC, el controlador UHD y la FPGA en un marco de software libre para el procesamiento de señales. En cada etapa de la integración se muestran los lenguajes y componentes utilizados. [26].

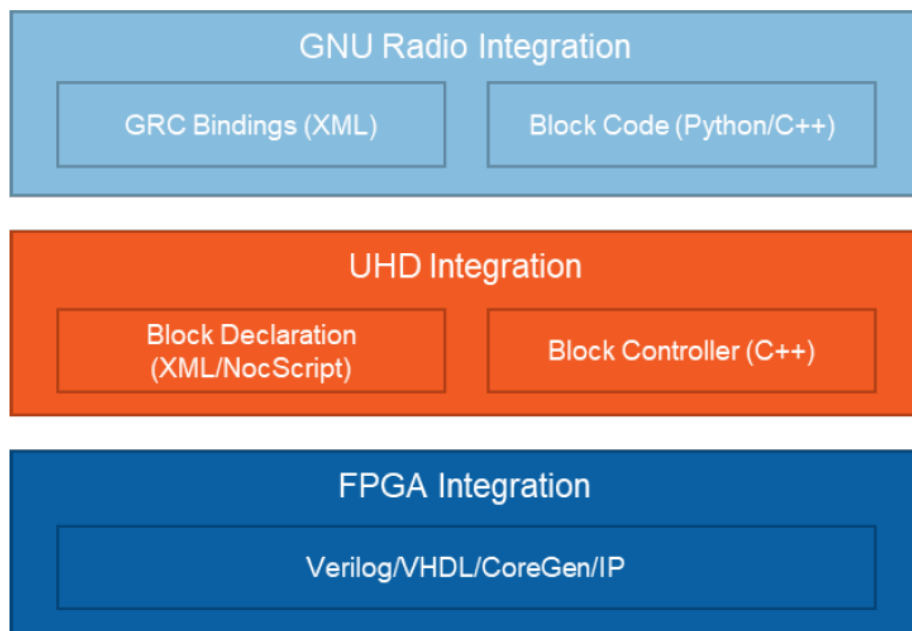


Figura 5. Pilar de Software RFNoC [26]

GNU Radio constituye un conjunto de archivos y aplicaciones organizados en bibliotecas, que posibilitan la manipulación de señales mediante procesamiento digital. Estas bibliotecas permiten llevar a cabo el diseño de sistemas definidos por software cuando se conecta una computadora a un SDR. Asimismo, es viable trabajar con la

tarjeta de audio, entre otros dispositivos. GNU Radio es compatible con sistemas operativos GNU/Linux, como Ubuntu, y puede instalarse en Mac y Windows. No obstante, la instalación en este último sistema operativo puede resultar más compleja, requiriendo la herramienta Cygwin, cuya finalidad es adaptar software de sistemas POSIX a Windows. Los diseños elaborados en GNU Radio se programan en Python, un lenguaje de programación interpretado y orientado a objetos, lo que significa que no se compila, sino que el sistema operativo lo ejecuta directamente. [1].

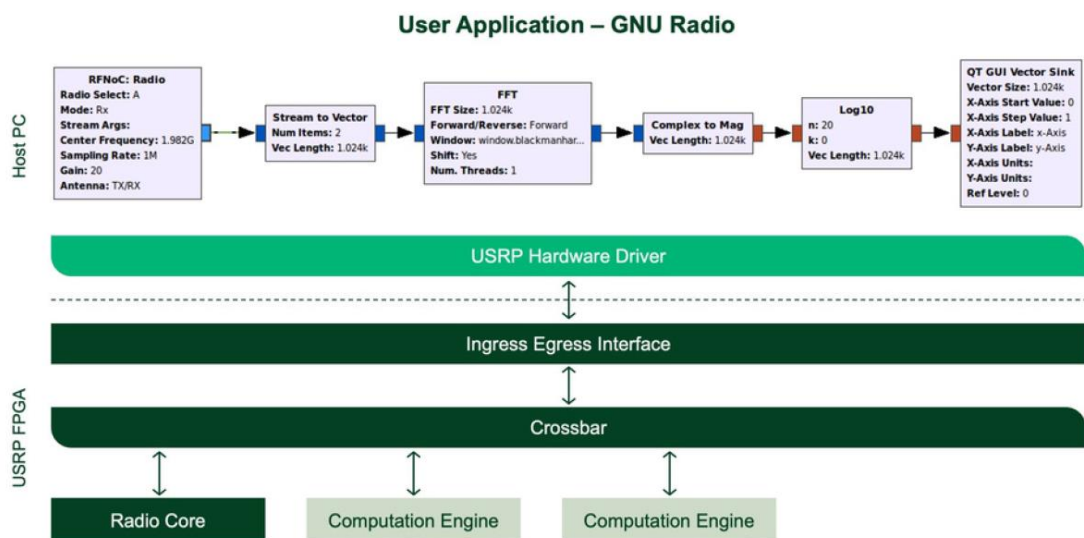


Figura 6. Conexión usuario-USRP a través de GNU Radio [14]

La Figura 7 muestra la arquitectura y el proceso en el diseño de sistemas en GNU Radio.

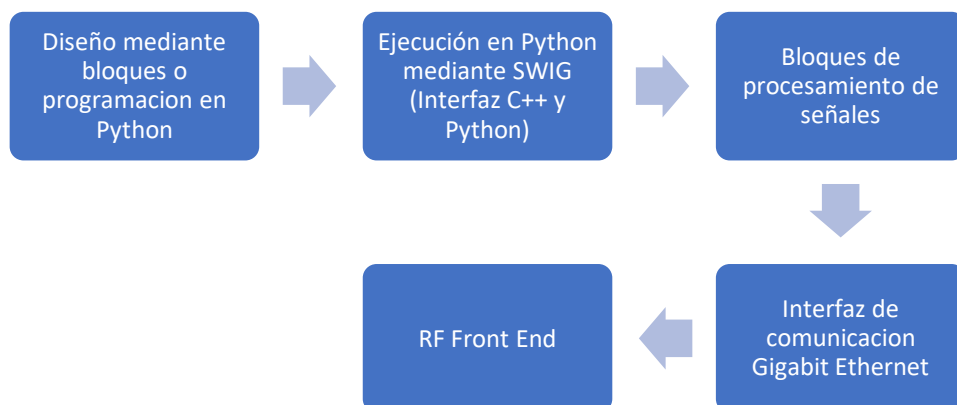


Figura 7. Arquitectura de GNU Radio [1]

## b. LabVIEW

LabVIEW, que significa Laboratory Virtual Instrumentation Engineering Workbench, representa una plataforma y un entorno de desarrollo diseñados para un lenguaje de programación visual creado por National Instruments, conocido como "G". Este lenguaje gráfico se emplea de manera frecuente en LabVIEW para llevar a cabo tareas como la adquisición de datos, el control de instrumentación y la automatización industrial. LabVIEW es compatible con diversas plataformas, incluyendo Microsoft Windows, UNIX, Linux y Mac OS X [27].

Es posible realizar la programación de FPGAs de forma más intuitiva, incluso sin experiencia en lenguajes de descripción de hardware, mediante el uso de LabVIEW. Este entorno proporciona un lenguaje de programación gráfico que refleja el paralelismo inherente al hardware. LabVIEW facilita la dirección de procesadores multicore de propósito general, NI Linux en tiempo real y FPGAs, a la vez que se integra de manera estrecha con hardware SDR [26].

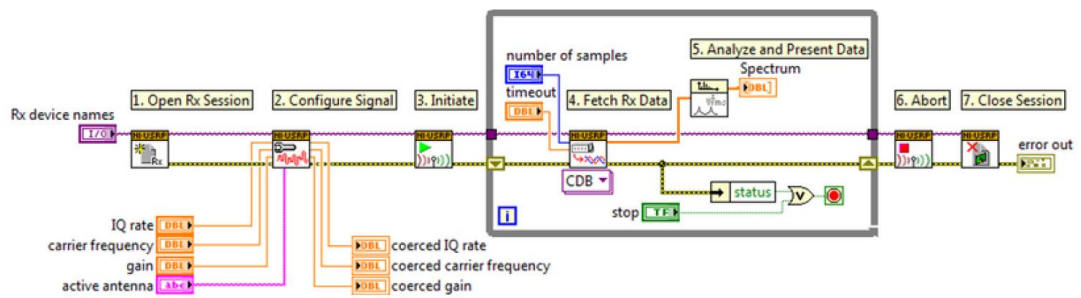


Figura 8. USRP API de LabView [14]

SDR hace alusión a la tecnología en la cual se emplean módulos de software que se ejecutan en una plataforma de hardware genérica para llevar a cabo funciones de radio. La Figura 9 muestra la combinación del hardware NI USRP con el software LabVIEW, logrando así la flexibilidad y funcionalidad necesarias para crear una plataforma de prototipado rápido que abarca el diseño de la capa física, la grabación y reproducción de señales inalámbricas, la inteligencia de señales, la validación de algoritmos y más [28].

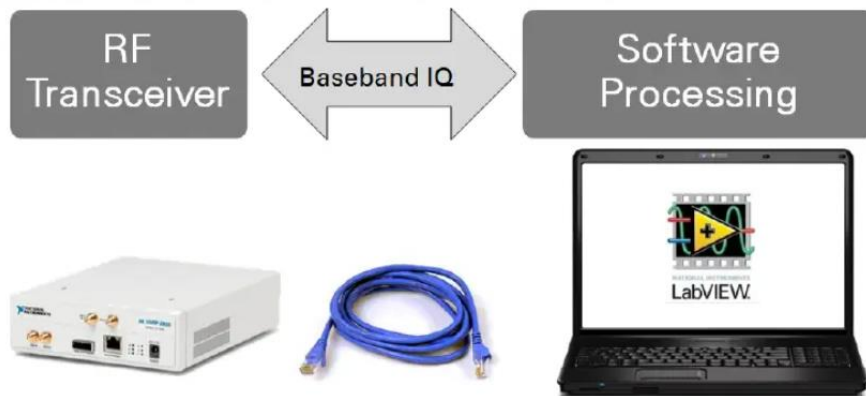


Figura 9. Comunicación entre USRP y NI LabVIEW [28]

### c. Matlab & Simulink

MATLAB es un entorno de programación y software de alto nivel ampliamente utilizado en diversos campos como la ingeniería, las ciencias, la economía, entre otros. Su destacada característica radica en su capacidad para realizar cálculos numéricos y simbólicos, así como en la visualización de datos y el desarrollo de algoritmos. Proporciona un lenguaje de programación propio junto con una amplia variedad de funciones y herramientas especializadas diseñadas para el análisis y procesamiento de datos, así como para la simulación y modelado de sistemas. Además, cuenta con un entorno de desarrollo integrado (IDE) que facilita la escritura, ejecución y depuración del código. Con el soporte de MATLAB y Simulink para el hardware de SDR de uso más frecuente, puede comunicarse con plataformas de SDR para realizar pruebas de radio-in-the-loop, prototipado y aprendizaje práctico [29].

La Figura 10 representa la conexión mediante SDR y Matlab.

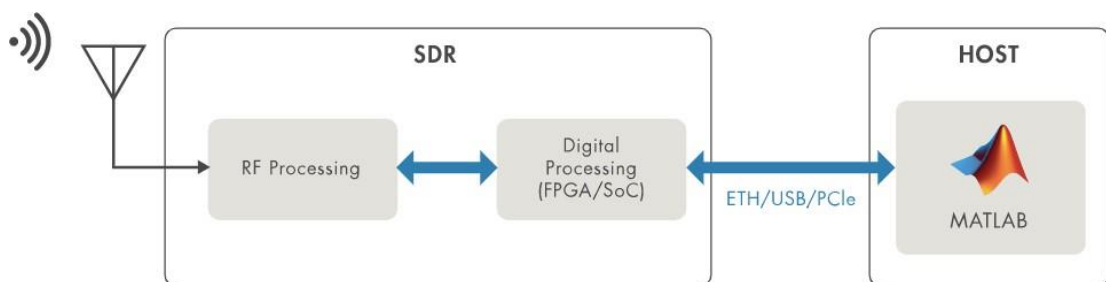


Figura 10. Diagrama de bloques de componentes SDR y cómo se conectan con MATLAB [30]

### **1.3.8 Sistema de comunicaciones**

Un sistema de comunicaciones se compone de dispositivos diseñados para transmitir, emitir y recibir señales de diversos tipos, como voz, datos, audio, video, entre otros. Estas señales pueden ser de naturaleza tanto digital como analógica. Se puede desglosar de manera sencilla en tres componentes esenciales: un transmisor, encargado de generar la señal deseada y prepararla para su transmisión a través del canal mediante técnicas como modulación, filtrado, codificación, etc.; un medio de transmisión, que constituye el canal por el cual la señal viajará, pudiendo ser desde fibras ópticas y cables coaxiales hasta el aire; y por último, un receptor, cuya función es llevar a cabo el proceso inverso al del transmisor con el propósito de reconstruir la señal de manera que se asemeje lo más posible a la original [31].

Los sistemas de comunicaciones se basan en la teoría de sistemas lineales para modelar y analizar el comportamiento de las señales en el dominio del tiempo y la frecuencia. El Procesamiento digital de señales se utiliza para procesar y manipular señales, aplicando algoritmos de filtrado, modulación, demodulación y corrección de errores. Las materias de Comunicaciones Analógicas y Comunicaciones Digitales proporcionan los fundamentos teóricos para comprender cómo se transmiten y reciben señales tanto en dominios analógicos como digitales. Por último, en Sistemas Inalámbricos, la teoría se aplica para diseñar sistemas de comunicaciones sin cables, considerando aspectos como la propagación de ondas electromagnéticas, la modulación digital y las técnicas de acceso al medio para redes inalámbricas. La combinación de estos conocimientos permite el desarrollo y la optimización de sistemas de comunicaciones eficientes.

La relación que existe entre estas áreas académicas está representada en la Figura 11



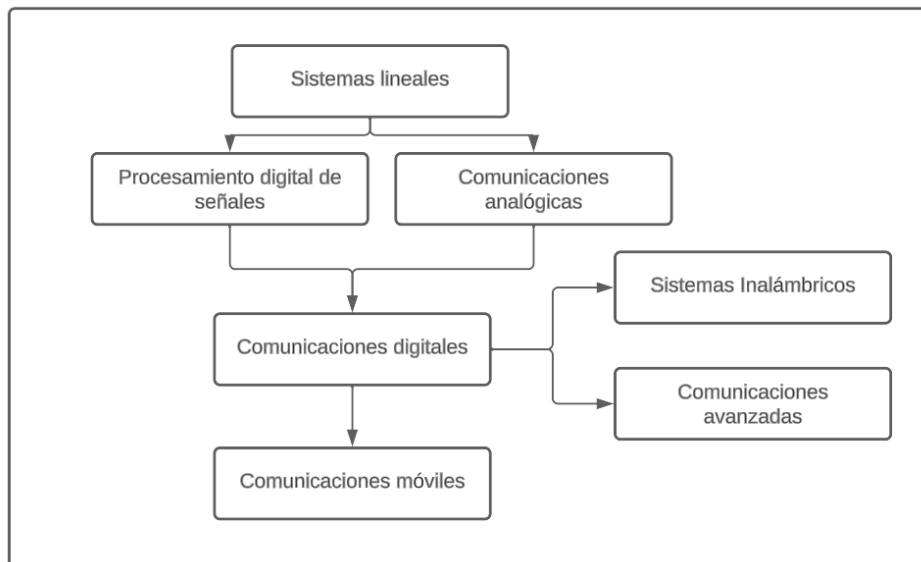


Figura 11. Áreas académicas necesarias para la comprensión de los sistemas de comunicaciones

Este proyecto se centra en áreas académicas que son fundamentales para el entendimiento de los estudiantes hacia los sistemas de comunicaciones, por lo que se abarcan asignaturas como los sistemas lineales, procesamiento digital de señales, comunicaciones digitales y analógicas y sistemas inalámbricos, debido a que se trata de diseñar explicaciones prácticas de los distintos procesos por la cual la información debe pasar para ser transmitida como el filtrado, la modulación analógica y digital, la recepción de señales de radiofrecuencia, entre otros.

## 1.4 Objetivos

### 1.4.1 Objetivo general

- Desarrollar Guías de prácticas de laboratorio interdisciplinarias en sistemas de comunicaciones utilizando radio definida por software

### 1.4.2 Objetivos específicos

- Analizar las características y funcionalidades de los equipos USRP para el análisis y proceso de señales en sistemas de comunicaciones.

- Diseñar experimentos de laboratorio que utilice radio definida por software para la enseñanza de temas aplicados al campo de las Telecomunicaciones.
- Documentar el procedimiento de forma detallada de los experimentos de laboratorio.

## CAPÍTULO II.- METODOLOGÍA

### 2.1 Materiales

Para el desarrollo de este proyecto de investigación, se utilizaron equipos USRP N210 como el elemento de hardware principal para la realización de diversas prácticas de laboratorio. Estos dispositivos, que forman parte de la familia Universal Software Radio Peripheral (USRP), brindan una plataforma adaptable y personalizable para la implementación de sistemas de radio definidos por software (SDR). La conexión principal entre los equipos USRP y la estación de trabajo se realizó mediante un cable Gigabit Ethernet, permitiendo la transferencia eficaz de datos entre el hardware y la computadora central. Además, se instaló el paquete de software UHD (USRP Hardware Driver), que proporciona una interfaz de bajo nivel para el control y la configuración de los equipos USRP, así como una capa de abstracción para facilitar la integración con GNU Radio. Los equipos USRP están equipados con dos antenas VERT900, antena del tipo dipolo con polarización lineal que trabaja a frecuencias de 824, 960, 1710 y 1990 MHz y su conexión es mediante conector SMA.

En lo que respecta al entorno de software, se escogió GNU Radio como la herramienta principal para el diseño y la implementación de los bloques de procesamiento de señales. Este software de código abierto ofrece un marco flexible para el desarrollo de sistemas de radio definidos por software. Asimismo, todo el trabajo se ejecutó sobre el sistema operativo Ubuntu virtualizado a través del programa VirtualBox, proporcionando un entorno de ejecución sólido y altamente configurable. Para prácticas específicas, como la construcción de redes GSM, se emplearon servicios como OpenBTS, Sipauthserve, smqueue y Asterisk. Estos componentes de software jugaron roles fundamentales en la simulación y el análisis de sistemas de comunicación avanzados durante el transcurso de las investigaciones.

## **2.2 Métodos**

### **2.2.1 Modalidad de la investigación**

El trabajo de investigación presentado se caracterizó por centrarse en la investigación aplicada, con el objetivo de abordar y resolver un problema específico mediante la aplicación de conocimientos teóricos y prácticos.

La investigación bibliográfica se justifica debido a la necesidad de adquirir información relevante, principalmente en lo que respecta al uso de equipos USRP y software comúnmente empleado en configuraciones y control de SDR.

La investigación de campo se llevó a cabo en laboratorios académicos afiliados al ámbito de profesionalización de la carrera de Telecomunicaciones, que forma parte de la Facultad de Ingeniería en Sistemas Electrónica e Industrial.

### **2.2.2 Recolección de información**

Con el propósito de recopilar información, se recurrió a diversos repositorios de las universidades del país, así como a libros, revistas de difusión científica en línea y otras bases de datos confiables. El objetivo era asegurar que dichas fuentes fueran pertinentes y capaces de contribuir de manera significativa al desarrollo del proyecto.

### **2.2.3 Procesamiento y análisis de datos**

Para poder procesar y analizar los datos obtenidos mediante la investigación se procedió realizar los siguientes pasos:

- Recolección de la información obtenida.
- Revisión de la información disponible.
- Análisis de fundamentos teóricos adquiridos dirigidos a los ámbitos de interés.

- Deducción de la información más importante que ayude al correcto desarrollo del proyecto de investigación y que permita dar una solución adecuada a la propuesta.

## CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

### 3.1 Desarrollo de la propuesta

Mediante el uso de equipos USRP se dio paso el desarrollo de una guía de prácticas de laboratorio con las que los estudiantes de pregrado complementen los conocimientos teóricos aplicados a temas relevantes de la carrera de ingeniería en Telecomunicaciones, lo que proporcionó una relación teoría-práctica mediante el uso de software y hardware sirviendo como una introducción al área de dispositivos de radio definida por software.

El desarrollo de este proyecto de guías de prácticas de laboratorio interdisciplinarias para sistemas de comunicaciones utilizando radio definida por software se basó en las etapas que se muestran en la Figura 12.

- **Inicio.** Para el desarrollo de este trabajo es necesario adquirir conocimientos previos y básico en temas como fundamentos de electrónica y comunicaciones, conocimiento en el uso de software y de forma necesaria aspectos basados en matemáticas, sistemas y algebra lineal. De la misma manera para un excelente entendimiento en el uso de tecnicismos el trabajo está dirigido a estudiantes con un perfil académico basado en el área de ingeniería e Telecomunicaciones
- **Desarrollo.** Mediante dispositivos SDR y software libre se plantea el desarrollo e implementación de prácticas que demuestren temas de interés adquiridas en distintas áreas de conocimiento en la carrera de telecomunicaciones, facilitando de una manera práctica e intuitiva la visualización de aspectos teóricos en las áreas de ingeniería lo que facilitara la comprensión y uso de los distintos sistemas a desarrollar. Para ello se hace uso de equipos especializados en radiofrecuencia, así como distintos softwares que permiten el control y configuración de sistemas de comunicaciones.
- **Finalización.** Como resultados generales se obtiene una documentación física y explicativa de las prácticas desarrolladas con el fin de que los estudiantes adquieran el conocimiento necesario para desarrollar habilidades en el análisis de sistemas de

comunicaciones al igual que en el manejo de equipos basados en radio definidos por software.

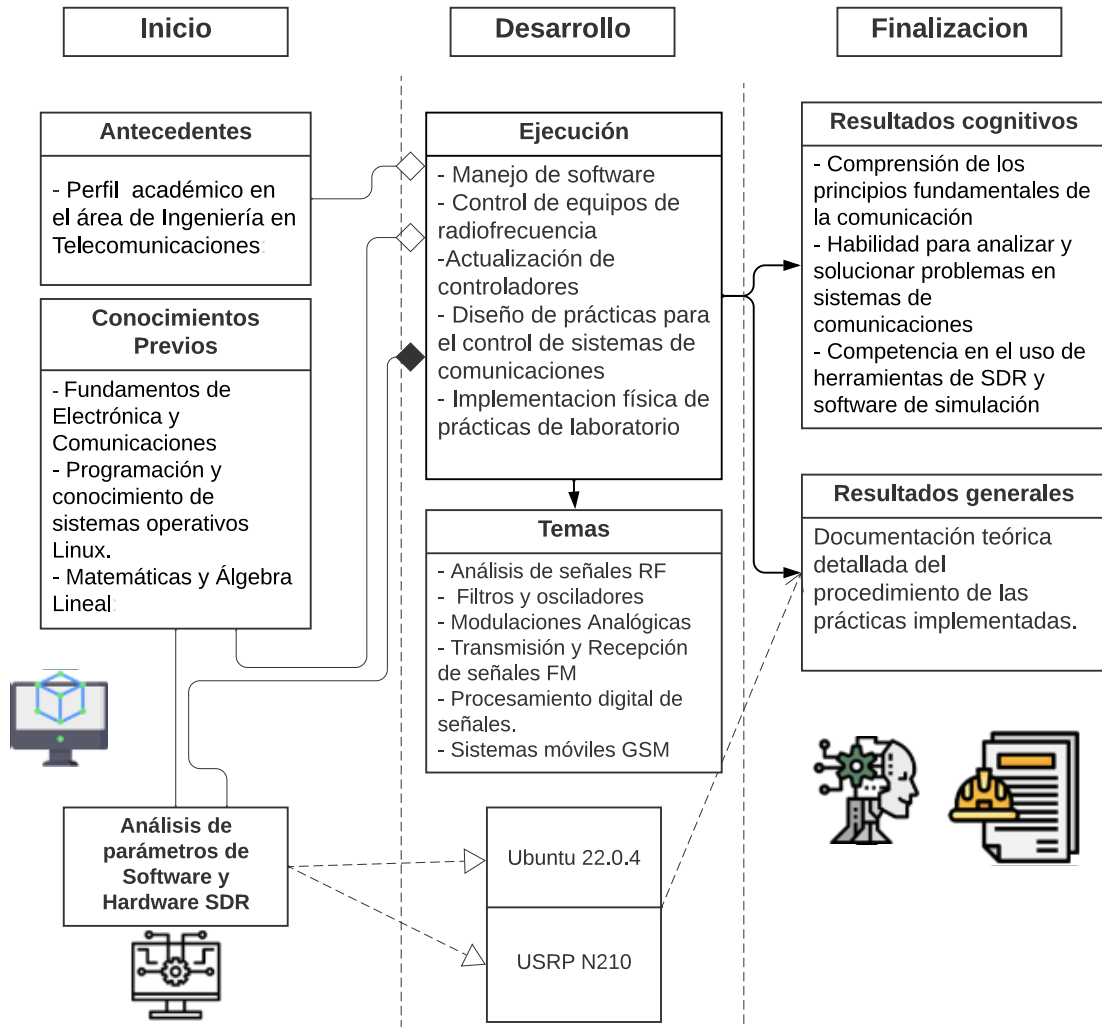


Figura 12. Diagrama del proceso del desarrollo de la propuesta

### 3.2 Desarrollo de prácticas

Se presentan a continuación las prácticas de laboratorio desarrolladas, en este espacio se encuentra el objetivo de la práctica, una introducción del tema a trabajar y los pasos para completar el laboratorio.

### 3.2.1 Práctica 1: Introducción a GNU Radio y USRP

#### Objetivo

Esta práctica tiene como objetivo conocer los fundamentos y conceptos de equipos USRP al igual que del software GNU Radio, esta práctica presenta la instalación de software de virtualización, creación de máquina virtual, instalación del programa GNU Radio y los paquetes necesarios para los equipos USRP.

#### Materiales y equipos

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio
- SDR USRP N210
- Antenas
- Cable Gigabit Ethernet

La Figura 13 muestra el proceso del desarrollo de la práctica.

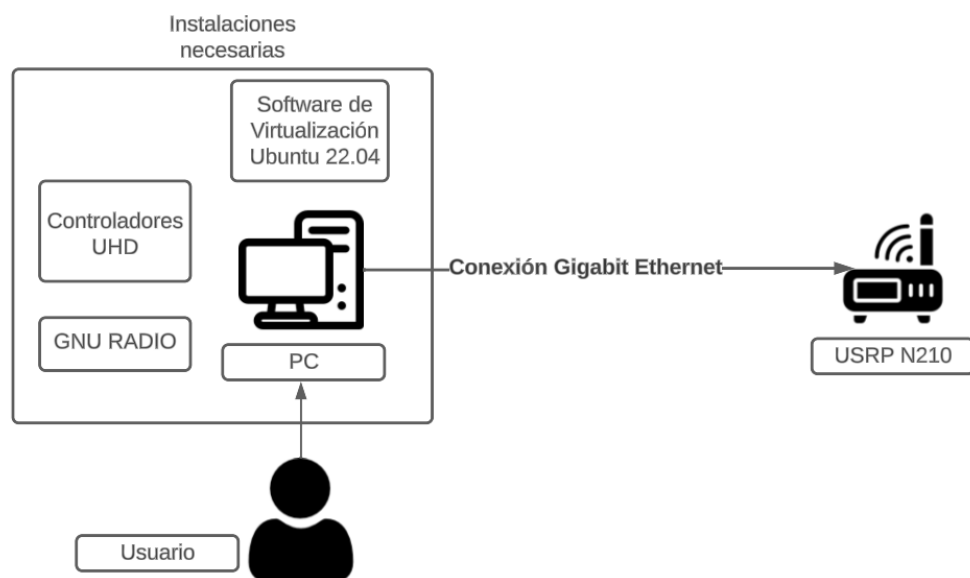


Figura 13. Diagrama de conexión e instalación necesarias



## Procedimiento

### 1. Instalación de software de virtualización

Para realizar la instalación de una máquina virtual se necesita de un software de virtualización instalado previamente en la PC, algunas opciones son:

- Virtual Box
- VMware Workstation

Se hará énfasis en Virtual Box, dado que la máquina virtual que se maneja está basada en las interfaces de red que maneja Virtual Box.

Para instalarlo basta con acudir a la página oficial de Virtual Box <https://www.virtualbox.org>, descargarlo dependiendo del sistema operativo a usar, en este caso Windows.

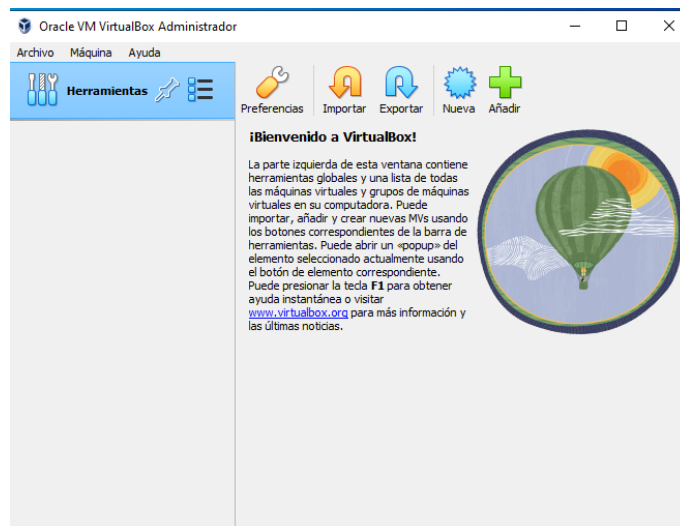


Figura 14. Ventana de inicio Virtual Box

### 2. Instalación de Ubuntu

Una vez instalado el software de virtualización, se deberá instalar Ubuntu 22.04, el cual lo podemos encontrar la imagen ISO en la página oficial de Ubuntu.

<https://old-releases.ubuntu.com/releases/22.04.1/>

Cuando la imagen se haya descargado, volvemos a VirtualBox, seleccionamos la opción Nueva, para crear una máquina virtual e iniciar la instalación del sistema operativo.

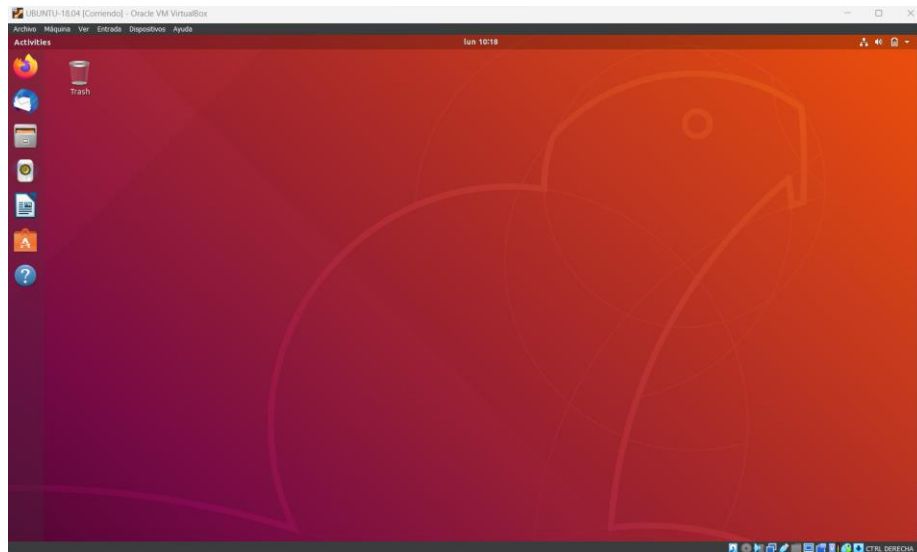


Figura 15. Ubuntu 22.04

### 3. Instalación de GNU RADIO

Mediante el terminal del sistema. Ingresaremos al modo super usuario o root colocando `sudo su` en el terminal, pedirá la contraseña del sistema para continuar.

Agregamos el repositorio de la versión requerida a la lista de repositorios del sistema operativo (SO).

***sudo add-apt-repository ppa:gnuradio/gnuradio-releases***

Actualizamos la lista de repositorios del SO, lugar de donde se accederá a la descarga e instalación del software.

***sudo apt-get update***

Instalamos el software necesario (tomará la versión del último repositorio descargado), se instalará la versión más actual del repositorio del SO por defecto si no se especifica una versión a descargar.

Ingresamos

*sudo apt-get install gnuradio*

#### 4. Instalación de drivers UHD

Mediante uso de repositorios:

*sudo add-apt-repository ppa:ettusresearch/uhd*

*sudo apt-get update*

*sudo apt-get install uhd-host*

#### 5. Conexión USRP-PC

Para conectar el equipo USRP al PC se debe realizar mediante cable Gigabit Ethernet y cuya **dirección IP** del PC es necesario cambiar, esto se hace en el apartado de Network en configuraciones, allí cambiaremos las opciones de Wired en IPv4 de **Automatic (DHCP)** a **Manual** e ingresaremos una IP que se encuentre dentro de la clase **192.168.10.X**, el equipo USRP viene por defecto con la dirección **192.168.10.2**, un ejemplo de dirección IP sería 192.168.10.12 para la PC, guardamos los cambios y reiniciamos la red, esto debería establecer la comunicación entre la PC y el equipo USRP, para comprobar, en un terminal podemos poner el comando **ping** 192.168.10.2, proporcionará información sobre si la dirección es alcanzable y cuánto tiempo tarda en recibir una respuesta del destino.

The screenshot shows the 'Profile 2' configuration window for IPv4. The 'IPv4 Method' is set to 'Manual'. The 'Addresses' table is as follows:

Address	Netmask	Gateway	
192.168.10.12	255.255.255.0	192.168.10.2	

Figura 16. Asignación de IP estática

Si la instalación de los drivers necesarios se completó utilice el comando *uhd\_find\_devices*, este comando es utilizado para buscar y mostrar los dispositivos conectados al PC y que es parte del kit de desarrollo de hardware (UHD) de Ettus Research instalados anteriormente.

### Resultado de la práctica

Una vez terminado, GNU Radio estará instalado al igual que sus dependencias y controladores UHD para equipos USRP, para ejecutarlo colocamos en una ventana terminal el comando *gnuradio-companion*, lo cual ejecutará el programa.

GNU Radio propone un entorno de trabajo intuitivo y sencillo, en él se encuentran las siguientes ventanas presentadas en la Tabla 13:

Tabla 13. Distribución ventana general de GNU Radio

Ventana	Descripción
1. Barra de herramientas	Contiene las distintas opciones que representan funciones y acciones comunes del software
2. Flujo de trabajo	Área de trabajo en el cual se colocan los bloques y se realizan las conexiones necesarias para la construcción de los sistemas
3. Entorno de bloques	Contiene una lista de todos los bloques disponibles que puedes utilizar en el canvas, estos están organizados en categorías
4. Consola	Proporciona una salida de texto en el cual se muestran mensajes de estado
5. Entorno de variables.	Muestra las variables y sus valores utilizados en la ejecución del programa

La Figura 17 muestra la ventana de inicio del programa GNU Radio

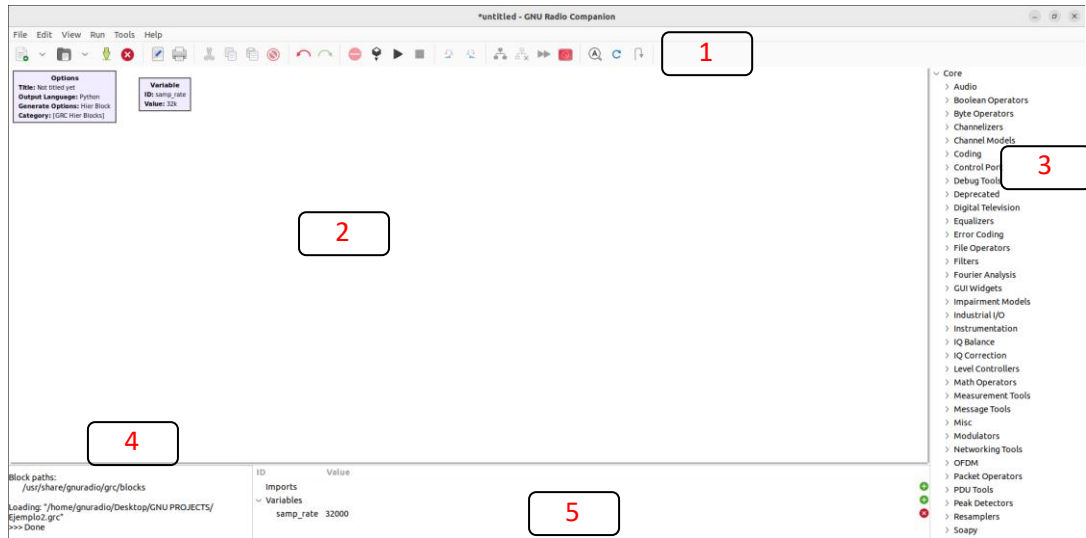


Figura 17. Ventana de inicio de GNU Radio

### 3.2.2 Práctica 2: Análisis de señales

#### Objetivo

En esta práctica el estudiante logrará conocer el entorno de trabajo de GNU Radio, proponiendo un sistema para visualizar distintas señales con sus respuestas en tiempo y frecuencia, mediante uso de bloques de generación de señales, bloques de interfaz de usuario y bloques de control de variables se logrará el control de parámetros básico de la señal como Amplitud y Frecuencia.

#### Materiales y equipos

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio

#### Procedimiento

La Figura 18 muestra el sistema a desarrollar en la práctica.

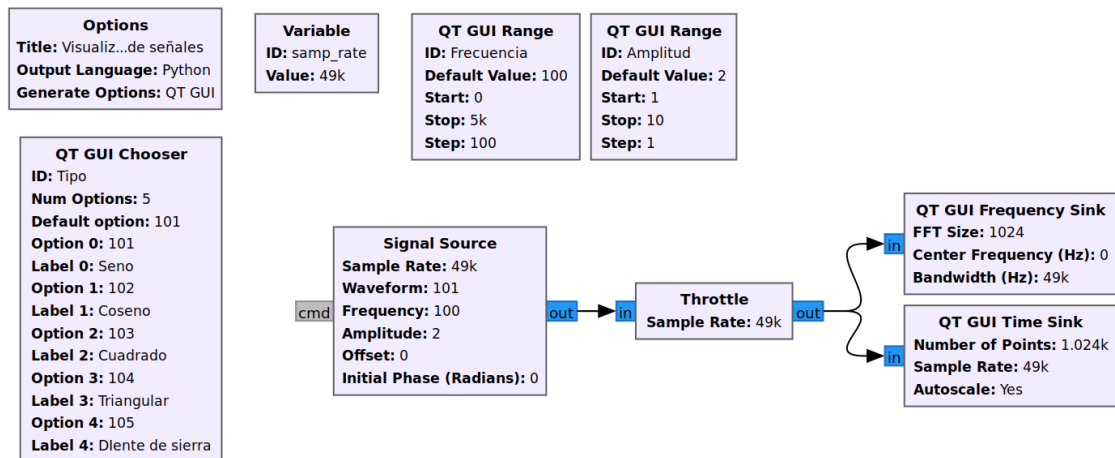


Figura 18. Diagrama de práctica 2

## 1. Generar señales

Las señales son generadas por el bloque Signal Source. La onda generada tendrá las características propuestas en la Tabla 14.

Tabla 14. Parámetros para generador de señales

Parámetro	Valor
Forma de onda	<ul style="list-style-type: none"> <li>- Cosenoidal</li> <li>- Senoidal</li> <li>- Cuadrada</li> <li>- Triangular</li> <li>- Diente de sierra</li> </ul>
Frecuencia	0 a 5000 Hz (5 KHz)
Amplitud	1 a 10

Los parámetros establecidos en la tabla anterior son un ejemplo y pueden cambiar según las necesidades.

## 2. Creación de variables

Establecer un ID a cada bloque de control y variable, de acuerdo con los parámetros que se quieran utilizar, dos bloques QT GUI Range se usaran como deslizadores para variar parámetros de frecuencia y amplitud de la señal.

### 3. Controlar el uso de CPU

Mediante el uso de un bloque Throttle conectado a la salida de del generado de señales se puede limitar la velocidad de procesamiento de las muestras tomadas con el fin de evitar la saturación y exceso de uso en recursos del sistema.

### 4. Configurar bloque para el tipo de señales

El Bloque QT GUI Chooser crea una variable con distintas opciones, pueden colocarse en forma de caja o como botón de selección, útil para la selección de distintos parámetros a establecer.

### 5. Visualizar señales

La visualización de señales se da a través de dos bloques de visualización, uno en el dominio del tiempo (QT GUI TIME SINK) y otro en el dominio de la frecuencia (QT GUI FREQUENCY SINK).

### Resultados de la práctica

A continuación, la Figura 19 y Figura 20 presentan el resultado del sistema propuesto en la Figura 18.

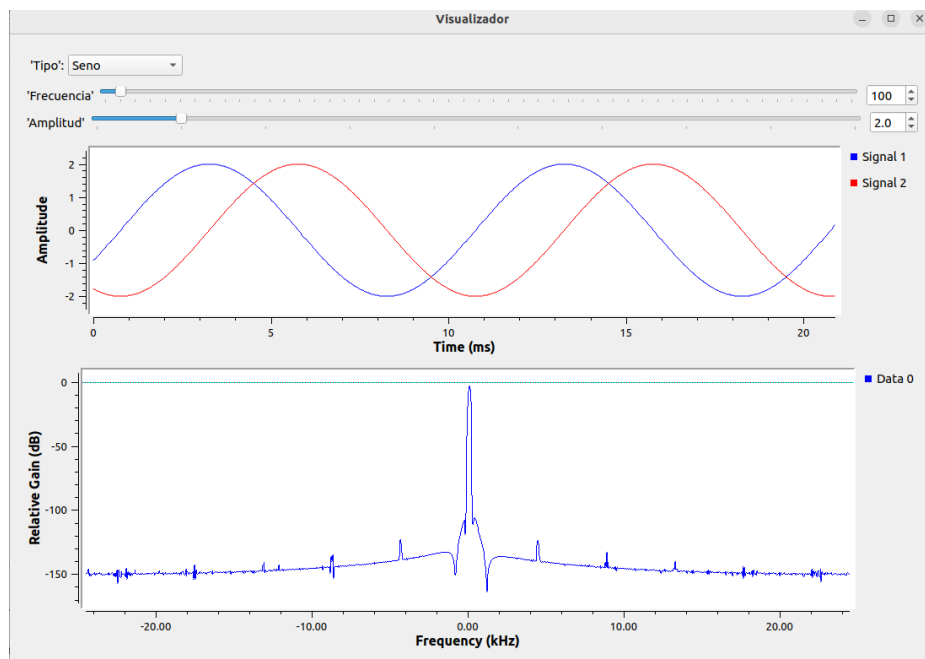


Figura 19. Señal senoidal generada

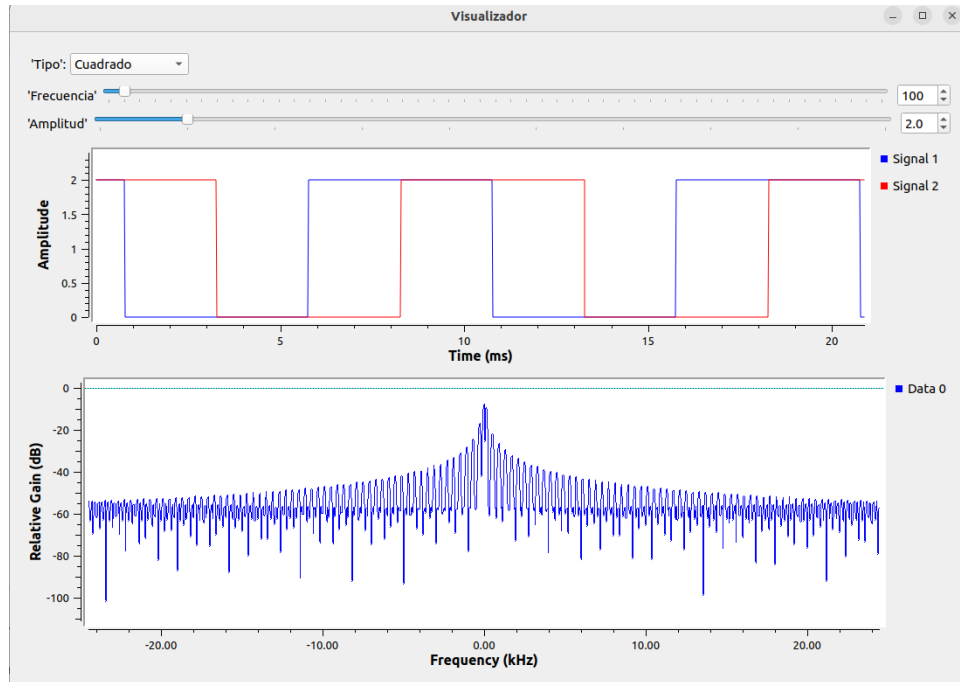


Figura 20. Señal Cuadrada generada

Utilizando bloques de generación de señales y una interfaz de usuario, es posible seleccionar la amplitud y frecuencia de distintos tipos de señales. Como resultado, se puede observar en una gráfica la forma de onda de la señal en el dominio del tiempo, en el cual se puede visualizar su periodo, amplitud y cualquier distorsión o variación en la señal generada.

Los bloques de control y variables permiten que el sistema pueda ser cambiado mientras está en ejecución, el bloque QT GUI Chooser crea un recuadro en la ventana en el cual dependiendo de las variables que contenga puede hacer cambios en los bloques de visualización, de la misma manera el bloque QT GUI Range cambia el valor de los datos que tenga en sus configuraciones, permitiendo cambiar valores de frecuencia, el cual como se presentaba en la tabla puede ir desde los 0 Hz hasta los 5000 Hz, así mismo tiene el control de la amplitud.



### 3.2.3 Práctica 3: Operaciones con señales continuas

#### Objetivo

Esta práctica tiene como objetivo comprender el comportamiento de las señales a las distintas operaciones algebraicas disponibles para señales continuas, mediante un GUI de usuario se visualizará los cambios en el tiempo y frecuencia de la operación seleccionada, de igual forma se usará como entrada una señal discreta analizar las diferencias y cambios con respecto a señales continuas.

#### Materiales y equipos

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio

#### Procedimiento

La Figura 21 muestra el sistema a desarrollar de la práctica.

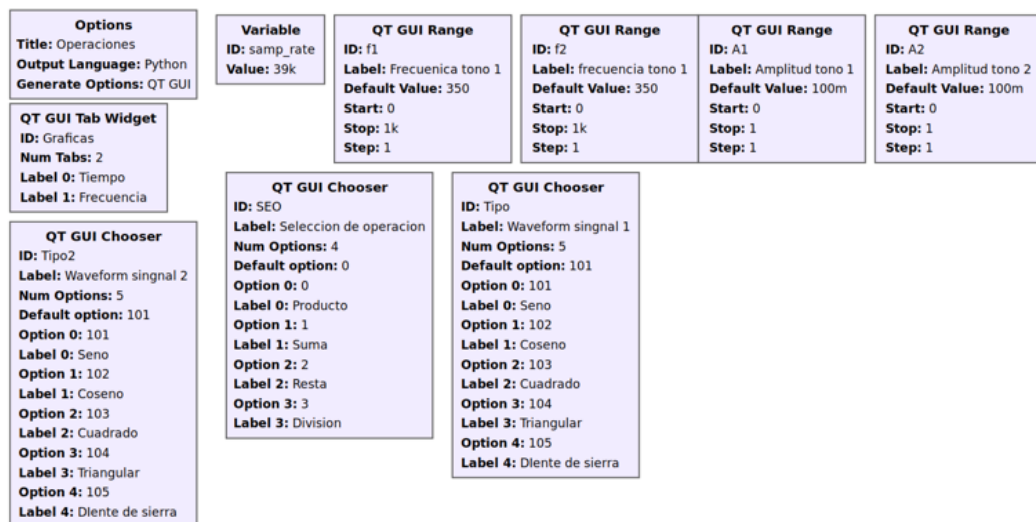


Figura 21. Diagrama de variables práctica 3

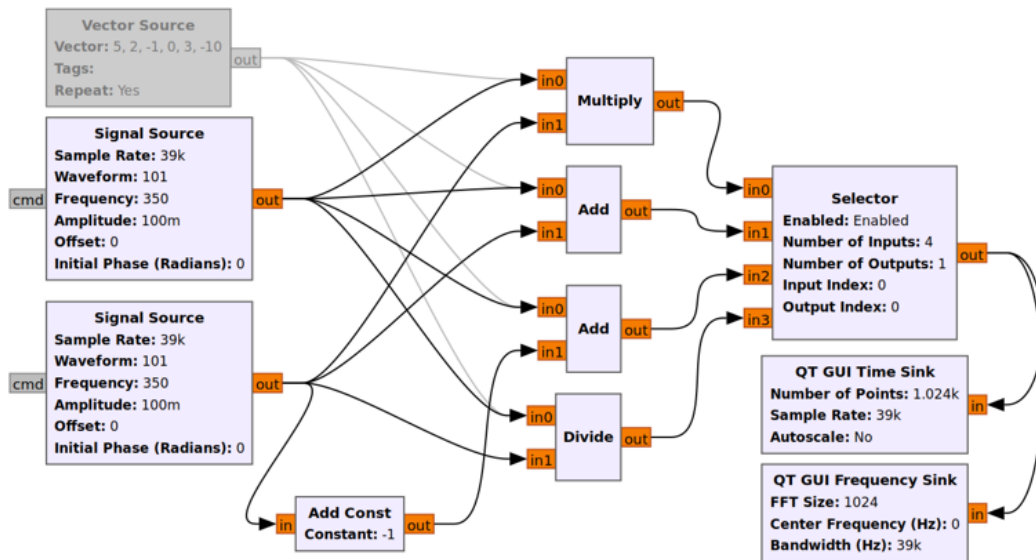


Figura 22. Diagrama de práctica 3

## Primer parte

### 1. Generación de ondas

Las ondas generadas a través de dos bloques Signal Source tendrán las características propuestas en la Tabla 15.

Tabla 15. Parámetros para tomar de bloques Signal Source

Parámetro	Valor
Forma de onda	Cosenoidal Senoidal Cuadrada Triangular Diente de sierra Discreta en puntos (5, 2, -1, 0, 3, -10)
Operaciones	Adición Multiplicación División Sustracción
Frecuencia	0 a 1000 Hz (1 KHz)
Amplitud	0 a 1

La elección del tipo de forma de onda a usar será controlada mediante el bloque QT GUI Chooser.

Para utilizar el bloque Vector Source es necesario desconectar uno de los bloques Signal Source y deshabilitarlo utilizando la opción disponible en la barra de herramientas

## **2. Creación de variables**

las variables permiten controlar los valores de algunos casos, la variable samp\_rate indica la cantidad de muestras a procesar en el momento de la ejecución del sistema.

## **3. Crear el sistema de operaciones**

Cada una de la salida de los bloques generadores de señales se conectarán a los distintos bloques de operaciones disponibles: Add, Multiply y Divide. Cada bloque consta de dos entradas, por lo que para cada entrada del bloque será el de la salida de los bloques Signal Source.

## **4. Seleccionar el tipo de operación**

A cada una de las salidas de los bloques de operaciones se conectará un bloque Selector, este bloque se encargará de la elección de las operaciones que se quiera realizar.

## **5. Configurar bloques de control**

Los bloques de control QT GUI Range, QT GUI Chooser y QT GUI Tab Widget se encargarán de los cambios en las gráficas y la distribución de las opciones y ventanas en el GUI, configurar sus parámetros de acuerdo con los establecidos en el sistema. Los valores de los bloques QT GUI Range pueden cambiar de acuerdo con lo que se desee, estos vienen siendo valores de ejemplos.

## **6. Visualizar las señales**

Para visualizar las señales resultantes del sistema conectar dos bloques de visualización en el dominio de tiempo y de frecuencia QT GUI Time Sink Y QT GUI Frequency Sink, respectivamente.

## Resultados de la práctica

La Figura 23 y Figura 24 muestran las gráficas obtenidas después de diseñar el sistema de acuerdo con la Figura 22.

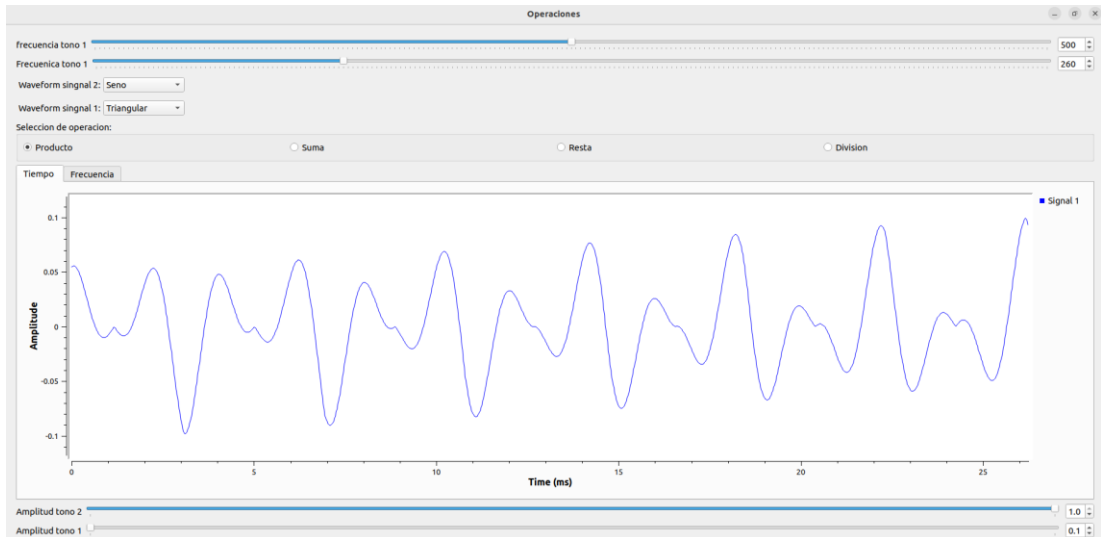


Figura 23. Interfaz y operación entre dos señales

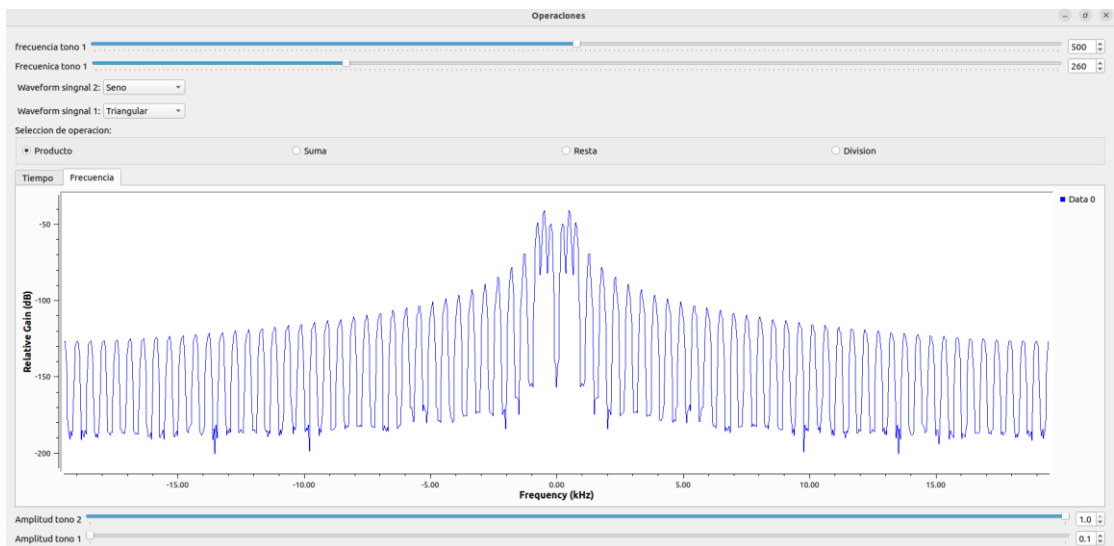


Figura 24. Respuesta en frecuencia de operación entre señales

## Segunda parte

1. Generar un bloque de vectores

Haciendo uso de un bloque Vector Source, desconectar uno de los bloques Signal Source y deshabilitarlo para conectar este nuevo bloque con los parámetros establecidos anteriormente.

## Resultados

La Figura 25 y Figura 26 muestran las gráficas obtenidas entre las operaciones con un bloque de señales continuas y un bloque de puntos.

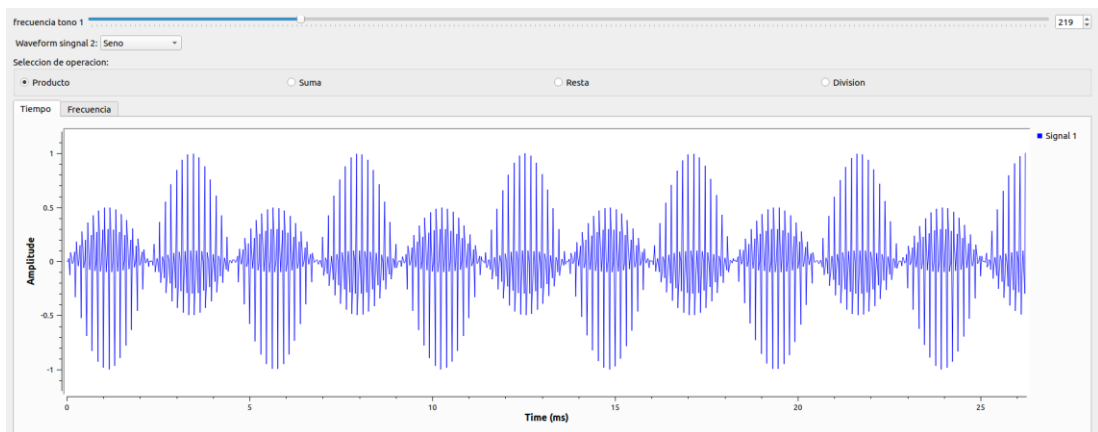


Figura 25. Operación entre bloques Signal Source y Vector Source

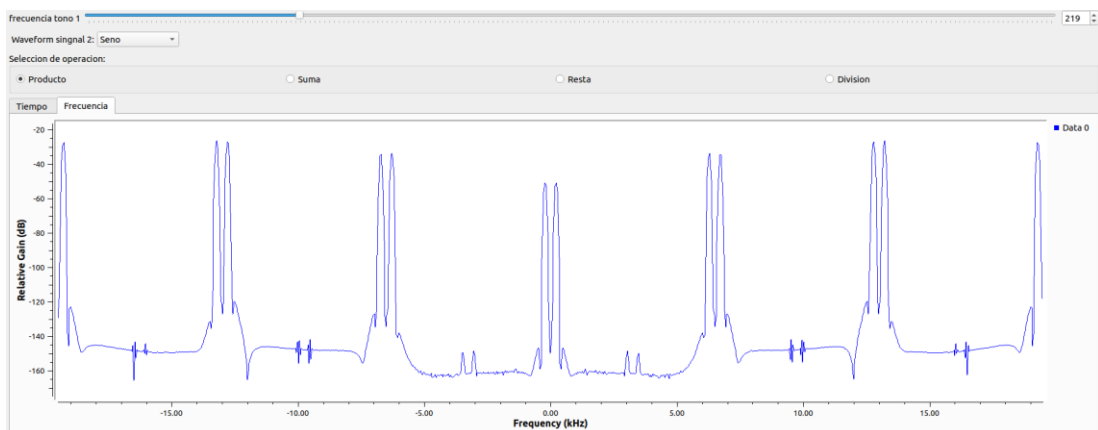


Figura 26. Respuesta en Frecuencia de operación

La práctica resultó en el desarrollo de una interfaz gráfica de usuario que permite seleccionar el tipo de onda de las señales de entrada, el tipo de operación a realizar y controlar parámetros como la amplitud y frecuencia. Esta GUI facilita la interacción con el sistema y brinda una forma intuitiva de controlar y visualizar los resultados de las operaciones con señales continuas. Además, proporciona una herramienta de análisis rápida y práctica para los investigadores y estudiantes.

### 3.2.4 Práctica 4: Operaciones con señales singulares

#### Objetivo

Esta práctica tiene como objetivo comprender el comportamiento de las señales singulares a las distintas operaciones algebraicas disponibles y comprender el funcionamiento de bloques Python para crear bloques personalizados acorde a las características requeridas para el sistema.

En realidad, no existe una forma directa de generar señales matemáticas singulares usando bloques estándar en GNU Radio Companion (GRC), ya que GRC se enfoca en el procesamiento de señales en tiempo real y no está diseñado para trabajar con funciones matemáticas teóricas con singularidades.

Por lo que se puede utilizar el bloque *Python Block* en GNU Radio para implementar un bloque propio personalizado en Python.

#### Materiales y equipos

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio

#### Procedimiento

La Figura 27 muestra el sistema a desarrollar de la práctica.

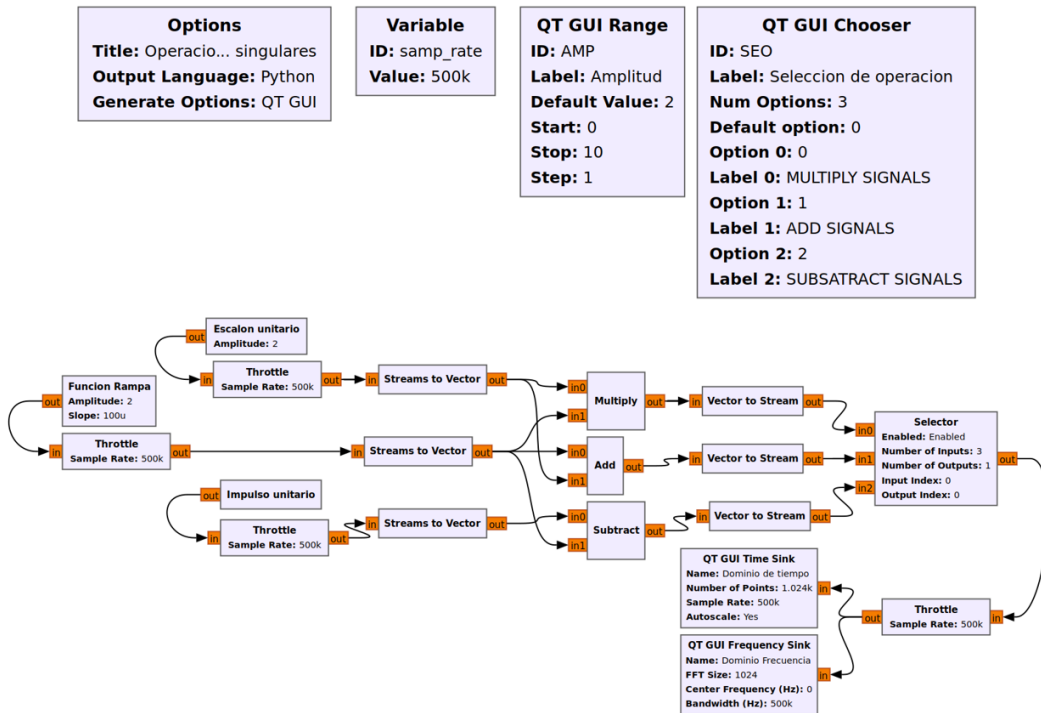


Figura 27. Diagrama de práctica 4

## 1. Generar señales

Las señales se generan a través de bloques de código personalizado Embedded Python Block. Mediante un editor de código disponible en el sistema es posible crear bloques con características específicas al sistema necesario.

Crear 3 bloques con códigos específicos para ejecutar señales Escalón Unitario, Rampa Unitaria e Impulso unitario.

## 2. Creación de variables de control

Tener definido las variables que se vayan a usar ya que los valores que se utilicen se usaran en todos los bloques a los que sean llamados. La variable Samp\_rate, indica la cantidad de muestras que tomara el sistema.

## 3. Controlar y almacenar tipo de datos

Una vez creado los bloques se conectarán a bloques Throttle para limitar el uso de recursos, y sus salidas conectadas a los bloques Stream to Vector, que toma los datos de la secuencia de elementos de entrada y los almacena en un vector.

#### 4. Crear el sistema de operaciones

Su salida a los distintos bloques de operaciones disponibles Multiply, Add y Subtract.

#### 5. Configurar bloques de control

El resultado de cada una de las operaciones va dirigido al bloque Selector el cual mediante la configuración de sus parámetros se controlará la elección del tipo de operación que se quiera realizar.

#### 6. Visualizar señales

Colocar y conectar bloques de visualización de señales para comprobar los resultados de las operaciones.

### Resultados de práctica

La Figura 28 muestra las gráficas obtenidas de acuerdo con el diagrama de conexión que se muestra en la Figura 27.

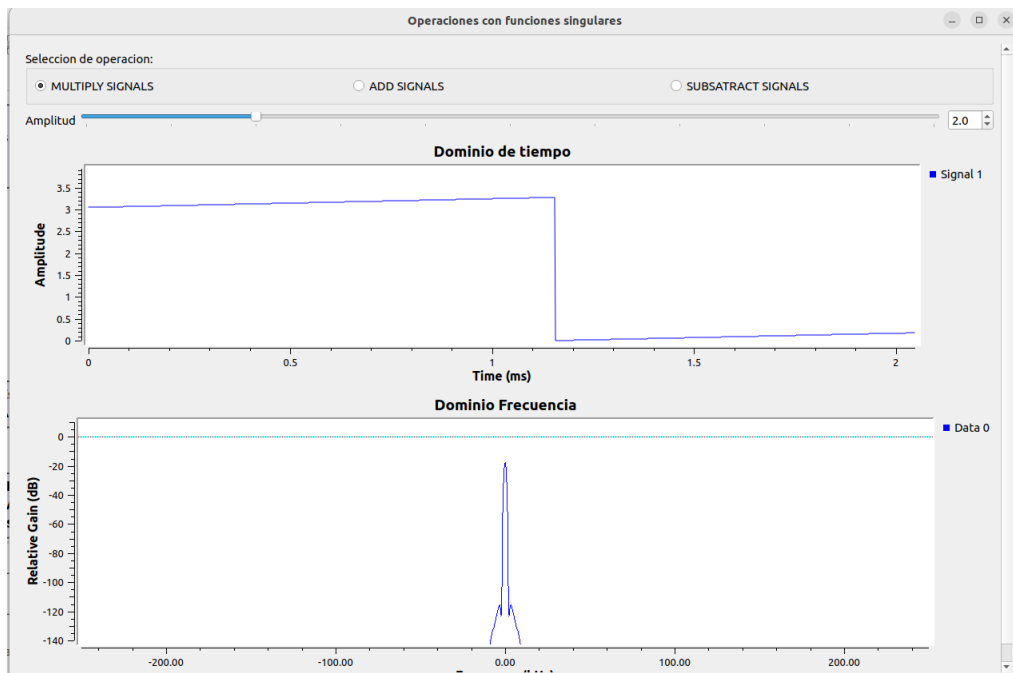


Figura 28. Interfaz y operación entre bloques personalizados

La ventana presenta las gráficas de las operaciones realizadas entre las señales obtenidas a través de los bloques Embedded Python Block, los cuales son bloques específicos para la generación de señales singulares en GNU RADIO. Utilizando el



bloque embebido de Python disponible en la interfaz de bloques y un editor de texto se puede implementar código personalizado, con lo cual se consigue general señales aproximadas de funciones singulares mediante programación. Los parámetros que obtiene el bloque personalizado son establecidos dentro del código a ejecutar, es en tal caso el lugar donde se debe modificar si se requiere aspectos específicos.

### **3.2.5 Práctica 5: Osciladores de radiofrecuencia**

#### **Objetivo**

La siguiente práctica tiene como objetivo comprender el comportamiento de los osciladores de radiofrecuencia, para ello se diseñará un sistema el cual ayude a comprender y observar el comportamiento de las distintas señales generadas.

GNU Radio ofrece el bloque Python Block, con el cual se generará el código pertinente a un oscilador de radiofrecuencia para señales AM, FM y de radioafición.

#### **Materiales y equipos**

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio

#### **Procedimiento**

La Figura 29 muestra el sistema a desarrollar de la práctica.

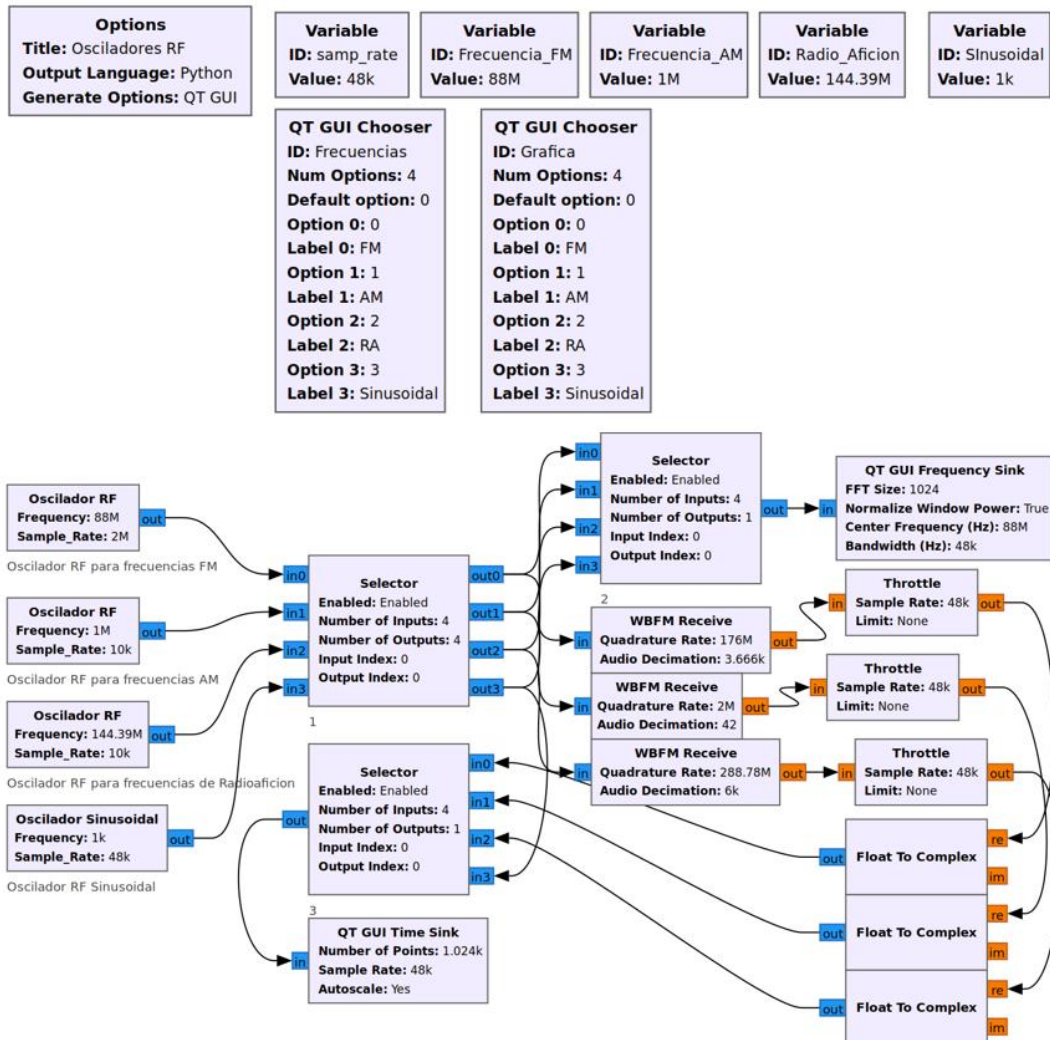


Figura 29. Diagrama de práctica 5

## 1. Generar señales

Mediante el bloque Embedded Python Block personalizar el código para crear 3 bloques de oscilación para frecuencias FM, AM, Radioafición y una frecuencia de señal sinusoidal. Los parámetros que tomen estos bloques dependerán de las frecuencias requeridas para los sistemas FM, AM y de radioafición.

## 2. Crear variables para el control

Las variables controlaran los valores de las frecuencias de los osciladores, esto es útil para tener un control de valores y poder cambiarlos si es necesario, caso contrario si se desea modificar estos valores debería hacerse desde dentro del código de cada uno de los bloques personalizados.

### **3. Configurara bloque de control**

Mediante bloque Selector configurar sus para metros para controlar la elección del tipo de oscilador a escoger. De la misma manera estos bloques servirán para elegir la gráfica que se desea observar de los osciladores generados.

### **4. Reducir tasa de muestreo**

Configurar el bloque WBFM, esto ayudará a reducir la tasa de muestreo de la señal recibida por los bloques osciladores, mediante la decimación de audio se controlará la reducción de las muestras, obteniendo señales más manejables y adecuadas para la reproducción o procesamiento.

### **5. Control de datos**

La salida de cada uno de estos a un bloque Throttle y bloques conversores de datos float a complejos para su visualización.

### **6. Visualizar señales**

Configurar un segundo bloque selector encargado de seleccionar la gráfica de que señal queremos observar, estos parámetros de configuración son controlados por bloques QT GUI Chooser, los cuales contienen las variables necesarias para el control de las ventanas graficas. Y su salida a un bloque de visualización para observar el comportamiento de las señales.

### **Resultados de la práctica**

La Figura 30 y Figura 31 muestran las gráficas obtenidas de acuerdo con la conexión establecida en el diagrama de la Figura 29.

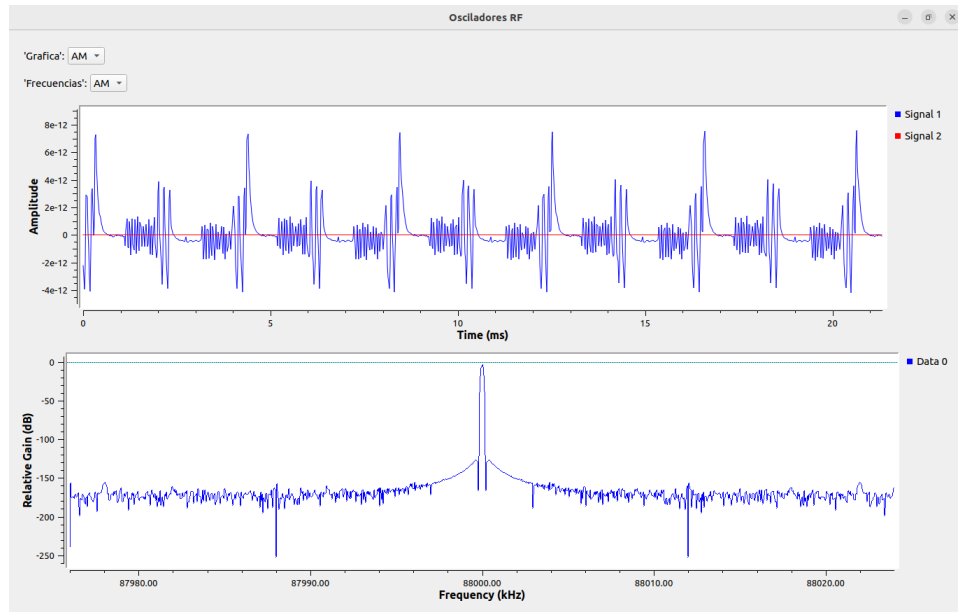


Figura 30. Interfaz Señal de oscilador AM

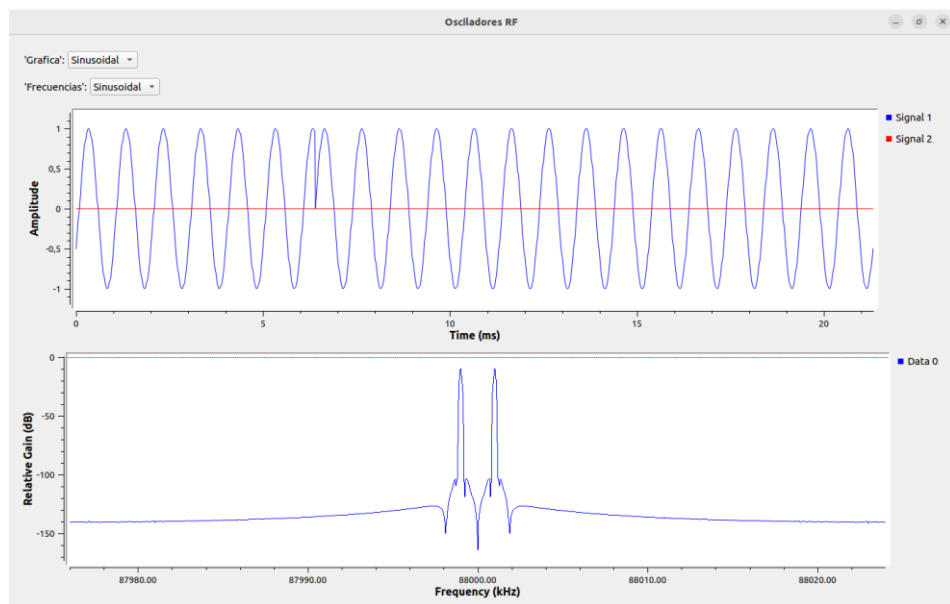


Figura 31. Grafica en tiempo y frecuencia del oscilador Sinusoidal

La adaptación que ofrece GNU Radio para utilizar bloques personalizados para generar sistemas ayuda a crear osciladores a frecuencias específicas y de onda sinusoidal. Esto permite generar señales de audio controladas. Lo cual es útil para la generación de tonos de prueba, simulaciones de sistemas de audio y pruebas de hardware de audio teniendo la capacidad de ajustar y personalizar señales para distintos sistemas.

Los osciladores de onda sinusoidal son fundamentales en la simulación de sistemas de comunicación. Al generar señales sinusoidales con frecuencias y tasas de muestreo específicas, se puede simular transmisión, modulación y demodulación de señales de RF. Esto permite evaluar sistemas de comunicación antes de implementarlos en hardware real.

### **3.2.6 Práctica 6: Filtros**

#### **Objetivo**

La siguiente práctica tiene como objetivo comprender los distintos filtros disponibles en el software GNU Radio y su aplicación con señales de audio, para ello se diseñará un sistema el cual permita filtrar frecuencias en específico para observar el comportamiento de la señal en un visualizador.

#### **Materiales y equipos**

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio
- Archivo de audio .wav

#### **Procedimiento**

La Figura 32 muestra el diagrama del sistema a desarrollar.

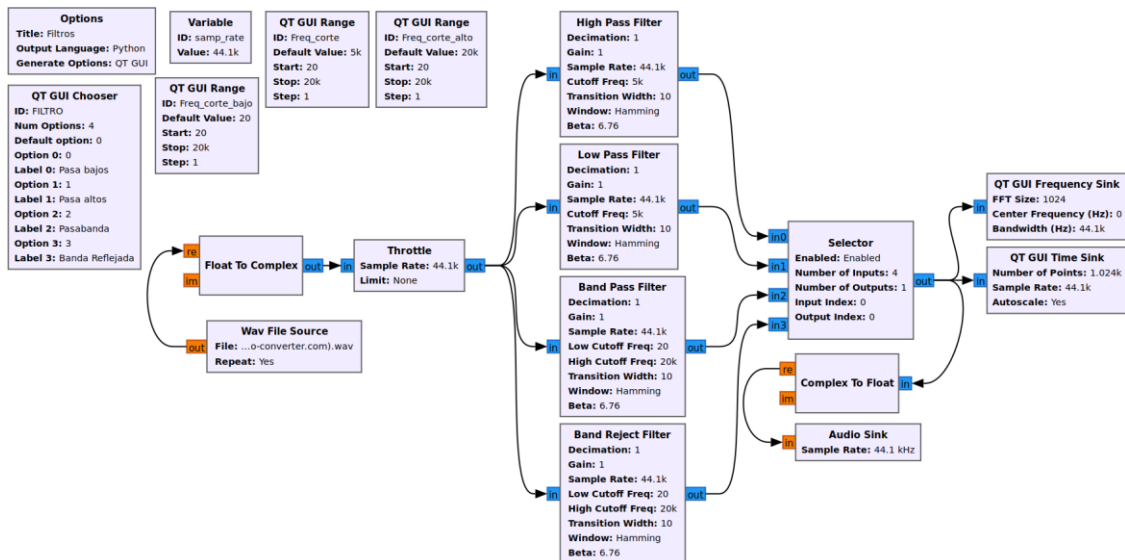


Figura 32. Diagrama de práctica 6

### 1. Generar la señal

La fuente en esta práctica viene procedida desde un archivo de audio WAV. Haciendo uso de un bloque WAV File Source, se podrá cargar archivos con extensión .WAV, lo cual servirá como base para los distintos filtros a aplicar.

### 2. Establecer las variables del sistema

Determinar las variables que se utilizaran para el control de los bloques de filtros, entre ellos se encuentran las variables de toma de muestras y los distintos valores para las frecuencias de corte específicas para cada filtro.

### 3. Controlar los datos

Convertir los datos flotantes procedentes del audio a usar, mediante el uso del bloque Float To Complex, y a su salida un bloque Throttle para limitar el uso de CPU

### 4. Crear el sistema de filtros

Conectar a las entradas de los distintos bloques de filtros con los que GNU Radio dispone, estos son los filtros activa alto, pasa bajo, rechaza banda, pasa banda. Los valores de los bloques de filtros están establecidos y controlados por los bloques QT GUI Range. La configuración de sus parámetros dependerá de los requisitos del usuario, se presenta en este trabajo valores como ejemplos para la práctica.

## 5. Configurar bloque de control

Conectar y configurar un bloque Selector para controlar el tipo de filtro a usar en el audio, el parámetro de control será dispuesto por el bloque QT GUI Chooser.

## 6. Visualizar señales

Mediante un bloque Audio Sink conectado a la salida del bloque Selector se podrá escuchar los cambios que los filtros realizan al audio original, y mediante los bloques de visualización los cambios en la frecuencia del archivo.

## Resultados de la práctica

La Figura 33 muestra las gráficas resultantes de la conexión realizada de acuerdo con el diagrama establecido en la Figura 32.

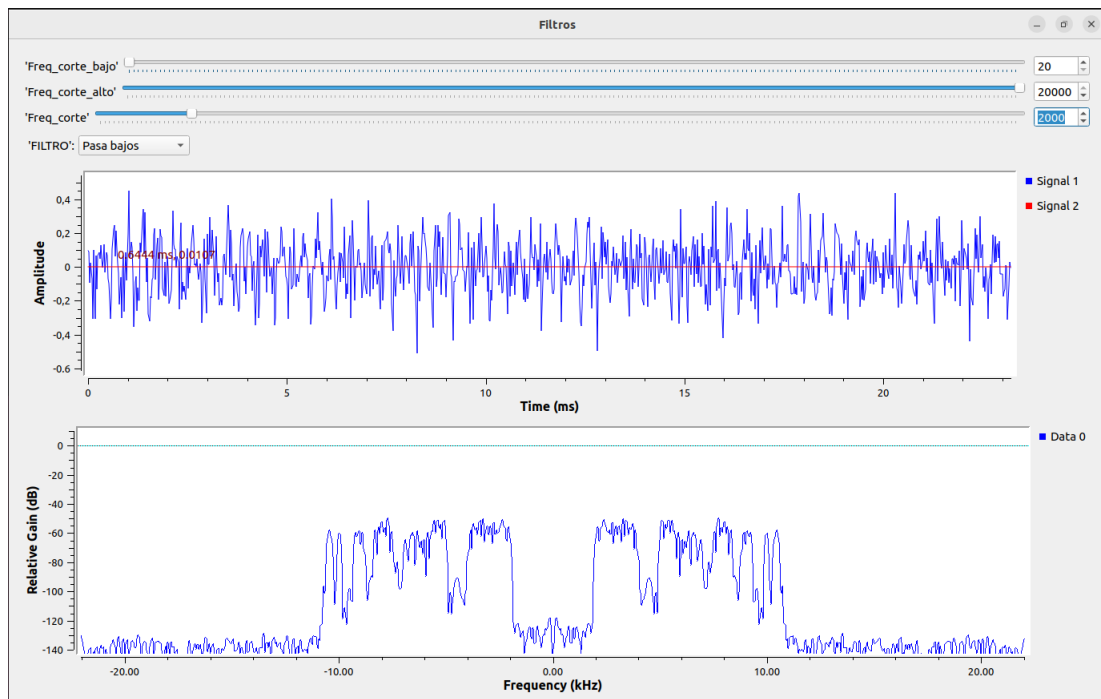


Figura 33. Gráficas de señal de audio ante filtro pasa bajos

Para el filtro pasa bajos según los parámetros se estableció una frecuencia de corte de 5 KHz, es decir frecuencias menores a aquel valor podrán reproducirse y mostrarse, mientras que los valores que superen el valor serán atenuados.

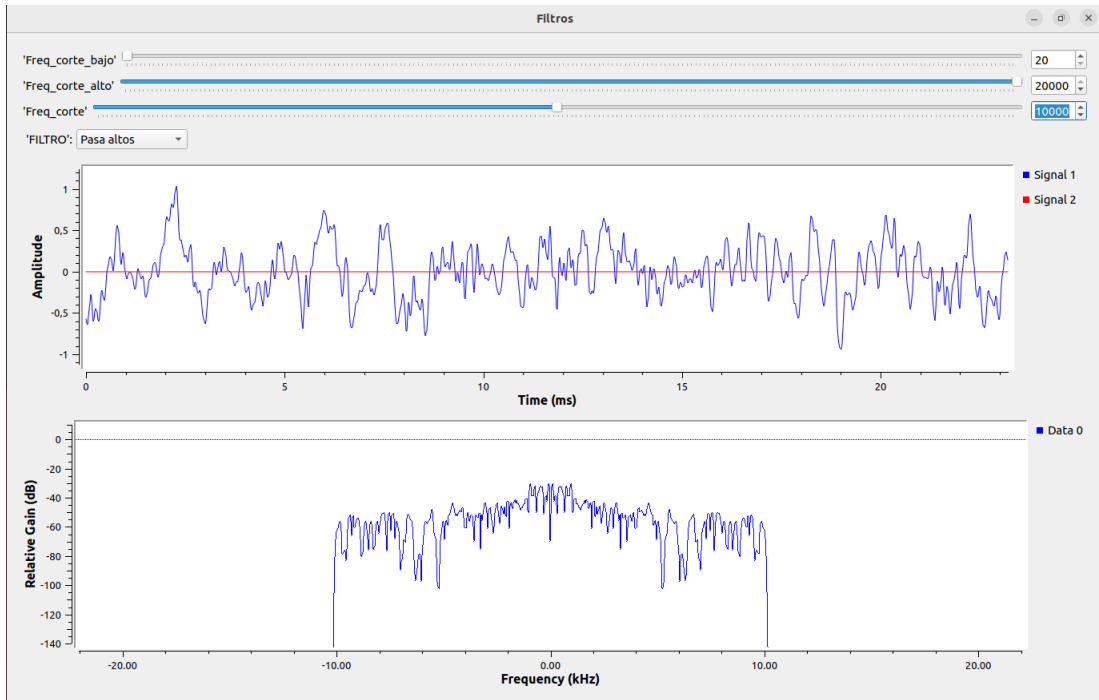


Figura 34. Gráficas de señales de audio ante filtro pasa altos

El mismo archivo de audio ahora con un filtro pasa altos, la frecuencia de corte se encuentra en el mismo valor de 5 KHz, por lo que valores menores a esta frecuencia de corte serán atenuadas.

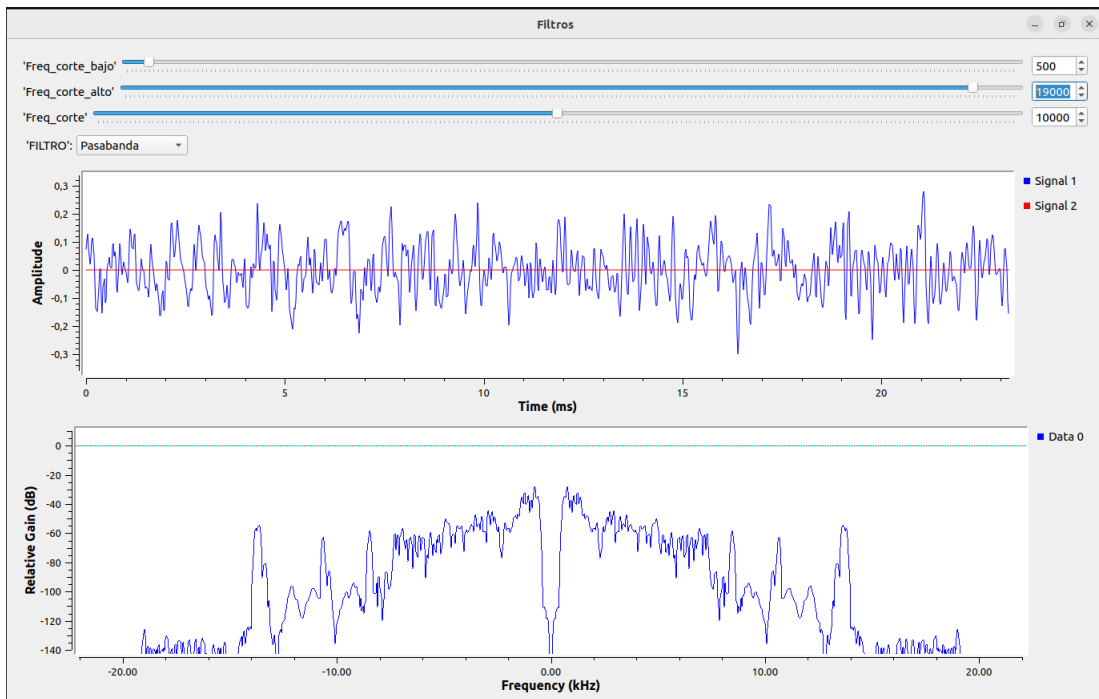


Figura 35. Gráficas de señal de audio ante filtro pasa banda



El filtro pasa banda permite pasar frecuencias que se encuentren dentro de un rango de valores establecidos, los valores establecidos anteriormente son frecuencia de corte bajo de 20 Hz y frecuencia de corte alto de 20 KHz.

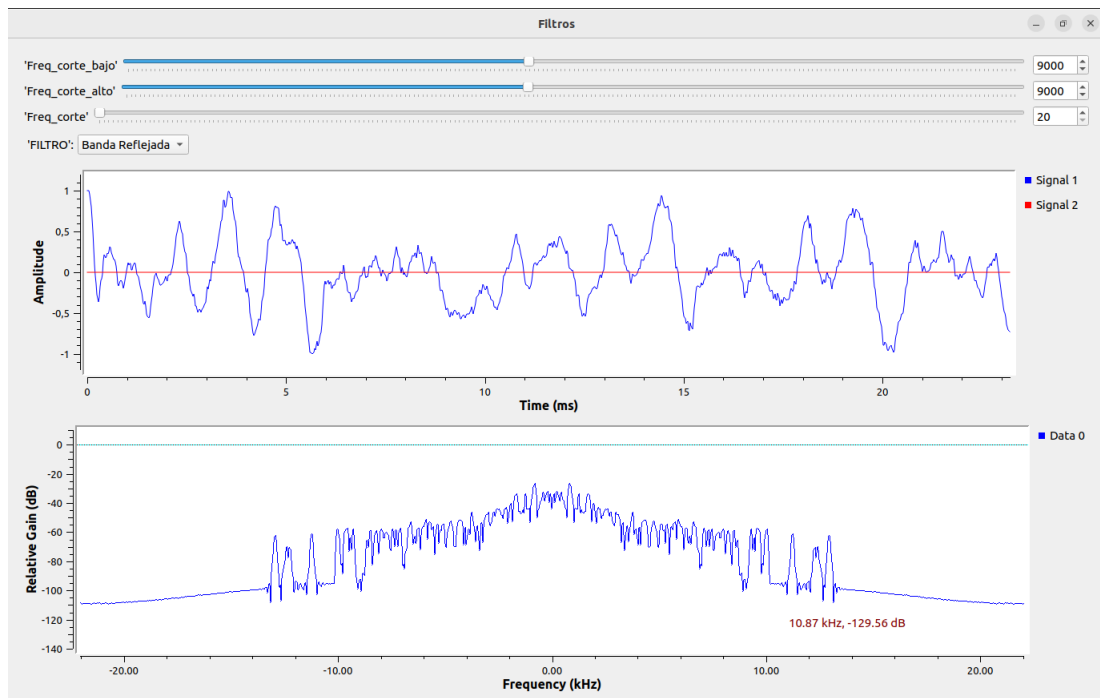


Figura 36. Gráficas de señal de audio ante filtro de banda relegada

El filtro rechaza banda atenúa los valores que se encuentren dentro de un rango de frecuencia, los valores son los mismos que se usaron en el filtro pasa banda.

Mediante el uso de los visualizadores se puede analizar la respuesta en frecuencia de los distintos filtros, esto implica la indagación de como el filtro afecta a las distintas frecuencias que se pueden encontrar en un archivo de audio o cualquier señal a recibir.

La implementación de un sistema filtros ayuda a comparar los efectos causados sobre una señal permitiendo entender el uso de cada filtro. Al ajustar las frecuencias de corte y otros parámetros del filtro, se puede observar cómo varía la atenuación o ganancia en diferentes partes del espectro de frecuencia. Esto proporcionaría información sobre cómo el filtro modifica selectivamente las frecuencias.

### 3.2.7 Práctica 7: Modulador AM

#### Objetivo

Esta práctica tiene como objetivo la comprensión de la modulación por amplitud en sistemas de comunicación y su comportamiento al cambio en frecuencias. A través de esta práctica, los estudiantes tendrán la oportunidad de configurar un sistema de modulación AM, explorar sus componentes clave y analizar la respuesta del sistema ante variaciones en la amplitud y frecuencia de la señal moduladora.

#### Materiales y equipos

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio
- Archivo de audio .wav

#### Procedimiento

La Figura 37 muestra el diagrama del sistema a desarrollar.

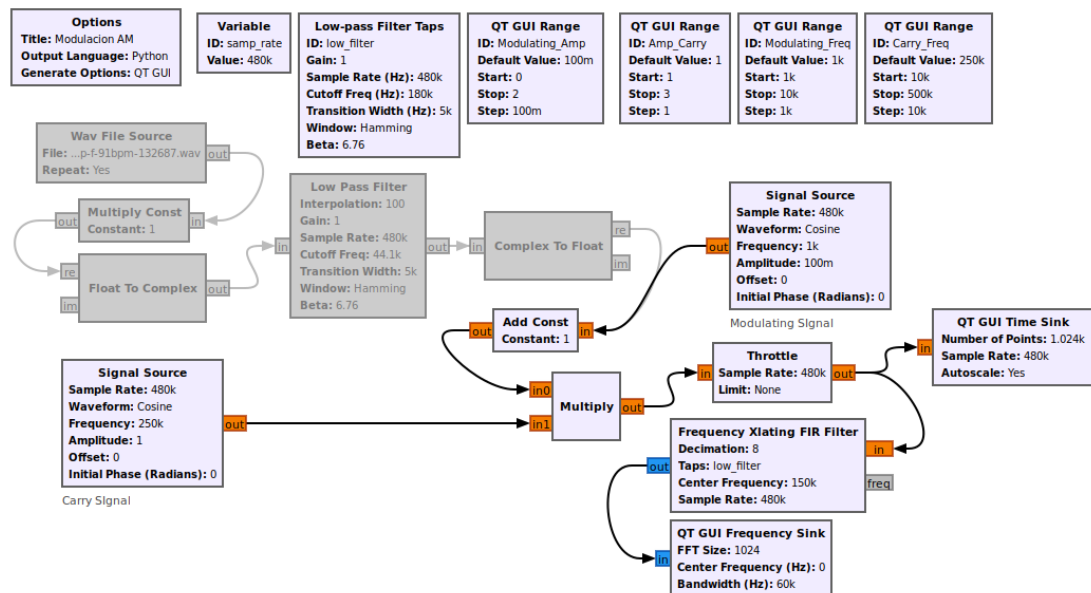


Figura 37. Diagrama de práctica 7

## 1. Generar la señal

Para realizar la modulación AM es necesario obtener una señal que trabaje como portadora y otra como señal carry. La señal carry será generada por un bloque Signal Source, mientras que la portadora o señal a modular será tomada de un bloque generado de señales igual o mediante un archivo de audio usando un bloque Wav File Source.

## 2. Seleccionar el tipo de señal

Si se usa un archivo de audio los datos deben ser convertidos a complejos para su tratamiento, de la misma manera usar un filtro pasa bajos ayudará a eliminar cualquier señal no deseada, en un sistema de simulación como este el filtro no es tan requerido, pero si se deseara reemplazar aquel bloque por un bloque de recepción de señales USRP, entonces si es necesario el uso de filtros en el sistema. Para ello es necesario desconectar la fuente de portadora generada por el bloque Signal Source y realizar el cambio como se muestra en el la Figura 38.

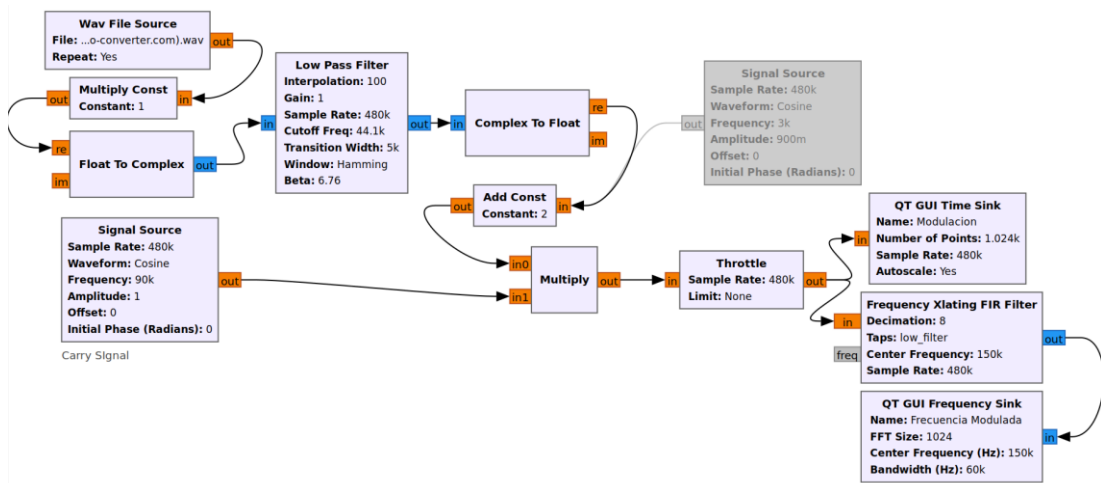


Figura 38. Sistema de modulación AM con archivo de audio

Si se usa un bloque generado de señales el diagrama queda de la manera mostrada en la Figura 37, para una señal de portadora deberá conectarse a un bloque Add Constant, este bloque realizará la operación correspondiente para modificar la amplitud de la señal portadora, con lo cual servirá como índice de modulación.

### **3. Crear las variables de control necesarias**

Las variables necesarias para el sistema emplean bloques de control para la variación de parámetros como la amplitud y la frecuencia de las señales portadora y carry.

Algo necesario a tomar en cuenta es que la frecuencia de la señal moduladora o carry debe ser mucho menor que la frecuencia de la señal portadora para cumplir con el teorema de modulación de banda angosta.

### **4. Diseñar el modulador**

La conexión mediante un bloque Multiply operara ambas señales para obtener la modulación respectiva, su salida a un bloque Throttle para el control excesivo de CPU.

### **5. Reajustar la señal**

Un último bloque de filtrado ayudará a controlar las señales resultantes mediante el desplazamiento de las frecuencias, aplicando una decimación de muestras para poder apreciar la gráfica en el dominio de las frecuencias.

### **6. Visualizar la señal**

Conectar los respectivos bloques de visualización necesarios para comprobar el funcionamiento.

### **Resultados de la práctica**

La Figura 39 muestra las señales resultantes después de realizar la modulación tanto con dos fuentes generadas como con un archivo de audio, la conexión se establece de acuerdo con el diagrama de la Figura 37.

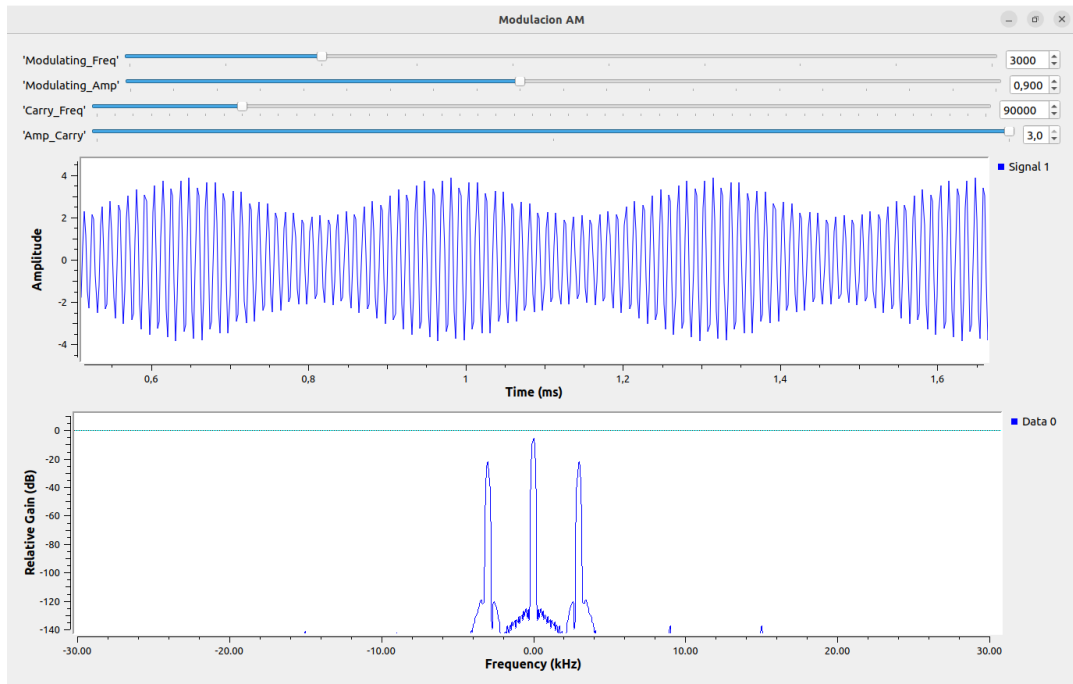


Figura 39. Modulación AM con  $m=50\%$

Para una modulación con  $m=50\%$ , indica que el valor de la amplitud de la portadora varía un 50% en exceso y muy por debajo de la amplitud que tenía originalmente.

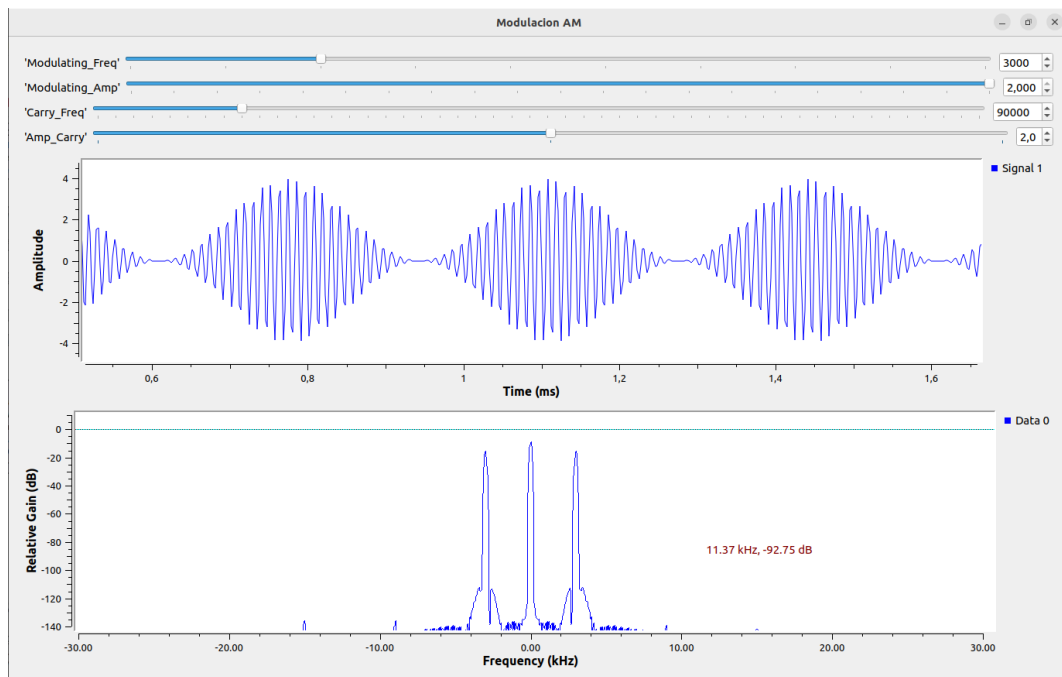


Figura 40. Modulación AM con  $m=100\%$

Indica que  $m = 1$ , el cual corresponde al máximo nivel (100%) permisible de modulación

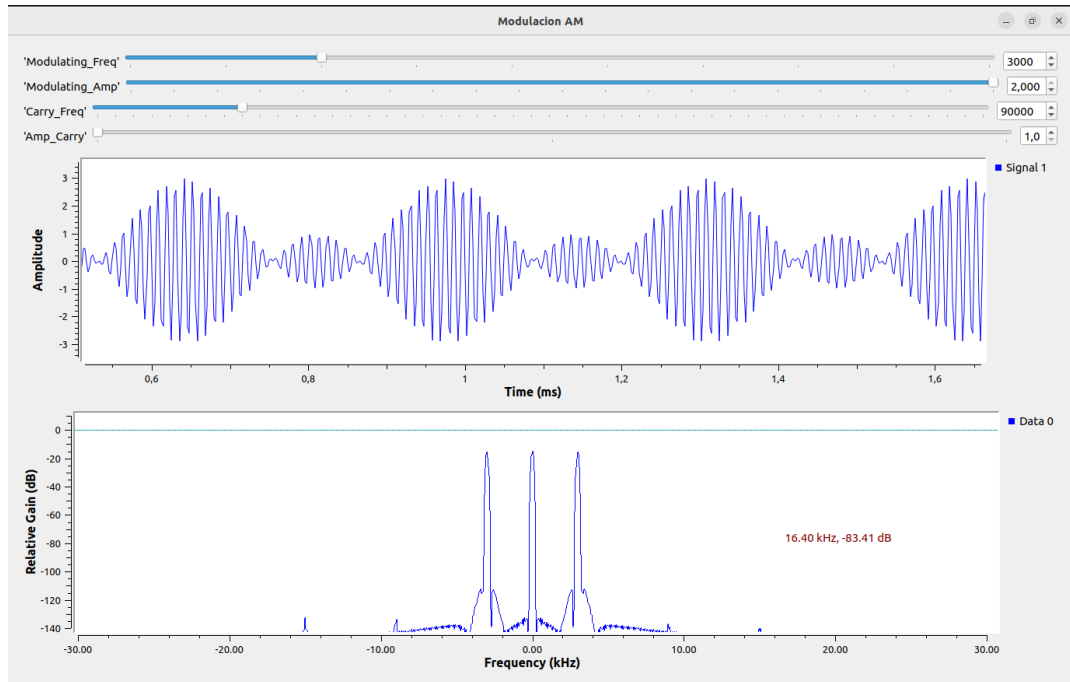


Figura 41. Modulación AM con  $m=150\%$

Por encima de  $m=100\%$  se encuentra la sobremodulación, el cual indica una distorsión en la señal cambiando los valores que le corresponden a la envolvente de su valor original de amplitud.

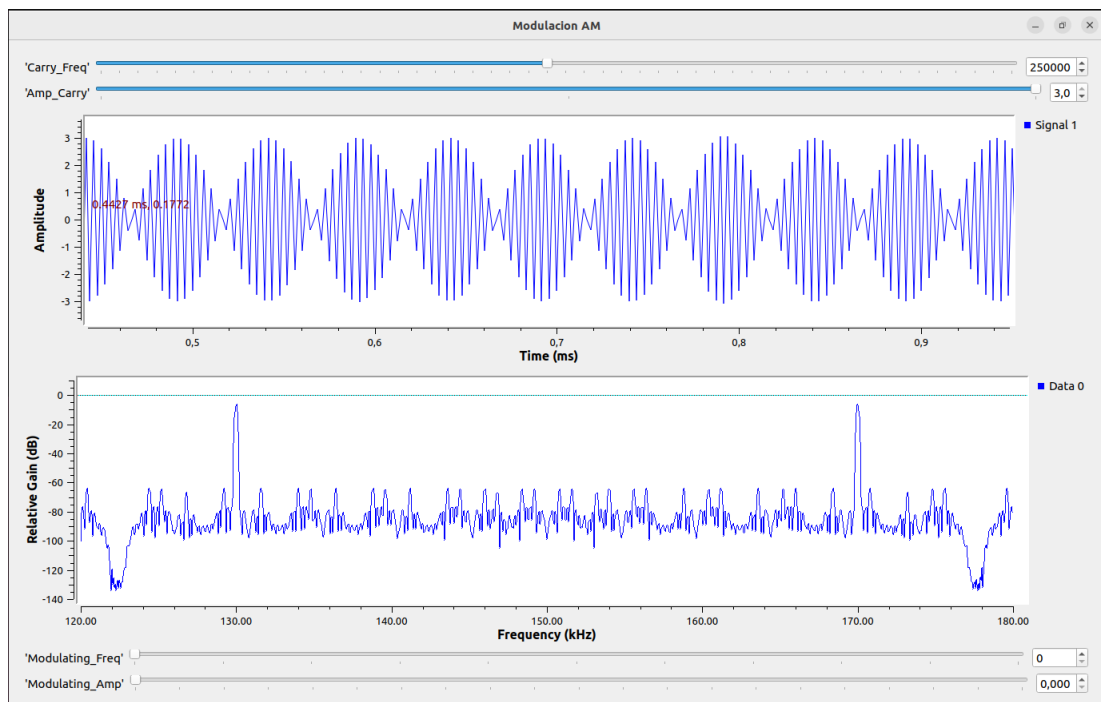


Figura 42. Modulación AM de un archivo de audio

Después de realizar la práctica de modulación de amplitud (AM), se observan diversos resultados significativos. Mediante la visualización de la señal modulada con bloques QT GUI Frequency Sink o QT GUI Time Sink, se puede apreciar claramente cómo la amplitud de la señal portadora se modificaba de acuerdo con la señal moduladora. Esta variación en la amplitud reflejó el proceso de modulación de la señal portadora por la información contenida en la señal moduladora.

### **3.2.8 Práctica 8: Demodulador AM**

#### **Objetivo**

Esta práctica tiene como objetivo la comprensión de la demodulación de señales AM. Para ejemplificar la demodulación AM, se hará uso del sistema realizado en la Práctica de laboratorio 7 Modulador AM, en el cual se agregará los bloques necesarios para la demodulación.

#### **Materiales y equipos**

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio
- Archivo de audio .wav

#### **Procedimiento**

La Figura 43 muestra el diagrama del sistema a agregar.

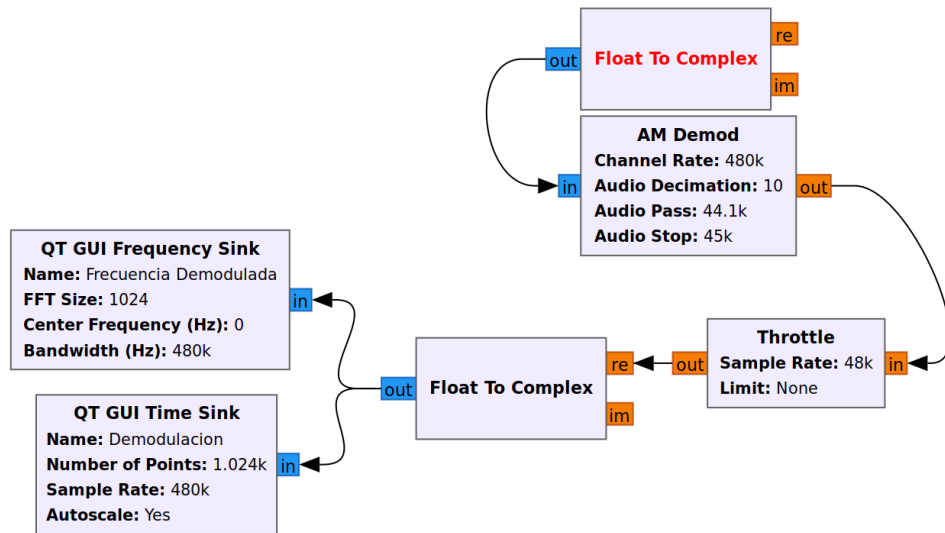


Figura 43. Bloques para demodular una señal AM

Esta parte forma como complemento al diseño realizado en la anterior práctica, por lo que la conexión completa quedaría como se muestra en la Figura 44

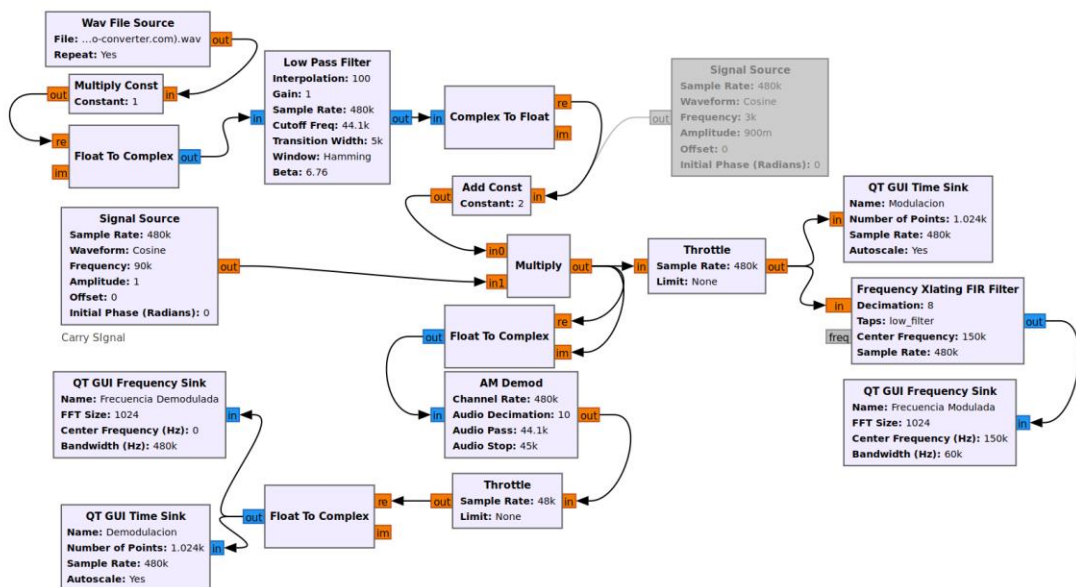


Figura 44. Sistema Modulador y Demodulador AM

## 1. Convertir los datos

Conectar la salida de la señal modulada con anterioridad a un bloque Float to Complex, ya que el bloque que se encarga de la demodulación acepta solo datos complejos.



## 2. Conectar el bloque demodulador

Configurar y conectar el bloque AM Demod, el cual a través de los parámetros disponibles del bloque se puede recuperar la señal. El radio de las muestras tomadas debe coincidir con la frecuencia de muestreo con la que se realizó la modulación. La decimación del bloque permite reducir la cantidad de muestras que procesara a la salida del bloque y eventualmente mediante un filtro integrado en el bloque determinar las frecuencias que podrán pasar por el bloque.

## 3. Visualizar la señal

Volver a convertir los datos si es necesario, y conectar a la salida los bloques de visualización para comprobar los resultados.

## Resultados de la práctica

La Figura 45 muestra las señales moduladas y demoduladas de acuerdo con la conexión que indica el diagrama de la Figura 44.

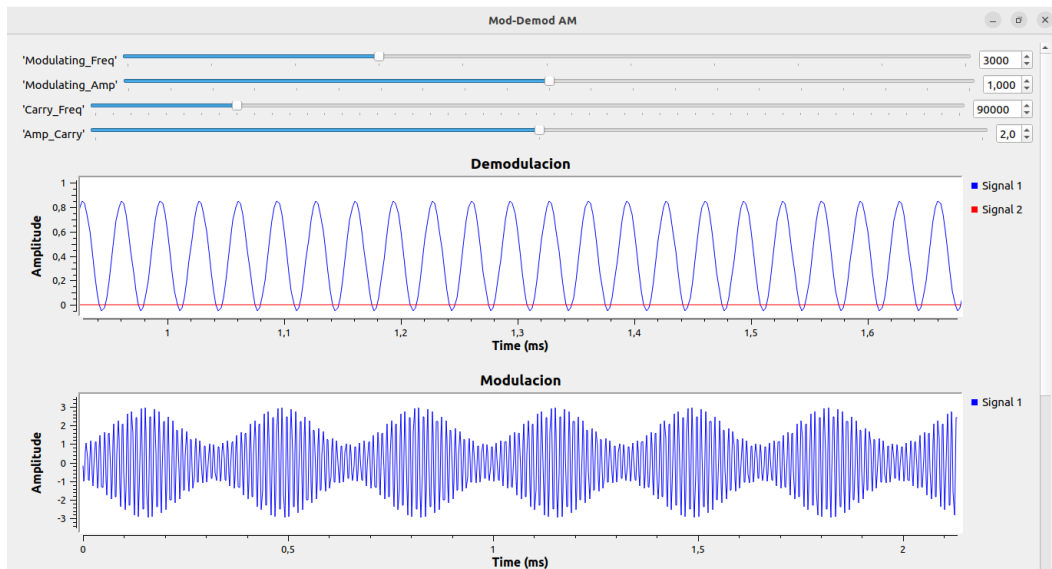


Figura 45. Modulación y demodulación de señal AM

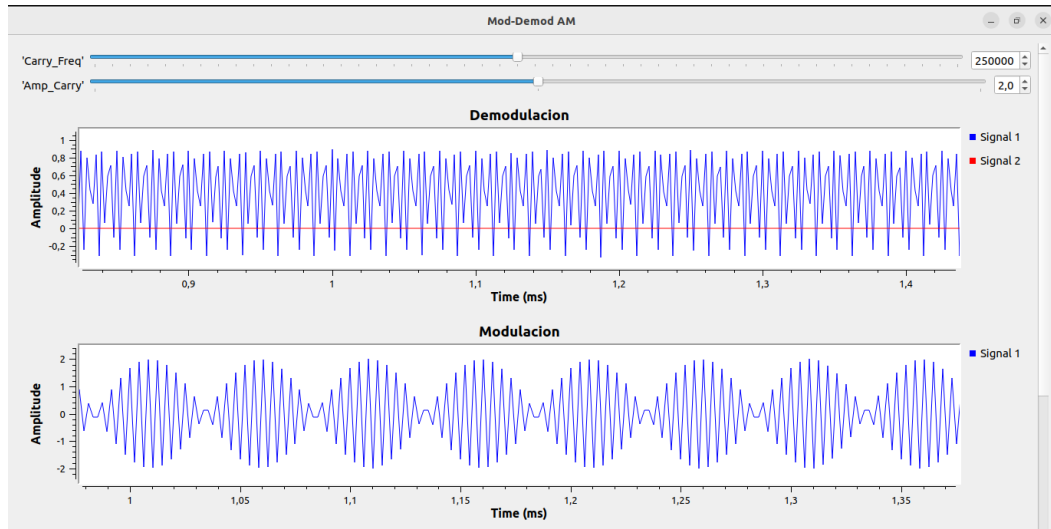


Figura 46. Modulación y demodulación AM de archivo de audio

En el transcurso de la práctica de laboratorio que empleó bloques de procesamiento para diseñar un sistema de demodulación de señales AM, se observa un éxito en la demodulación. El sistema diseñado demuestra su capacidad para extraer la señal de audio original de manera precisa a partir de la señal modulada en amplitud. La comparación entre el audio demodulado y la señal de audio original sirvió como indicador de la fidelidad de la demodulación.

Los resultados obtenidos incluyeron un análisis espectral de la señal modulada y la señal demodulada utilizando las herramientas proporcionadas por GNU Radio. Lo cual ayudó a identificar las componentes de frecuencia asociadas con la modulación AM, así como comprender cómo estas frecuencias se reflejaban en la señal demodulada.

### 3.2.9 Práctica 9: Modulador FM

#### Objetivo

Esta práctica de laboratorio tiene como objetivo explorar los conceptos y aplicaciones de la modulación de frecuencia utilizando GNU Radio. La modulación de frecuencia, en contraste con la modulación de amplitud (AM), varía la frecuencia de la onda portadora de acuerdo con la señal de entrada, ofreciendo ventajas como una mayor inmunidad al ruido.

## Materiales y equipos

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio
- Archivo de audio .wav

## Procedimiento

La Figura 47 muestra el diagrama del sistema a desarrollar

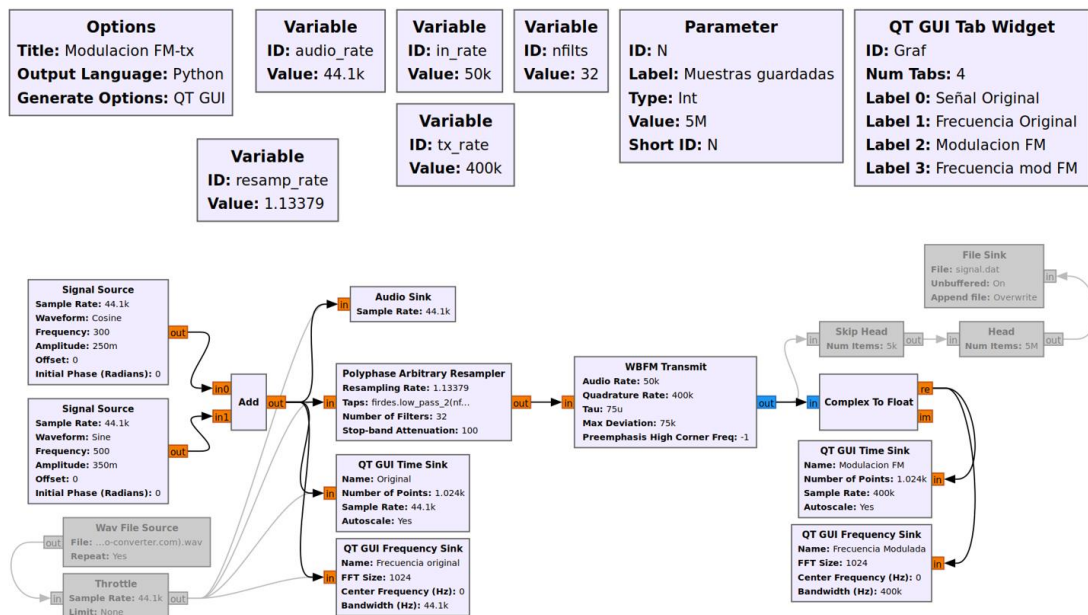


Figura 47. Diagrama de práctica 9

### 1. Generar la señal

La señal puede ser generada a través de dos bloques **Signal Source**, los cuales serán sumados para obtener una señal la cual se pueda distinguir por su forma de onda al momento de recuperar la señal.

Pero también es posible hacer uso de un archivo de audio el cual puede ser modulado para observar su comportamiento. Para ello se deshabilita los bloques generadores y se habilita los otros dos bloques a usar, la Figura 48 muestra cómo debería quedar una vez habilitado.

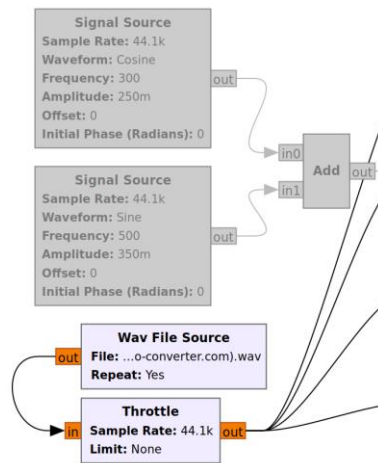


Figura 48. Habilitar bloques

Conectar un bloque Audio Sink permitirá escuchar a través de una salida de audio conectada al PC la señal a modular, esto es útil para comparar la calidad en el proceso de demodulación.

## 2. Crear variables a usar

Algunas de las variables que se necesita son la frecuencia de muestreo del sistema, la frecuencia de salida necesaria para el bloque de remuestreo, una frecuencia de muestreo para la transmisión y salida del bloque modulador, número de filtros a usar en el bloque de remuestreo y variables de control para la visualización de graficas en ventanas.

## 3. Remuestrear la frecuencia

Conectar y configurar un bloque Polyphase Arbitrary Resampler, el cual a través de un banco de filtros internos en el bloque realiza un remuestreo de la señal cambiando la frecuencia de la señal por un factor que puede ser mayor o menor que 1, esto ayuda a reducir el uso de procesos del PC y mejora la calidad de la señal a su salida.

## 4. Conectar bloque modulador

Configurar el bloque WBFM Transmit, con los parámetros necesarios para la modulación, este permite realizar una modulación FM de banda ancha tomando una sola entrada de muestras de datos flotantes produciendo a la salida una señal de banda base compleja y modulada.

## 5. Visualizar señal

Conectar los bloques de visualización QT GUI Time Sink y QT GUI Frequency Sink para la visualización de las señales.

### Resultados de la práctica

Las siguientes gráficas muestran las señales resultantes de la modulación FM.

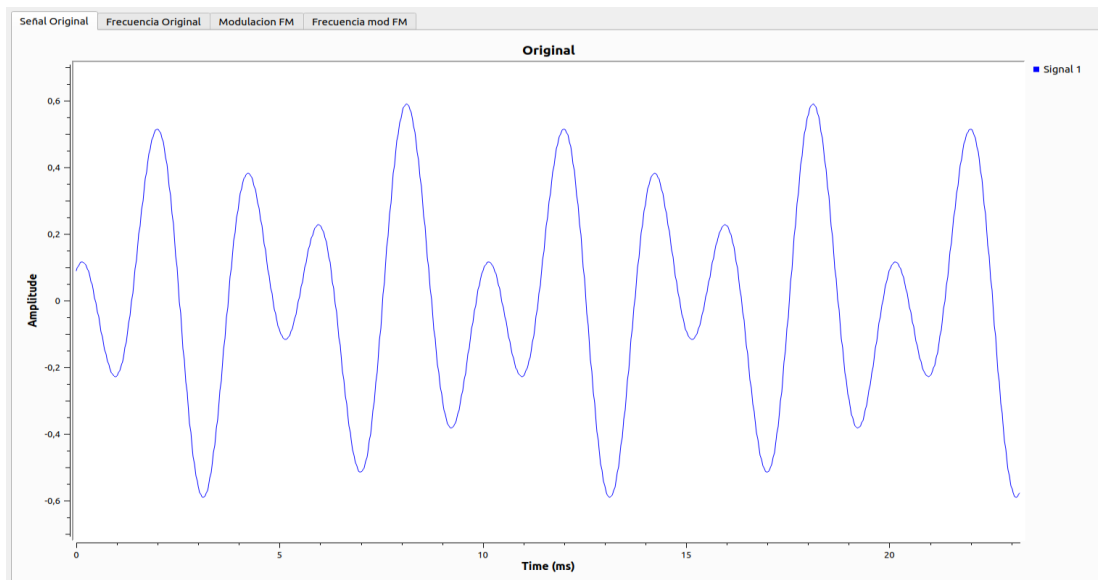


Figura 49. Señal original

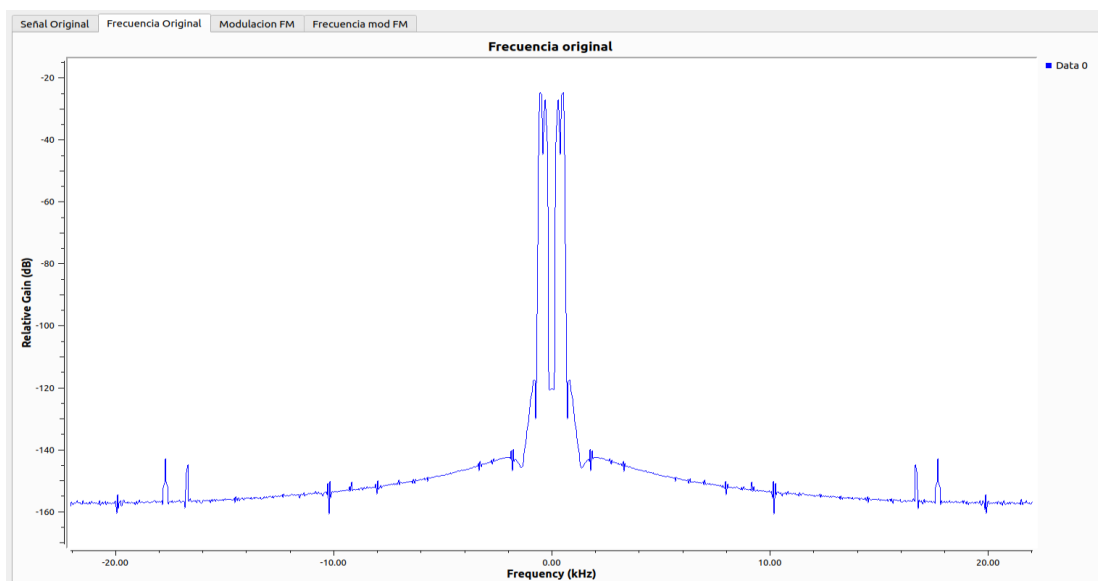


Figura 50. Frecuencia Original

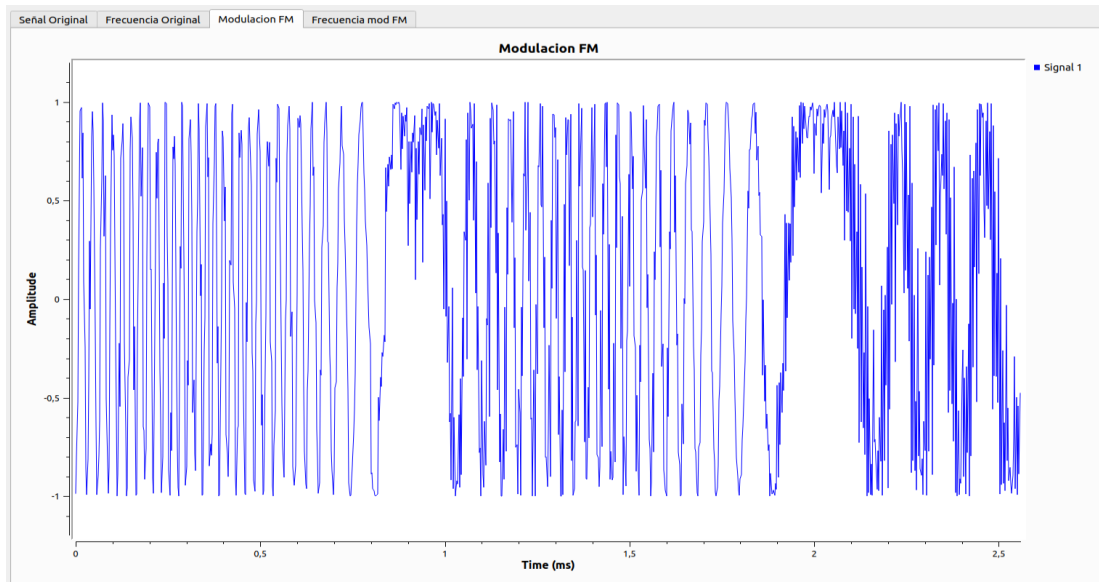


Figura 51. Señal resultante a la salida del modulador

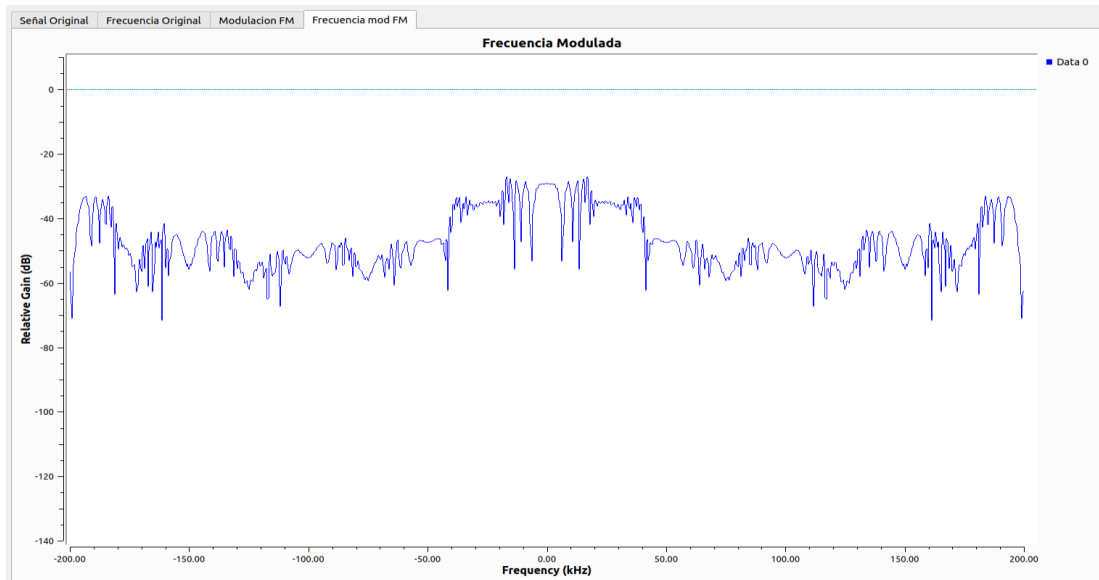


Figura 52. Frecuencia resultante a la salida del modulador

Mediante la visualización de distribución espectral de la señal FM utilizando herramientas de análisis de espectro en GNU Radio se puede analizar cómo varían las componentes de frecuencia en función de los parámetros de modulación. Este análisis proporcionará información valiosa sobre el ancho de banda ocupado por la señal FM y permite a los estudiantes optimizar los parámetros para lograr una transmisión eficiente.

### 3.2.10 Práctica 10: Demodulador FM

#### Objetivo

Esta práctica de laboratorio tiene como objetivo explorar los conceptos fundamentales de la demodulación FM utilizando GNU Radio. Durante el desarrollo de esta práctica, los estudiantes adquirirán conocimientos prácticos sobre la extracción de la información modulada en frecuencia, lo que es crucial para sistemas de radiodifusión, comunicaciones inalámbricas y transmisiones de audio.

#### Materiales y equipos

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio
- Archivo de señal FM

#### Procedimiento

La Figura 53 muestra el diagrama del sistema a desarrollar

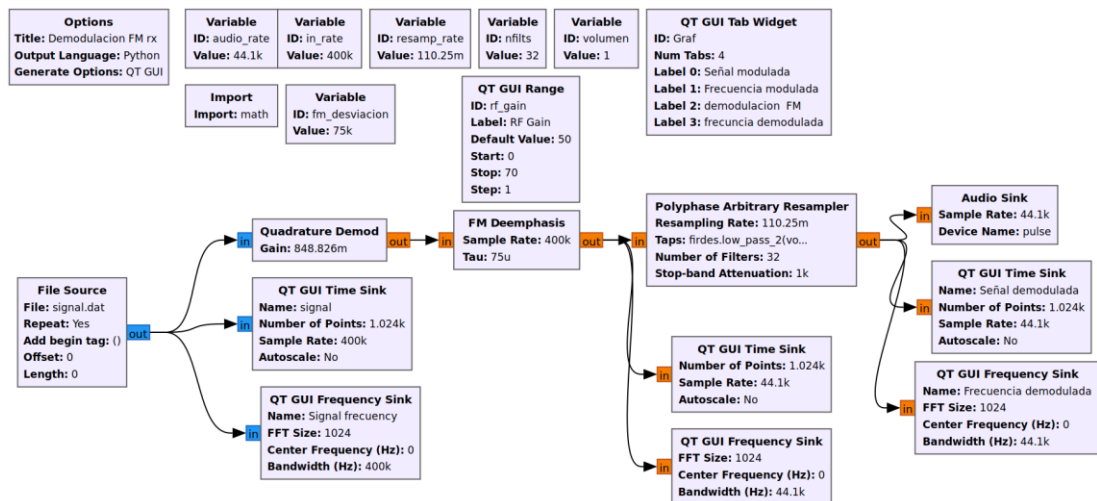


Figura 53. Diagrama de práctica 10

## **1. Generar la señal**

La señal en este caso es un archivo .dat, este almacena datos generados en un sistema específico, este bloque contiene la información almacenada de la modulación realizada anteriormente, por lo que al pasar por el proceso de demodulación se recuperara la información.

## **2. Crear las variables a usar**

Algunas de las variables que se necesita son la frecuencia de muestreo del sistema, la frecuencia de entrada necesaria para el bloque de remuestreo, una frecuencia de muestreo para la transmisión y entrada del bloque modulador, número de filtros a usar en el bloque de remuestreo y variables de control para la visualización de graficas en ventanas.

## **3. Conectar el bloque demodulador**

Conectar el bloque Quadrature Demod y configurar sus parámetros para que pueda recibir la señal en banda base que entrega el bloque File Source y entregar a su salida datos flotantes que representan la demodulación de la señal.

## **4. Mejorar la calidad de la señal**

Conectar el bloque FM Deemphasis y configurar sus parámetros para que a través de sus filtros integrados mejore y reduzca la posibilidad de ruido que exista en la señal.

Conectar y configurar el bloque Polyphase Arbitrary Resampler realizando un remuestreo de la señal mejora la calidad que se obtiene a la salida del bloque.

## **5. Visualizar la señal**

Conectar los bloques de visualización necesarios para observar el comportamiento de la señal en el dominio del tiempo y frecuencia. Un bloque Audio Sink ayudará a reproducir el audio si se utilizó un archivo de audio para la modulación FM.



## Resultados de la práctica

La Figura 54 muestra las señales obtenidas del sistema generado de acuerdo con el diagrama establecido en la

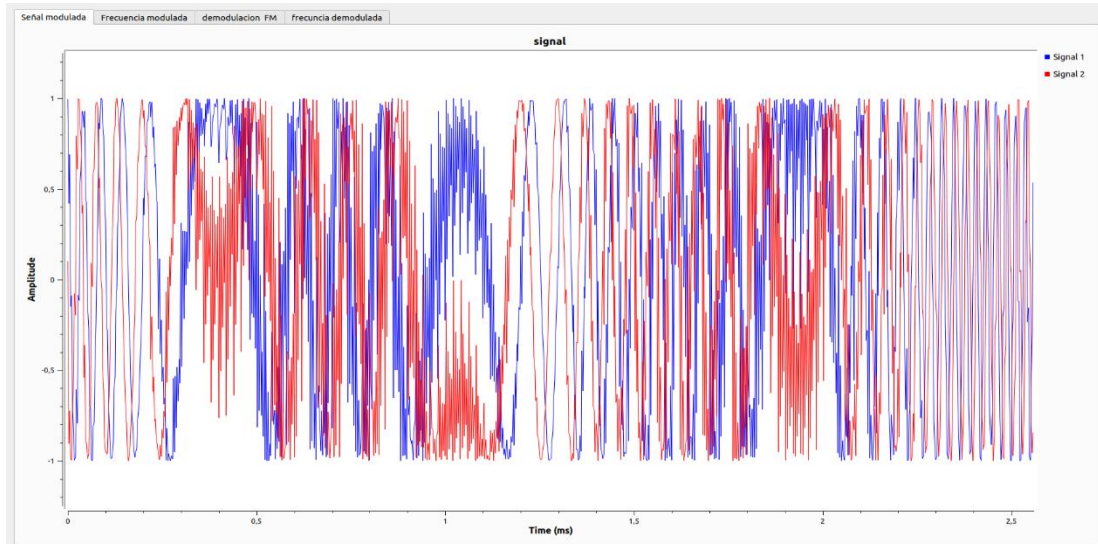


Figura 54. Señal modulada recibida a través del archivo de datos

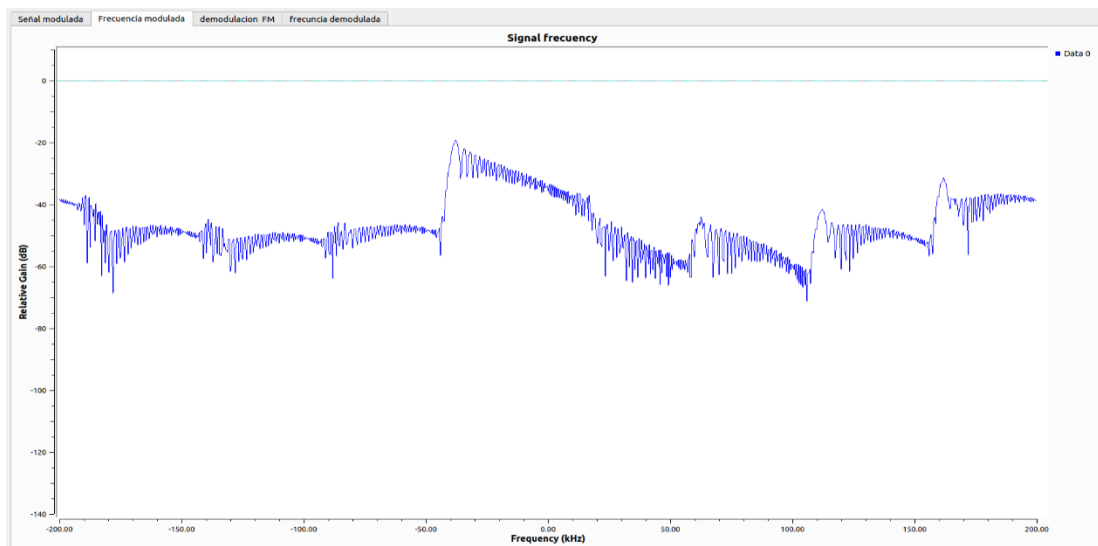


Figura 55. Frecuencia de la señal modulada

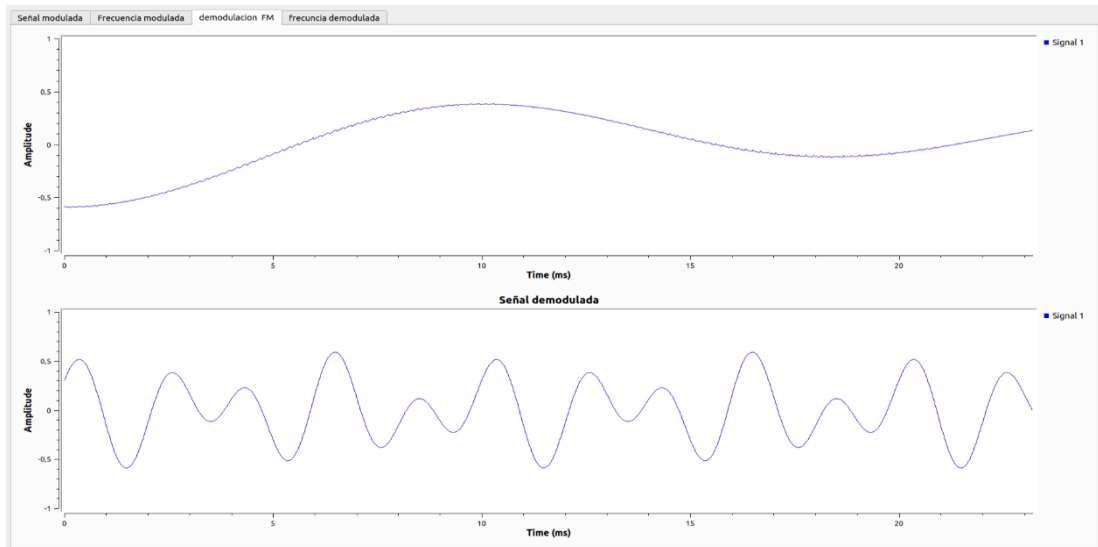


Figura 56. Salida del sistema de demodulación GFM Señal recuperada

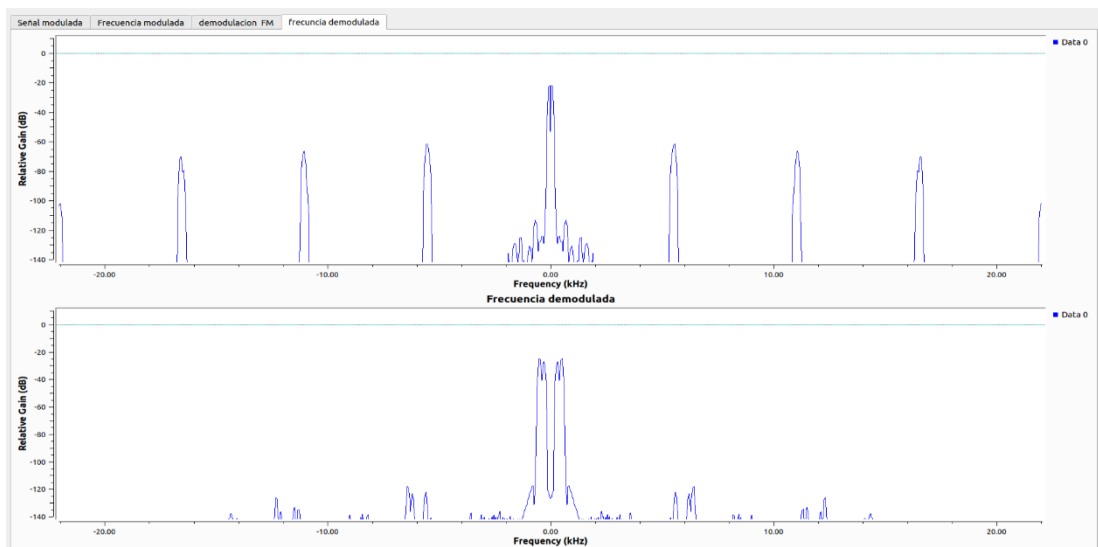


Figura 57. Frecuencia de la señal recuperada

La configuración de los parámetros de demodulación, como la constante de desviación y los filtros de bajo paso, ayudaron a maximizar la fidelidad en la reproducción de la información original. Esta capacidad para ajustar con precisión los parámetros demuestra una comprensión profunda de los principios de la demodulación FM y destaca la habilidad para adaptar el receptor a diferentes condiciones de transmisión.

A través del análisis detallado de la señal demodulada, se puede evaluar la calidad de la demodulación en términos de distorsiones y pérdida de información. Esta evaluación permite identificar posibles mejoras en el diseño del receptor y ajustar la configuración para lograr una recuperación más precisa de la información original.

### **3.2.11 Práctica 11: Receptor de Frecuencias FM**

#### **Objetivo**

El propósito de esta actividad consiste en desarrollar un sistema para recibir señales en frecuencia modulada (FM). Utilizando equipos USRP N210 como dispositivos receptores, se creará un sistema empleando bloques de modulación FM disponibles en GNU Radio. Este sistema captará emisiones de 88 MHz a 108 MHz y las mostrará en ventanas de visualización y reproducción de audio, ajustando los parámetros necesarios para lograr un funcionamiento óptimo.

#### **Materiales y equipos**

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio
- USRP N210
- Cable Gigabit Ethernet
- Antenas

#### **Procedimiento**

La Figura 58 muestra el diagrama del sistema a desarrollar.

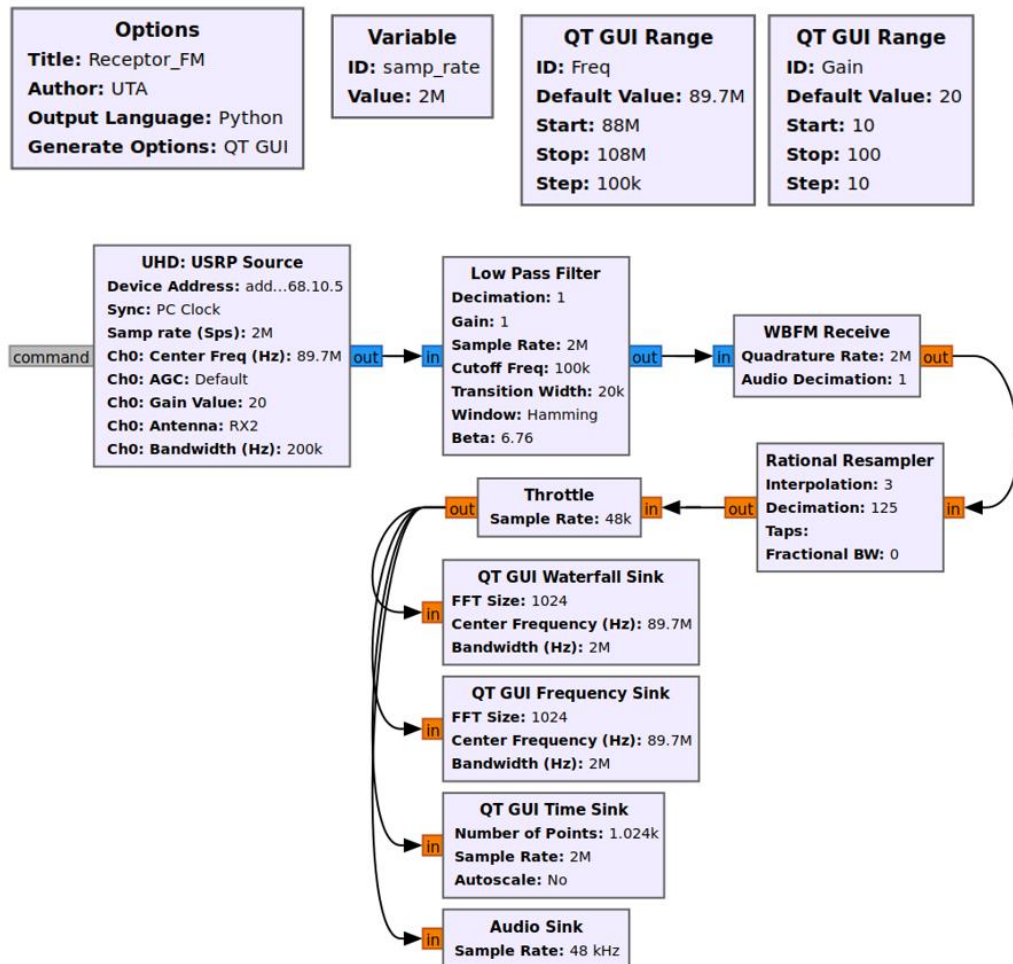


Figura 58. Diagrama de práctica 11

## Procedimiento

### 1. Conectar y preparar el equipo USRP N210

Mediante la conexión a través de cable Gigabit Ethernet establecer la comunicación entre la PC y el equipo USRP, asegurándose que el equipo sea reconocido mediante los comandos UHD instalados anteriormente. La PC debe estar configurada con una dirección de red estática que se encuentre dentro del rango establecido.

### 2. Obtener la señal

La fuente del sistema será el equipo USRP que servirá como receptor, para ello el bloque UHD: USRP Source permite conectar y recibir señales que se encuentren en ese momento.

Configurar el bloque de acuerdo con los parámetros que se necesiten como las frecuencias a captar, la ganancia del canal, entre otros.

### **3. Crear las variables a usar**

La tasa de muestreo del sistema establecerá la cantidad de muestras que tomara el equipo para procesarlas, los bloques de control ayudan a variar parámetros como la frecuencia central del receptor y la ganancia que tendrá el canal.

### **4. Filtrar la señal**

Conectar un bloque Low Pass Filter para configurar una frecuencia de corte de acuerdo con el ancho de banda de los canales FM, particularmente las emisoras de FM se establecen en 200 KHz para transmisión de audio, por lo que el filtro pasa bajo se establece en 100 KHz.

### **5. Demodular la señal FM**

Configurar el bloque demodulador WBFM Receive, cambiando los valores de las frecuencias de muestreo con el objetivo de reducir las muestras tomadas hasta obtener una frecuencia de audio aceptable.

### **6. Ajustar la señal**

Conectar un bloque Rational Reampler y configurar sus parámetros para que a la salida del bloque se obtenga una frecuencia de 48 KHz que es el valor que las tarjetas de audio de las PC trabajan para ello se realiza una interpolación y decimación de las muestras recibidas originalmente.

### **7. Visualizar y reproducir la señal.**

Conectar los bloques necesarios para la visualización de la señal y mediante un bloque Audio Sink poder reproducir y escuchar a través de dispositivos de salida de audio.

## Resultados de la práctica

La Figura 59 muestra el resultado del sistema de recepción FM, en ella se capta una emisora de FM.

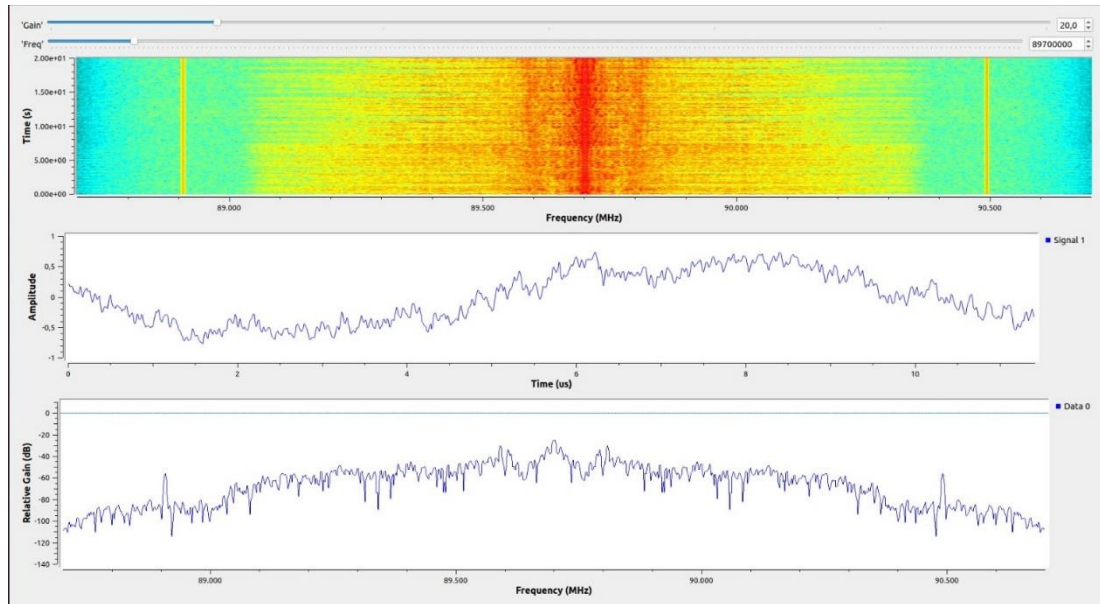


Figura 59. Receptor FM

Mediante bloques de procesamiento de señales se diseñó un sistema para la visualización de señales FM, pudiendo analizar el espectro de la señal y configurando varios parámetros como la eliminación de ruido mediante un filtro y la configuración del ancho de banda. Los bloques de control ayudan al usuario a observar el cambio y efecto producido cuando se busca una señal que se encuentre dentro del rango de 88 MHz a 108 MHz. El equipo USRP será capaz de detectar las frecuencias que estén dentro de ese rango asegurándose de que se encuentre bien establecida la comunicación entre el SDR y la PC.

### 3.2.12 Práctica 12: Transmisión y recepción de señales FM

#### Objetivo

La práctica tiene como objetivo diseñar un sistema para la transmisión de señales FM, mediante bloques de procesamiento de señales se modulará y preparará la señal para enviarse en una frecuencia en específico del espectro radioeléctrico, el cual será

captado por el sistema de recepción diseñado anteriormente completando el sistema de comunicaciones.

## Materiales y equipos

- Computadora
- Sistema Operativo Ubuntu 20.04
- Software GNU Radio
- USRP N210
- Cable Gigabit Ethernet
- Antenas

## Procedimiento

La Figura 60 muestra el diagrama del sistema a desarrollar

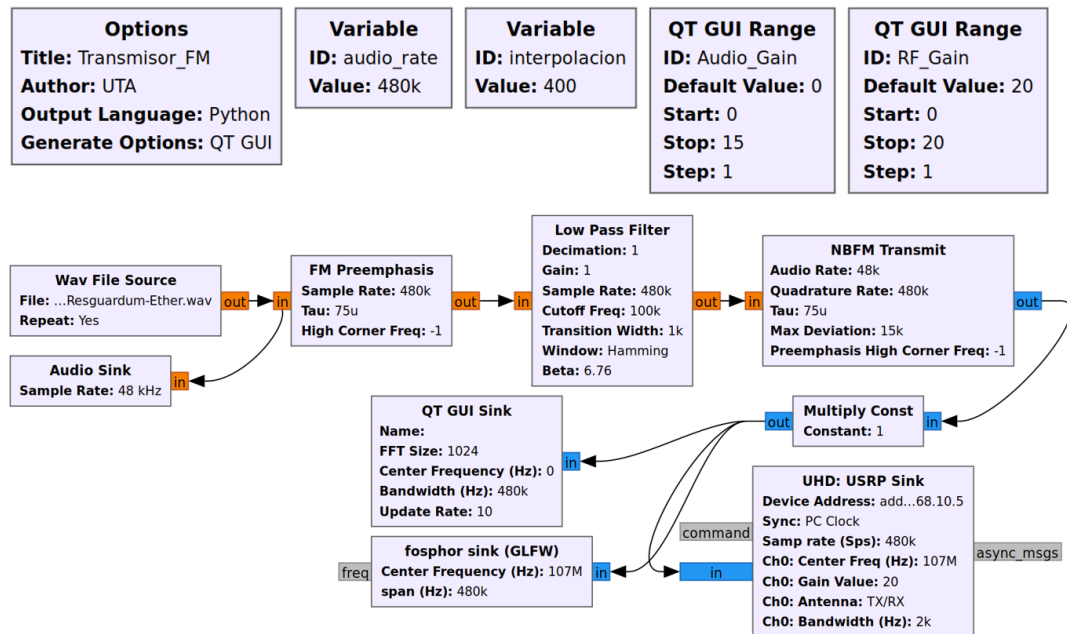


Figura 60. Diagrama de práctica 12

### **1. Conectar y preparar el equipo USRP N210**

Mediante la conexión a través de cable Gigabit Ethernet establecer la comunicación entre la PC y el equipo USRP, asegurándose que el equipo sea reconocido mediante los comandos UHD instalados anteriormente. La PC debe estar configurada con una dirección de red estática que se encuentre dentro del rango establecido.

### **2. Establecer la fuente**

La fuente será un archivo de audio este tiene una tasa de bits de 44100 Hz, este archivo .wav se prepara para ser transmitido en una frecuencia en específico.

### **3. Crear las variables a usar**

Las variables necesarias para la transmisión son la tasa de muestreo del sistema, un valor de interpolación necesario para establecer el valor de la cuadratura en el bloque transmisor FM, bloques de control para la ganancia del audio y la ganancia del canal del equipo USRP.

### **4. Ajustar la señal**

Conectar y configurar la señal a un bloque FM Preemphasis para mejorar la calidad de la señal antes de la modulación, un bloque de filtro Low Pass Filter ayuda a limitar el espectro del audio y el ancho de banda usado, la frecuencia de corte se encuentra debajo del valor utilizado por emisoras FM para la transmisión de audio que es de 200 KHz. Esto permitirá generar una atenuación o amplificación dependiendo del valor de la ganancia que obtenga el filtro.

### **5. Conectar el bloque modulador**

Conectar el bloque de transmisión en banda angosta NBFM Transmit y configurar sus parámetros necesarios para la velocidad de muestreo del sistema mediante el valor de la interpolación.



## 6. Conectar el bloque para transmisión USRP

Conectar el bloque UHD: USRP Sink y configurar sus parámetros para la transmisión de señales a través de dispositivos USRP, estableciendo una frecuencia en el cual transmitir, esta práctica envía la señal a través de la frecuencia 107 MHz

## 7. Visualizar las señales

Colocar bloques de visualización de señales para observar el proceso de la señal en el momento de la transmisión.

## Resultados de la práctica

La Figura 61 muestra el proceso de modulación de la señal enviada resultante de la conexión realizada a partir del diagrama establecido en la Figura 60.

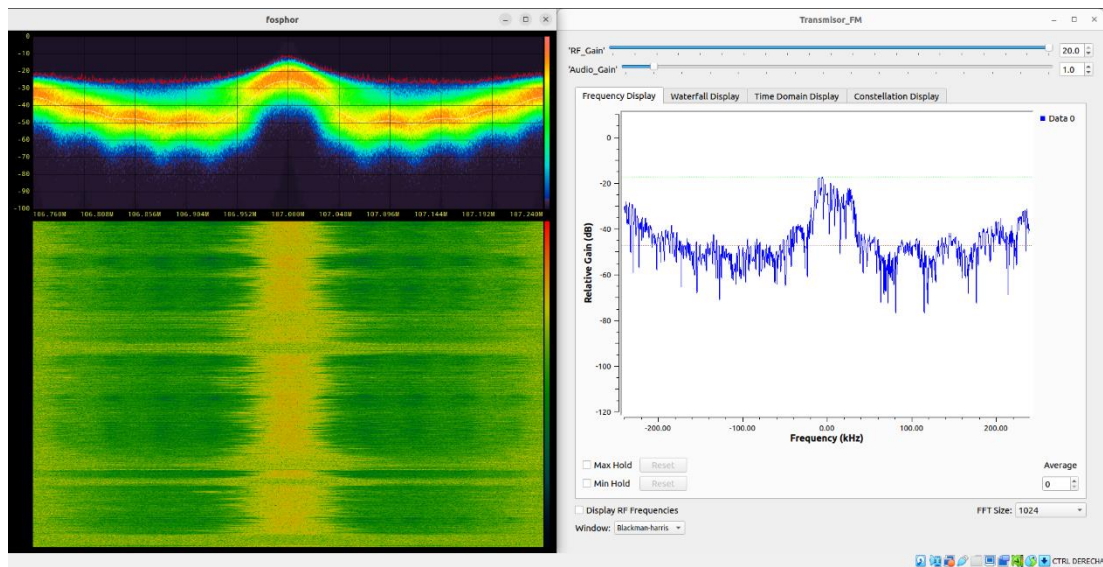


Figura 61. Frecuencia de la señal enviada a través del trasmisor

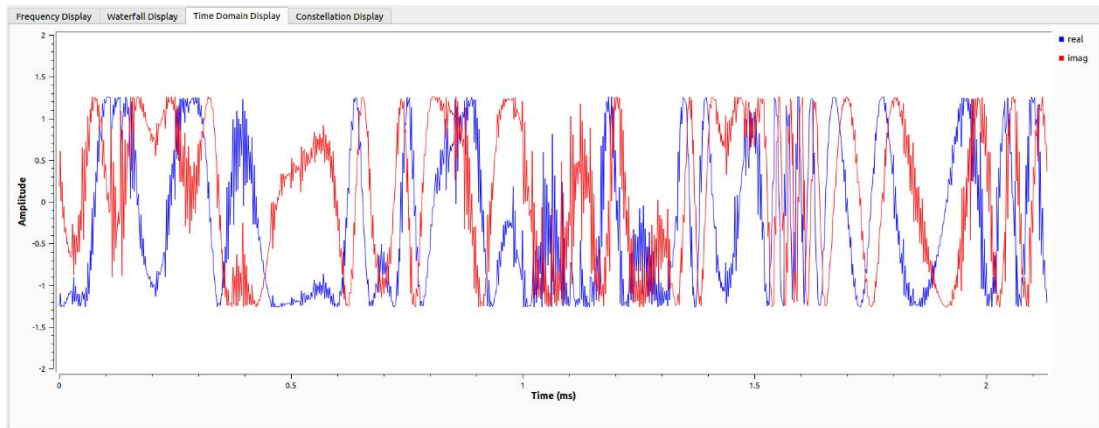


Figura 62. Modulación FM de la señal

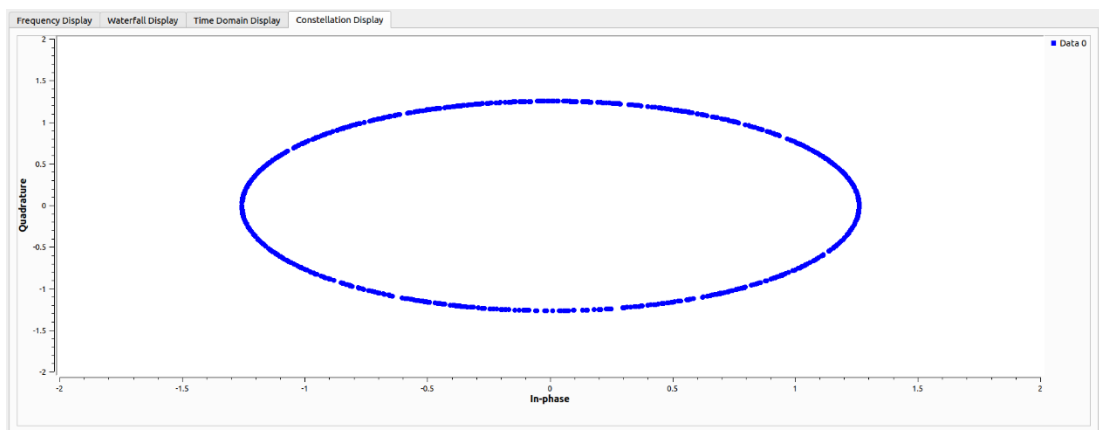


Figura 63. Diagrama de constelación de la señal

Por otra parte, en el lado del Receptor FM creado anteriormente de igual manera con un equipo USRP, se deberá sintonizar la frecuencia central colocada en los parámetros del bloque de transmisión UHD.

Si la transmisión de la señal se está realizando de manera adecuada, se observará en las ventanas los cambios en la frecuencia recibida, de la misma manera como se hace uso de un bloque para la reproducción de audio este comenzará a reproducir el audio enviado a través del sistema de transmisión.

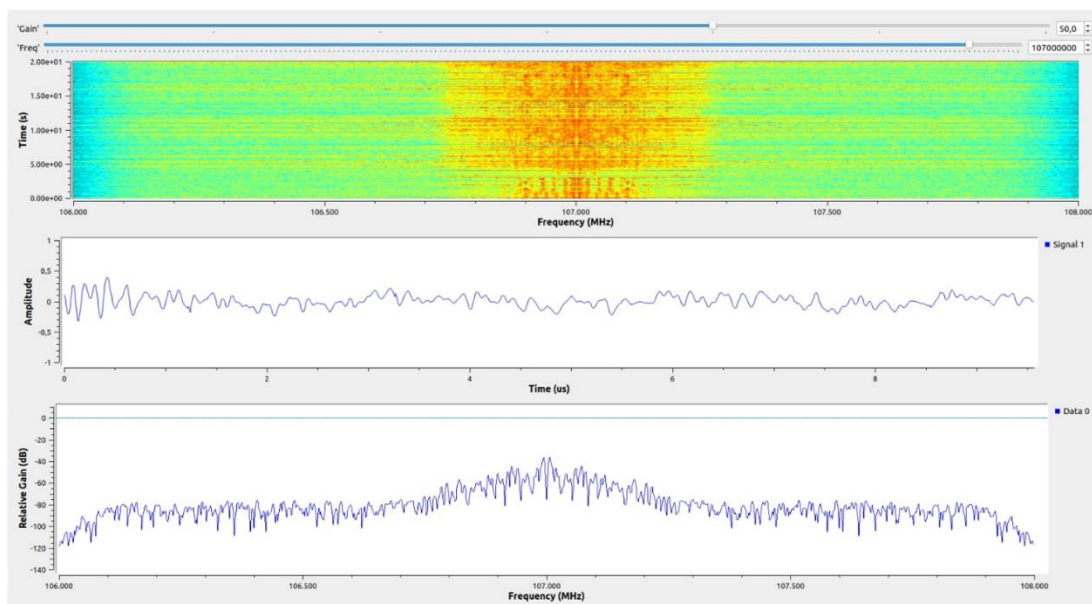


Figura 64. Señal FM captada en el recetor FM

Mediante bloques de procesamiento se implementó un sistema de transmisión de señales en frecuencia modulada, en la cual se realizó distintas configuraciones como el filtrado, cambios en la tasa de muestreo y el acondicionamiento necesario para la modulación FM en banda estrecha, estableciendo la comunicación del sistema con el equipo USRP y determinado una frecuencia para la transmisión de 107 MHz, lo cual haciendo uso del sistema de recepción creado se comprobó la funcionalidad y la captación de la señal por parte de un segundo USRP ubicado en el lado contrario del laboratorio.

### 3.2.13 Práctica 13: Red GSM con USRP

#### Objetivo

Esta práctica tiene como objetivo implementar una estación base celular usando herramientas de software libre soportadas sobre una plataforma de radio definida por software OpenBTS y un hardware SDR de radiofrecuencia USRP. Como parte fundamental en el aprendizaje del estudiante será capaz de manejar servicios GSM y conocer el proceso de registro y suscripción de usuarios a través de terminales intuitivos y demostrativos.

## Materiales y equipos

- Computadora
- Sistema Operativo Ubuntu 14.04
- Software OpenBTS y servicios
- Software Asterisk
- USRP N210
- Cable Gigabit Ethernet
- Antenas
- 2 teléfonos móviles

## Procedimiento

La Figura 65 muestra el diagrama del sistema a desarrollar

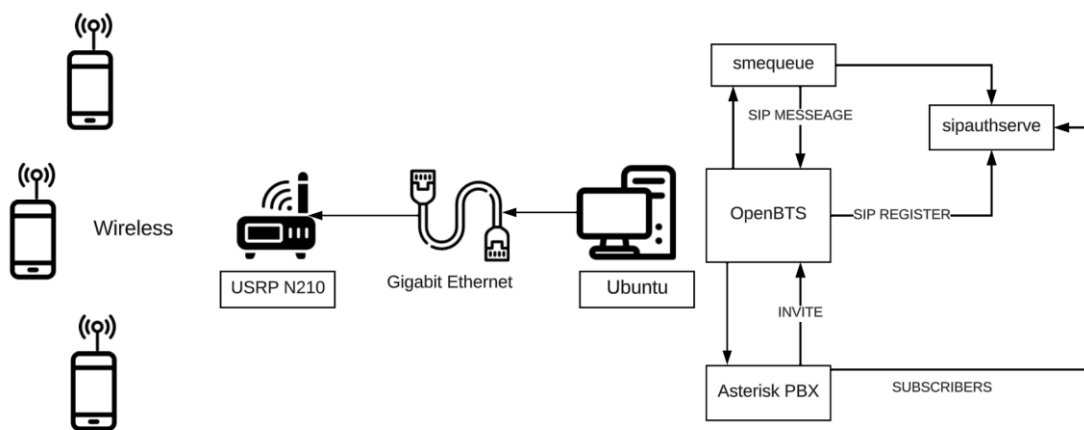


Figura 65. Diagrama implementación red GSM

### 1. Instalación de sistema operativo Ubuntu 14.04.5

Mediante el software VirtualBox montamos una máquina virtual 14.04.5 LTS de 64 bits, siguiendo los pasos realizados en el laboratorio 1 Introducción a GNU Radio y USRP en la creación de máquinas virtuales.

## **2. Instalación de controlador UHD**

Una vez creada la máquina virtual instalamos el controlador UHD. Los controladores permiten la comunicación entre la PC y el equipo, la versión del controlador a instalar debe ser compatible con el sistema operativo y también con la versión de OpenBTS a usar, se recomienda la versión 003.007.003 de UHD

## **3. Descarga de repositorios necesarios**

Mediante el uso de repositorios y junto con el comando git clonar en la maquina el repositorio <https://github.com/RangeNetworks/dev.git> que alberga los servicios de OpenBTS, smqueue, sipauthserve y Asterisk.

De la misma manera los servicios asociados para el funcionamiento de smqueue.

## **4. Modificaciones de archivos necesarios**

Antes de ejecutar la instalación es necesario modificar el archivo build.sh, esto a través de un editor de texto como vim, nano o gedit

Después de cambiar los necesario ejecutar el comando buiid.sh para empezar la instalación de paquetes y servicios necesarios.

## **5. Instalación de OpenBTS**

Dentro de la carpeta de compilación e instalación de servicios instalar el software OpenBTS, el cual una vez iniciado pedirá varias confirmaciones los cuales afirmaremos para la instalación. Al terminar la instalación es necesario reiniciar la PC

## **6. Inicio de servicios**

Una vez reiniciado el PC, conectamos a través del cable Gigabit Ethernet el equipo USRP al PC, establecemos una dirección de red estática y asegurándose de que la máquina virtual este configurado el tipo de red en Bridge comprobamos la conexión entre ambos, para ello se puede usar el comando ping o los comandos de UHD uhd\_usrp\_probe o uhd\_find\_devices.

Inicializar los servicios sipauthserve, smqueue, astersik y openbts.

Ejecutamos el programa OpenBTS, el dispositivo USRP reaccionará al servicio y encenderá la luz led indicadora de la tarjeta transceiver.

### **7. Registrar suscriptores**

En cada uno de los dispositivos móviles que vayamos a conectar a la red buscamos la red creada por OpenBTS esta tiene el nombre de *Test PLMN 1-1*.

Mediante las configuraciones de redes móviles de cada teléfono buscar la red que aparecerá disponible para el registro, registrase en la red y esperar a recibir un mensaje automático en el cual aparecerá el número IMSI del teléfono conservar este numero dado que servirá para hacer los registros.

### **8. Registro en sipauthserve**

Para que se detecte los abonados estos deben ser registrados en la base de datos de sipauthserve y Asterisk, por lo cual se debe preparar y tener listo los números IMSI de cada teléfono a suscribir e igual sus respectivos umeros que se les vaya a colocar.

### **9. Configuración para Asterisk**

Dentro de la base de datos de Asterisk hay que realizar el mismo procedimiento de registro tanto de número IMSI como de número telefónico.

Modificar el archivo extensions.conf para crear el plan de marcado que tendrá la red y cargar el plan dentro del CLI de Asterisk.

Con esto el servicio de mensajería y llamadas entre suscriptores debe funcionar.

### **10. Pruebas de red y resultados**

Mensajería entre suscriptores

## Teléfono 1

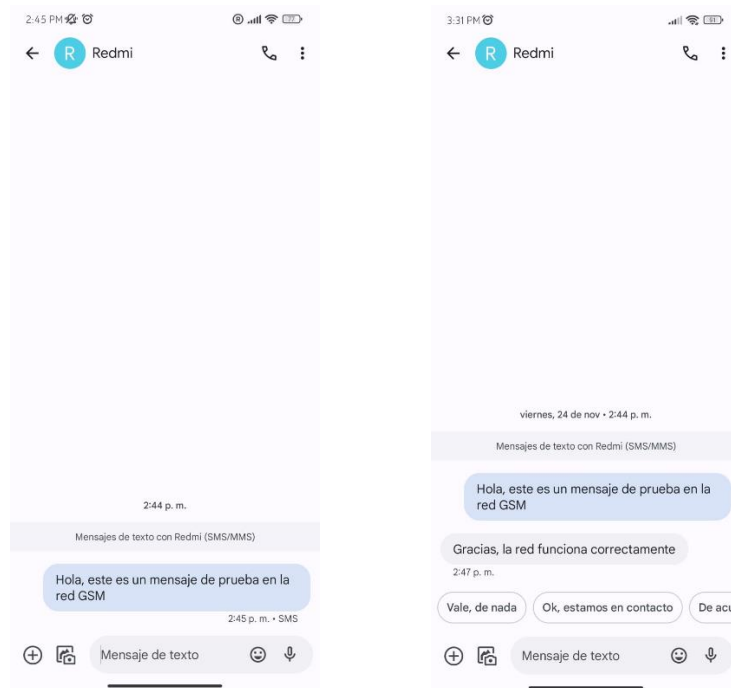


Figura 66. Envío y respuestas de mensajería entre suscriptores Poco-Redmi

## Teléfono 2



Figura 67. Envío y respuestas de mensajería entre suscriptores Redmi-Poco

## Llamadas entre suscriptores



Figura 68. Llamada entrante entre suscriptores Poco-Redmi

Los servicios asociados como sipauthserve, smqueue y Asterisk, integran los fundamentos de servicios VoIP, con lo cual se podría incluir autenticación de usuarios, envío de mensajerías y servicios de llamadas avanzadas con el uso e integración de Asterisk.



## **CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES**

### **4.1 Conclusiones**

GNU Radio junto a equipos SDR proponen una herramienta en el desarrollo de sistemas de comunicaciones. Los estudiantes pueden aprender algoritmos de procesamiento de señales y parámetros de transmisión y recepción RF. La capacidad de utilizar bloques de procesamiento permite adaptar el sistema a distintas configuraciones y requisitos lo que facilita la implementación de sistemas más complejos. Tanto los equipos USRP como GNU Radio son de código abierto, lo que significa que su diseño y código fuente son accesibles para todo tipo de usuarios. Esto fomenta la colaboración, la mejora y creación de nuevas funcionalidades por parte de la comunidad de estudiantes y docentes. Además, el enfoque de código abierto brinda a los usuarios la libertad de personalizar y adaptar los sistemas a sus propias necesidades.

El trabajo presentó la implementación de un conjunto de prácticas de laboratorio, mediante los controladores UHD que permiten la comunicación y configuración de equipos SDR se trabajó en temas como transmisión y recepción de señales, filtrado y modulaciones analógicas, análisis del espectro y se estableció una red GSM privada utilizando OpenBTS y un equipo USRP N210 como transceptor, lo que significa que es posible proporcionar servicios de telefonía móvil dentro de un espacio controlado, lo cual es muy útil para realizar pruebas u experimentaciones de redes móviles en un entorno de laboratorio.

Se realizó la documentación detallada de 13 prácticas de laboratorio para complementar los conocimientos en sistemas de comunicaciones, mostrando e indicando los bloques utilizados y sus configuraciones, para ello se realizó el estudio de los conceptos necesarios para el control del dispositivo USRP a usar y de igual manera del software GNU Radio como medio de aprendizaje, cumpliendo con los objetivos que se indicaron al inicio de este trabajo.

## 4.2 Recomendaciones

Es recomendable al momento de la creación de las máquinas virtuales asignar una cantidad mínima de 4 Gb de memoria RAM y al menos 2 núcleos de procesadores dedicados para la máquina virtual, ciertamente mientras más CPU tenga la máquina virtual, más rápido ejecutara múltiples tareas.

Se debe utilizar una versión UHD y GNU Radio compatible con el sistema operativo que se vaya a utilizar, si el sistema operativo es el más reciente las versiones actualizadas de estos programas no deberían dar problemas, pero si se utiliza versiones de sistemas operativos anteriores, la compatibilidad con los equipos USRP puede ocasionar problemas de conexión, por lo que en ese caso se debería instalar específicamente las versiones respectivas que sean compatibles con tal versión de sistema operativo.

Se recomienda el manejo cuidadoso de los equipos USRP al momento de cargar imágenes a las FPGA, seguir la guía práctica como se indica si es necesario hacer una actualización de imagen o consultar la página oficial de Ettus Research para tener más información acerca de la manipulación de la tarjeta FPGA del equipo USRP.

La velocidad de procesamiento de las señales y la toma de muestras de los sistemas depende de las capacidades de la PC, es recomendable hacer uso de bloques Throttle al momento de crear los sistemas para limitar y controlar el uso excesivo de CPU de la máquina principal.

Es necesario conocer previos conceptos y conocimientos en el manejo de comandos básicos de terminal en Linux, dado que este ofrece un acceso más directo en el sistema lo que permite realizar tareas de manera precisa.

Si se transmite y recibe señales, se recomienda tener cierta distancia entre ambos equipos USRP, ya que al estar demasiado juntos pueden crear distorsiones en las señales recibidas.

Comprobar la correcta conexión y comunicación entre la PC y el equipo USRP dado que para algunos servicios que se usan en la creación de la red GSM si el servicio no

detecta un equipo transceiver disponible este no ejecutará y creará archivos vacíos de ejecución lo que impedirá volver a conectar la tarjeta del equipo USRP.

Es posible hacer los diseños de estos laboratorios usando otros sistemas de desarrollo SDR, entre ellos Matlab & Simulink y LabVIEW, considerando que sería beneficioso explorar otras alternativas de software, como Matlab y LabVIEW, para la configuración de los equipos USRP. Estas plataformas ofrecen entornos de desarrollo confiables y ampliamente utilizados en aspectos técnicos.

Los equipos USRP pueden considerarse para el estudio en el área de hacking ético. La capacidad de emular entornos de red y simular ataques permitiría a los estudiantes evaluar la eficacia de las defensas cibernéticas y probar la confiabilidad de los sistemas ante posibles amenazas.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] I. Pinar Dominguez y J. Murillo Fuentes, Laboratorio de comunicaciones digitales Radio Definida por Software, Sevilla: Universidad de Sevilla, 2011.
  
- [2] F. J. Terreros Quiroz, «Emulador de un sistema de comunicaciones utilizando tecnología SDR,» SANGOLQUÍ / ESPE / 2010, SANGOLQUÍ , 2010.
  
- [3] Y. L. Bermemo Guerrero, «Diseño y desarrollo de guías de laboratorios utilizando la herramienta GNU Radio con base a la tecnología de Radio Definido por Software para el aprendizaje y comprensión del componente Sistemas de Comunicaciones,» Universidad de las Fuerzas Armadas, Sangolqui, 2022.
  
- [4] V. A. Balseca Acuña, «Sistema de comunicaciones empleando SDR para practicas multidisciplinarias de la carrera de Ingenieria en Electronica y Comunicaciones de la Universidad Tecnica de Ambato,» Universidad Tecnica de Ambato, Ambato, 2021.
  
- [5] J. Sanz Ruiz, «Implementación de modulaciones analógicas y digitales en GNU Radio,» Escuela Técnica Superior de Ingeniería de Telecomunicación, Murcia, 2020.
  
- [6] M. Sañay Carchi, «Implementación de prácticas de laboratorio para la materia de comunicación digital utilizando radio definida por software y GNU Radio,» Escuela Politecnica Nacional, Quito, 2020.
  
- [7] Á. Monteros Túquerres, «Diseño y elaboración de prácticas de laboratorio para la materia de fundamentos de comunicaciones usando radio definida por software,» Escuela Politecnica Nacional, Quito, 2019.

- [8] R. Jaimes Rico y S. Lazcano Salas, «SDR y GNU Radio como plataforma para un laboratorio de comunicaciones digitales,» *Revista electrónica de Computación, Informática*, vol. 8, nº 2, pp. 1-13, 2019.
- [9] ni.com, «National Instruments,» [En línea]. Available: <https://www.ni.com/es/shop/wireless-design-test/what-is-a-usrp-software-defined-radio.html>. [Último acceso: 2023 10 02].
- [10] J. Rodriguez, «Análisis software y hardware,» Universidad de Granada, Granada, 2017.
- [11] nuand, «bladerRF USB 3.0 Software Defined Radio,» New York.
- [12] Lime microsystems, «Lime microsystems,» [En línea]. Available: <https://limemicro.com/products/boards/limesdr/>. [Último acceso: 22 04 2023].
- [13] Rtl-sdr.com, «Rtl-sdr.com,» [En línea]. Available: [https://www.rtl-sdr.com/wp-content/uploads/2017/05/Digital\\_Forum\\_Talk\\_intro\\_to\\_rtlsdr.pdf](https://www.rtl-sdr.com/wp-content/uploads/2017/05/Digital_Forum_Talk_intro_to_rtlsdr.pdf). [Último acceso: 22 04 2023].
- [14] ni.com, «NI Ettus USRP Software Defined Radios,» National Instruments Corporation, 2023.
- [15] E. Research, «USRP B200/B210 Series Datasheet,» National Instruments Company.
- [16] E. Research, «USRP N200/N210 Series Datasheet,» National Instruments Company.

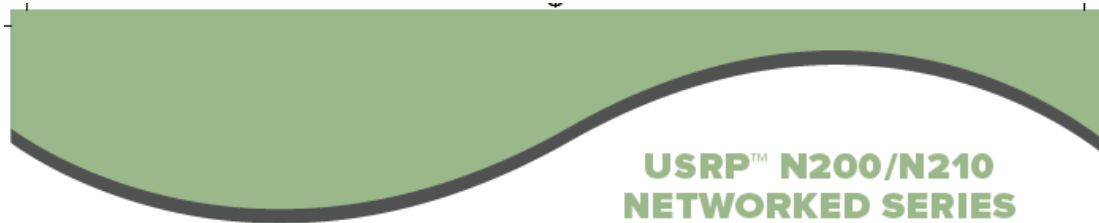
- [17] E. Research, «USRP X300/X310 Series Datasheet,» National Instruments Company.
- [18] N. Instruments, «USRP E310/E312/E313 Datasheet,» National Instruments, 2017.
- [19] N. Instruments, «USSRP RIO,» National Instruments, 2016.
- [20] E. Research, «USRP E320 Series Datasheet,» National Instruments Company.
- [21] E. Research, «USRP N310 Series Datasheet,» National Instruments Company.
- [22] E. Research, «USRP N320 Series Datasheet,» National Instruments Company.
- [23] «DigiKey,» [En línea]. Available: <https://www.digikey.com/es/product-highlight/d/digilent/ettus-usrp-n210-series-hardware>. [Último acceso: 9 12 2023].
- [24] «Ettus Research,» [En línea]. Available: <https://kb.ettus.com/N200/N210>. [Último acceso: 11 12 2023].
- [25] D. I. Pinar, «SOFTWARE DEFINED RADIO: USRP Y PLATAFORMAS DE DESARROLLO,» Universidad de Sevilla, Sevilla.
- [26] N. Instruments, «USRP Software Defined Radios,» 2019.
- [27] P. Corcuera, Introducción a Labview, Universidad de Cantabria.

- [28] N. Instruments, An Introduction to Software Defined Radio With NI LabVIEW and NI USRP Student Manual, 2013.
- [29] MathWorks, «Diseño y prototipado de sistemas SDR con MATLAB y Simulink,» MathWorks.
- [30] MathWorks, «Introducción a Radio definida por software (SDR),» [En línea]. Available: <https://la.mathworks.com/discovery/sdr.html>. [Último acceso: 05 11 2023].
- [31] E. Ramirez Gutierrez, «Sistemas de comunicaciones,» Universidad Nacional Autonoma de Mexico, Mexico, 2012.

# ANEXOS

## Anexo A. Datasheet USRP N210

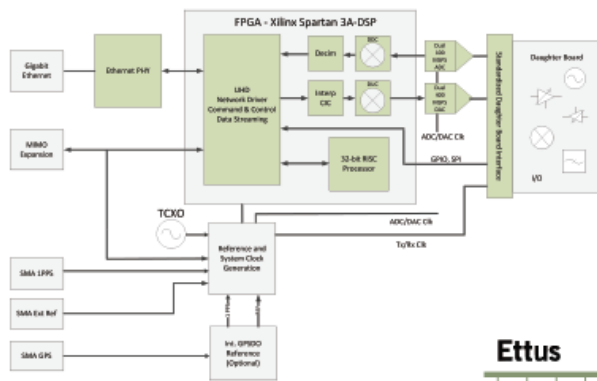
En la Figura A1 se muestra el Datasheet del equipo USRP N210



### SPECIFICATIONS

Spec	Typ.	Unit	Spec	Typ.	Unit
<b>POWER</b>			<b>RF PERFORMANCE (W/ WBX)</b>		
DC Input	6	V	SSB/LO Suppression	35/50	dBc
Current Consumption	1.3	A	Phase Noise (1.8 Ghz)		
w/ WBX Daughterboard	2.3	A	10 kHz	-80	dBc/Hz
<b>CONVERSION PERFORMANCE AND CLOCKS</b>			100 kHz	-100	dBc/Hz
ADC Sample Rate	100	MS/s	1 MHz	-137	dBc/Hz
ADC Resolution	14	bits	Power Output	15	dBm
ADC Wideband SFDR	88	dBc	IIP3	0	dBm
DAC Sample Rate	400	MS/s	Receive Noise Figure	5	dB
DAC Resolution	16	bits	<b>PHYSICAL</b>		
DAC Wideband SFDR	80	dBc	Operating Temperature	0 to 55°	C
Host Sample Rate (8b/16b)	50/25	MS/s	Dimensions (l x w x h)	22x16x5	cm
Frequency Accuracy	2.5	ppm	Weight	1.2	kg
w/ GPSDO Reference	0.01	ppm			

• All specifications are subject to change without notice.



### ABOUT ETTUS RESEARCH:

Ettus Research is an innovative provider of software defined radio hardware, including the original Universal Software Radio Peripheral (USRP) family of products. Ettus Research products maintain support from a variety of software frameworks, including GNU Radio. Ettus Research is a leader in the GNU Radio open-source community, and enables users worldwide to address a wide range of research, industry and defense applications. The company was founded in 2004 and is based in Mountain View, California. As of 2010, Ettus Research is a wholly owned subsidiary of National Instruments.

1043 North Shoreline Blvd  
Suite 100  
Mountain View, CA 94043  
P 650.967.2870 [www.ettus.com](http://www.ettus.com)  
F 866.807.9801

Figura A1. Datasheet USRP N210



## Anexo B. Datasheet antenas VERT900

En la Figura B1 y Figura B2 se muestra el Datasheet del de las antenas VERT900

SPECIFICATION	
1. Description	: WSS016 Antenna With SMA(M)
2. Customer	: Ettus Research LLC
3. Model No.	: WSS016
4. Part No.	: RU1MPA0001A
5. Antenna Profile	: 236.5 mm (see Drawing)
6. Color	: Black
7. Electrical Characteristics	
Operating Frequency	: 824~960-1710~1990MHz
Antenna Type	: Dipole
Polarization Type	: Linear
Type of Radiation	: Toroidal
Impedance	: 50 Ohm nominal
V.S.W.R.	: 3.0:1 Max.
8. Mechanical Characteristics	
Swivel	: 90 degrees
Connector	: SMA(M)
Core	: N/A
9. Raw Material	
Coaxial Cable	: RG-316
Housing	: TPU
Hinge	: PC+ALLOY

Figura B1. Especificaciones Antenas VERT900

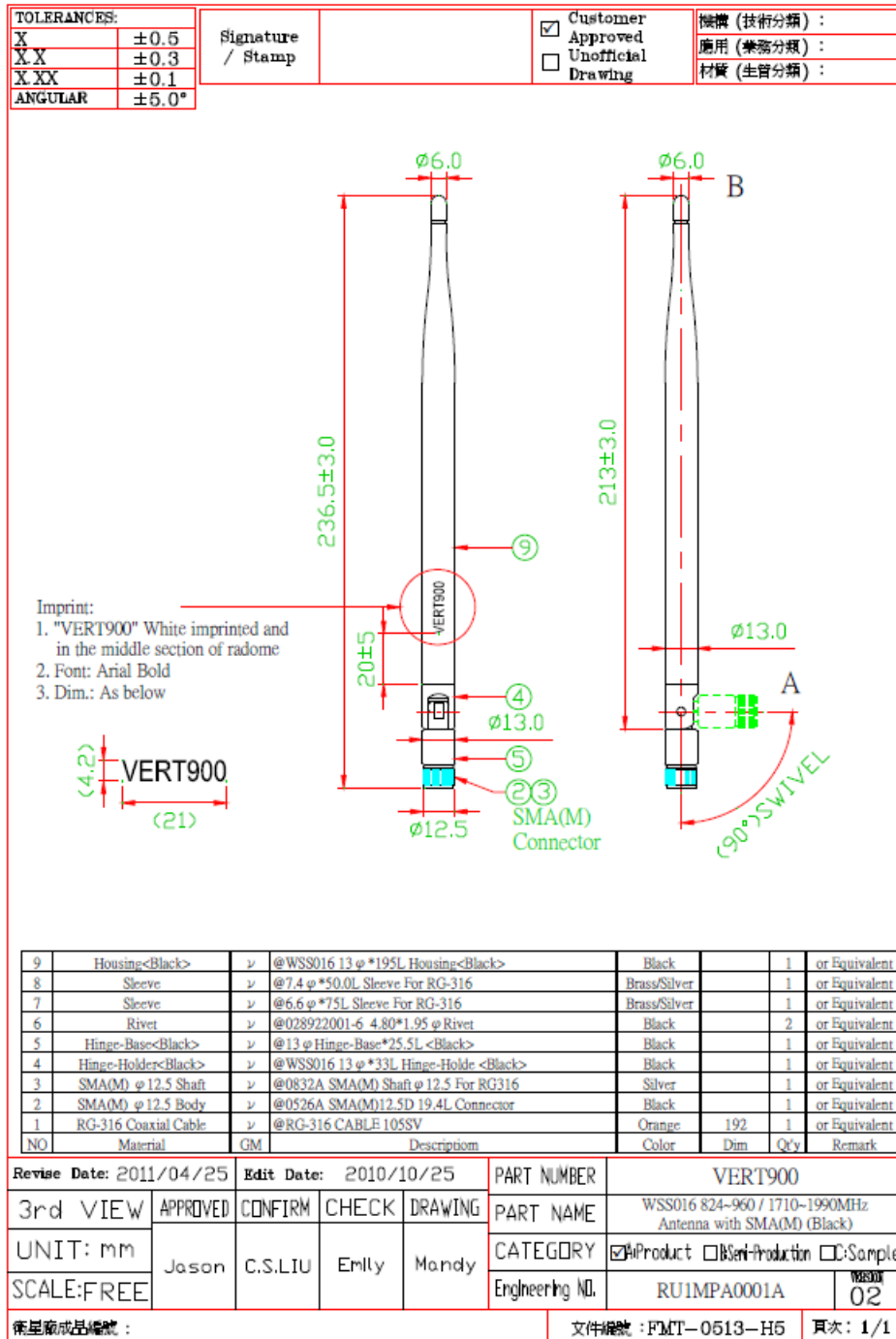


Figura B2. Dimensiones antenas VERT900