	GESTIÓN DE SERVICIOS ACADÉMICOS Y BIBLIOTECARIOS		CÓDIGO	FO-GS-15	
			VERSIÓN	02	
	ESQUEMA HOJA DE RESUMEN			FECHA	03/04/2017
				PÁGINA	1 de 1
ELABORO		REVISÓ		APROBÓ	
Jefe División de Biblioteca		Equipo Operativo de Calidad		Líder de Calidad	

RESUMEN TRABAJO DE GRADO

AUTOR(ES): NOMBRES Y APELLIDOS COMPLETOS

NOMBRE(S): ANGELICA ROCIO **APELLIDOS:** MEJIA SERRANO

FACULTAD: INGENIERÍA

PLAN DE ESTUDIOS: INGENIERÍA ELECTRÓNICA

DIRECTOR:

NOMBRE(S): KARLA CECILIA **APELLIDOS:** PUERTO LOPEZ

NOMBRE(S): LUIS EDUARDO **APELLIDOS:** RAMÍREZ CARVAJAL

TÍTULO DEL TRABAJO (TESIS): HERRAMIENTA COMPUTACIONAL PARA EL ANÁLISIS DE SISTEMAS DE COMUNICACIONES II UTILIZANDO LENGUAJE DE PROGRAMACIÓN PYTHON.

Se presenta una interfaz gráfica desarrollada con base en una metodología ágil enfocada al estudio de sistemas de comunicaciones digitales. La herramienta permite desarrollar procesos de códigos de línea y modulación digital. Se llevaron a cabo diferentes pruebas, ejecutando ejercicios para utilizar sus resultados como referencia y así, lograr un análisis cuantitativo y cualitativo entre la herramienta computacional y el análisis tradicional matemático. Estos resultados ayudan con el desarrollo del aprendizaje en las telecomunicaciones debido a que, su proceso de enseñanza se torna un poco compleja y al observar el proceso de códigos de línea o modulaciones digitales por medio de una interfaz gráfica el proceso para entender su teoría aporta dinamismo e intuitividad hacia la comunidad.

PALABRAS CLAVES: COMUNICACIONES, DIGITALES, MATLAB, PYTHON, MODULACIÓN DIGITAL, CODIGOS DE LÍNEA

CARACTERÍSTICAS:

PÁGINAS: 110 **PLANOS:** **CD ROOM:** **ILUSTRACIONES:** ____

HERRAMIENTA COMPUTACIONAL PARA EL ANÁLISIS DE SISTEMAS DE
COMUNICACIONES II UTILIZANDO LENGUAJE DE PROGRAMACIÓN PYTHON.

ANGELICA ROCIO MEJIA SERRANO

CÓDIGO: 1161231

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRICIDAD Y ELECTRÓNICA
PROGRAMA DE INGENIERÍA ELECTRONICA
SAN JOSÉ DE CÚCUTA

2022

HERRAMIENTA COMPUTACIONAL PARA EL ANÁLISIS DE SISTEMAS DE
COMUNICACIONES II UTILIZANDO LENGUAJE DE PROGRAMACIÓN PHYTON.

ANGELICA ROCIO MEJIA SERRANO

CÓDIGO: 1161231

DIRECTOR:

KARLA CECILIA PUERTO LOPEZ

M SC. Ingeniería en telecomunicaciones.

CODIRECTOR:

LUIS EDUARDO RAMÍREZ CARVAJAL

ING. Electrónico

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE ELECTRICIDAD Y ELECTRÓNICA

PROGRAMA DE INGENIERÍA ELECTRONICA

SAN JOSÉ DE CÚCUTA

2022

ACTA DE SUSTENTACIÓN DE UN TRABAJO DE GRADO

Fecha: CÚCUTA, 16 DE DICIEMBRE DE 2022

Hora: 08:00

Lugar: SALON SC 301

Plan de Estudios: INGENIERÍA ELECTRÓNICA

Título de la Tesis: "HERRAMIENTA COMPUTACIONAL PARA EL ANÁLISIS DE SISTEMAS DE COMUNICACIONES II UTILIZANDO LENGUAJE DE PROGRAMACIÓN PYTHON". Modalidad trabajo de investigación.


Jurados: IE MSc. ELMER PARADA PRIETO
IE MSc. ANDRES EDUARDO PAEZ PEÑA


Director: IE. MSc. KARLA CECILIA PUERTO LOPEZ

Codirector: IE. LUIS EDUARDO RAMÍREZ CARVAJAL

Nombre del Estudiante:	Código:	Calificación:	
		Número	Letra
ANGÉLICA ROCÍO MEJÍA SERRANO	1161231	4,4	Cuatro, cuatro

APROBADA


ELMER PARADA PRIETO


ANDRES EDUARDO PAEZ PEÑA


SERGIO SEPÚLVEDA MORA
Coordinador Comité Curricular
Ingeniería Electrónica

AGRADECIMIENTOS

Gracias primeramente a Dios por su fidelidad en toda esta trayectoria para conseguir el título Universitario. Gracias a mis padres Luis Angel Mejia Cortes y Aura Rocio Serrano Silva que siempre demostraron su apoyo incondicional en cada una de las decisiones de mi proyecto de vida.

Gracias a mi amiga Leidy Marcela Castaño Carvajal y a su familia por estar presentes, por sus consejos y su gran cariño y su compañía en el momento que más lo necesité. A mi amiga Angie Tatiana Gonzales Pinto que llegó a mi vida de la manera menos esperada. Gracias porque han creído en mí. A mis amigos y profesores de la facultad, muchas gracias por cada aporte que hicieron para mi vida tanto académica como profesional.

A mi directora de tesis la Ingeniera Karla Cecilia Puerto López y a mi Codirector el Ingeniero Luis Eduardo Ramírez Carajal, quienes admiro por su maravilloso trabajo como profesionales y fueron un apoyo incondicional en el desarrollo de este proyecto, especialmente en el contexto de la divulgación. Agradezco al Ingeniero Javier Portilla quien dio un gran aporte de investigación en el transcurso de este proyecto. Al ingeniero Dinael Guevara Ibarra, director del grupo de investigación y desarrollo en Electrónica y Telecomunicaciones (GIDET) de quien también recibí gran apoyo de manera diligente y amable. Gracias a mi querida alma mater, la Universidad Francisco de Paula Santander, por brindarme la oportunidad de formarme como persona y especialmente como una profesional integra.

Angelica Rocio Mejia Serrano

TABLA DE CONTENIDO

Introducción	15
1. Título	16
2. Planteamiento del problema	17
3. Justificación	20
3.1 Beneficios Tecnológicos	20
3.2 Beneficios Económicos	20
3.3 Beneficios Sociales	21
4. Alcances	22
4.1 Tipo de proyecto	22
5. Limitaciones y Delimitaciones	23
5.1 Limitaciones	23
5.2 Delimitaciones	23
6. Objetivo general	24
7. Objetivos Específicos	24
8. Marco referencial	25
8.1 Antecedentes	25
8.2 Marco teórico	27
8.2.1 Sistemas de Comunicaciones Digitales	27

8.2.2	Matlab para el estudio de Sistemas de Comunicaciones Digitales	27
8.2.3	Lenguaje de programación Python	28
8.2.4	Spyder IDE	29
8.2.4	Matplotlib	29
8.2.5	GNU Radio	30
8.3	Marco Legal	30
9.	Diseño Metodológico	33
9.1	Objetivo 1: Revisar el estado del arte	33
9.1.1	Conocimientos Previos sobre Python	33
9.1.2	Búsqueda de Referencias	33
9.1.3	Análisis Preliminar	33
9.2	Objetivo 2: Realizar el algoritmo	34
9.2.2	Representar el método mediante un diagrama de flujo	34
9.3	Objetivo 3: Codificar el algoritmo utilizando	34
9.3.1	Elección del Entorno de desarrollo	34
9.3.2	Escritura del Código de programación de Python	35
9.4	Objetivo 4: Evaluar el funcionamiento del algoritmo	35
9.4.1	Funcionamiento del algoritmo	35
9.4.2	Implementar algunas pruebas simuladas	35
10.	Cronograma de Actividades	37

11.	Presupuesto	38
11.1	Gasto Global	38
11.2	Presupuesto del Personal	38
11.3	Gastos de Equipo	39
11.4	Gastos de Materiales e Insumos	39
11.5	Gastos de Software	40
12.	Resultados	41
12.1	REVISAR EL ESTADO DEL ARTE	41
12.1.1	Conocimientos Previos de Python	41
12.1.2	Búsqueda de Referencias	43
12.1.3	Análisis Preliminar	44
12.2	REALIZAR EL ALGORITMO	46
12.2.1	Crear un método de análisis	46
12.3	CODIFICAR EL ALGORITMO	54
12.3.1	Elección del entorno de desarrollo	54
12.4	EVALUAR EL FUNCIONAMIENTO DEL ALGORITMO	64
12.4.1	Funcionamiento del Algoritmo	64
12.4.2	Implementar algunas pruebas	71
12.5	Productos	90
12.5.1	Guías de laboratorio	90

12.5.2	Divulgaciones	94
13.	Conclusiones	98
14.	Recomendaciones y trabajo futuro	99
15.	Referencias	100
16.	Anexos	104

Lista de figuras

Figura 1 Logo de Python	29
Figura 2 Logo de Spyder	29
Figura 3 Logo de Matplotlib	30
Figura 4 Diagrama de metodología usada en el proyecto	41
Figura 5 Entrada Binaria, Entrada Analógica	47
Figura 6 Imágenes tomadas del capítulo 12 del libro de Tomasi	48
Figura 7 Representación de un transmisor BPSK – imagen de Tomasi	49
Figura 8 Relación en función del tiempo para una modulador BPSK (Tomasi)	49
Figura 9 Fase de salida en función del tiempo para un modulador QPSK	50
Figura 10 Figura tomada del libro de la universidad cataluya	51
Figura 11 Modulaciones NRZ y RZ	52
Figura 12 Modulación AMI	52
Figura 13 Ejemplos de códigos de línea tomasi	53
Figura 14 Diagrama de flujo	54
Figura 15 Ingreso de datos para modulación digital	60
Figura 16 Modulación por desplazamiento cuaternario de fase	60
Figura 17 Modulación por desplazamiento de frecuencia binario	60
Figura 18 AMI RZ – AMI NRZ	61
Figura 19 Bipolar RZ – Bipolar NRZ	62
Figura 20 Unipolar RZ – Unipolar NRZ	62
Ilustración 21 Unipolar RZ – Unipolar NRZ	63

Figura 22 Mensaje generado por la interfaz	63
Figura 23 AMI RZ Python	64
Figura 24 AMI RZ Matlab	64
Figura 25 AMI NRZ Matlab	65
Figura 26 Python	65
Figura 27 BRZ Matlab	65
Figura 28 BRZ Python	65
Ilustración 29 Manchester Python	66
Figura 30 Manchester Matlab	66
Figura 31 URZ Matlab	67
Figura 32 URZ Python	67
Figura 33 UNRZ Matlab	67
Figura 34 UNRZ Python	67
Figura 35 BFSK Matlab	68
Figura 36 BFSK Python	68
Figura 37 BPSK Matlab	69
Figura 38 BPSK Matlab	69
Figura 39 QPSK Matlab	70
Ilustración 40 QPSK Python	70
Figura 41 Prueba AMI RZ 1010100110	71
Figura 42 Prueba AMI RZ 1011001010	72
Figura 43 Prueba AMI RZ 1101100010	72
Figura 44 Prueba AMI RZ 1101010001	73

Figura 45 Prueba AMI NRZ 1010100110	73
Figura 46 Prueba AMI NRZ 1011001010	74
Figura 47 Prueba AMI NRZ 1101100010	74
Figura 48 Prueba AMI NRZ 1101010001	75
Figura 49 BRZ Prueba BRZ 1010100110	75
Figura 50 Prueba BRZ 1011001010	76
Figura 51 Prueba BRZ 1101100010	76
Figura 52 Prueba BRZ 1101010001	77
Figura 53 Prueba Manchester 1010100110	77
Figura 54 Prueba Manchester 1011001010	78
Figura 55 Prueba Manchester 1101100010	78
Figura 56 Prueba Manchester 1101010001	79
Figura 57 Prueba URZ 1010100110	79
Figura 58 Prueba URZ 1011001010	80
Figura 59 Prueba URZ 1101100010	80
Figura 60 Prueba URZ 1101010001	81
Figura 61 Prueba UNRZ 1010100110	81
Figura 62 Prueba UNRZ 1011001010	82
Figura 63 Prueba UNRZ 1101100010	82
Figura 64 Prueba UNRZ 1101010001	83
Figura 65 Prueba BFSK 1011001010	84
Figura 66 Prueba BFSK 1101100010	84
Figura 67 Prueba BFSK 1101010001	85

Figura 68 Prueba BPSK 1011001010	86
Figura 69 Prueba BPSK 1101100010	86
Figura 70 Prueba BPSK 1101010001	87
Figura 71 Prueba QPSK 1010100110	88
Figura 72 Prueba QPSK 1011001010	88
Figura 73 Prueba QPSK 1101100010	89
Figura 74 Prueba QPSK 1101010001	89
Figura 75 Guía de laboratorio	90
Figura 76 Guía de laboratorio	91
Figura 77 Guía de laboratorio	92
Figura 78 Guía de laboratorio	93
Figura 79 Guía de laboratorio	94
Figura 80 Certificado Potencia 1	95
Figura 81 Certificado Ponencia 2	96
Figura 82 XXV Encuentro Nacional REDCOLSI	97

Lista de tablas

Tabla 1 Cronograma de Actividades	37
Tabla 2 Presupuesto global del proyecto (en miles \$)	38
Tabla 3 Descripción de los gastos de personal (en miles \$)	38
Tabla 4 Descripción de los gastos de equipo (en miles de \$)	39
Tabla 5 Descripción de los gastos de materiales e insumos (en miles de \$)	39
Tabla 6 Descripción de los gastos de software (en miles de \$)	40
Tabla 7 Datos Fase de Salida	50
Tabla 8 Elaboración propia	55

Introducción

El estudio de las Telecomunicaciones ha permitido entre los individuos facilitar la comunicación dando acceso a servicios que antes se encontraban fuera de su alcance y es por esto que tienen gran relevancia en la vida de las personas y en las actividades productivas de los países [1]. Por ende, en el transcurso del tiempo los sistemas de comunicaciones han evolucionado con respecto a la manera de transmitir información como el telégrafo, el descubrimiento del espectro electromagnético y radioeléctrico y sistemas guiados más robustos como el cable coaxial [2]. Las telecomunicaciones tienen diversos temas en su estudio como lo son los códigos de líneas que son también llamados Modulaciones digitales en banda base [3] y las modulaciones digitales [4].

La forma de enseñanza que se ha tenido en las Telecomunicaciones se ha considerado en el paradigma presencial, pero en la actualidad se deben tener en cuenta características como el entorno social, tecnológico, las cualidades de los estudiantes, mecanismos de interacción, etc; para así, poder brindar en la actualidad un medio por el cual los métodos de enseñanza se adapten al ámbito de la virtualidad [5].

Las herramientas virtuales producen un gran efecto en la educación, incentivando al estudiante y a los docentes a darse la oportunidad de conocer nuevos caminos en vista de que esto produce el manejo de diversos recursos tecnológicos en donde se pueden seguir planteando nuevos objetivos. De acuerdo con esto [6], se presenta una herramienta computacional para el estudio de códigos de línea y modulación digital.

1. Título

HERRAMIENTA COMPUTACIONAL PARA EL ANÁLISIS DE SISTEMAS DE COMUNICACIONES II UTILIZANDO LENGUAJE DE PROGRAMACIÓN PHYTON.

2. Planteamiento del problema

La Universidad Francisco de Paula Santander cuenta con la acreditación de alta calidad, lo cual le ha permitido consolidarse como una institución reconocida a nivel regional y nacional. Es por esto que los programas de estudio deben estar en la búsqueda constante de mejorar las tecnologías y métodos de enseñanza. Por este motivo, las dinámicas educativas impulsan fundamentalmente a la formación de los estudiantes al servicio de una sociedad más competitiva y suficientemente capacitada en las Tecnologías de la Información y la Comunicación (TIC) [7].

Por lo anterior, un ejemplo claro se puede observar en el programa de Ingeniería Electrónica que se encuentra actualmente acreditado, sin embargo, debe continuar en la mejora de sus tecnologías y para ello se requiere incentivar la filosofía del software libre que permite ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el conocimiento en sus diferentes dimensiones científicas y culturales mediante un lenguaje informático [8]. Esto permite al usuario (personas, organizadores, compañías, etc.) contemplar determinadas libertades como la libertad de ejecutar el programa, estudiarlo, comprender su funcionamiento y establecer copias para la modificación que futuros usuarios quieran agregar al programa y dejar esto a disposición del público para adaptarlo a cada necesidad [9].

Teniendo en cuenta que en los procesos de mejoramiento continuo y en pro de la reacreditación, el programa de Ingeniería Electrónica apoya a los estudiantes, profesores, semilleros y grupos de investigación al avance y manejo de diversos software y hardware para el entorno académico y práctico en la solución de problemas a nivel regional y nacional. De esta

manera, se abre paso hacia el entorno de la asignatura Sistemas de Comunicaciones II, la cual se desarrolla utilizando Matlab. Este es un programa lenguaje de programación es cerrado, su adquisición e instalación es complicada y esto se convierte en una problemática para los estudiantes que cuentan con bajos recursos en las tecnologías. Existen varios entornos de software, hardware de códigos abiertos y fáciles de adquirir que se podrían usar como lo son Arduino, GNU Radio, Linux, Python, entre otros.

La asignatura de sistemas de comunicaciones II suministra conocimientos en transmisión digital (Modulación PAM, Modulación PCM, PCM Diferencial), Modulación digital (FSK Binaria, M-arias, PSK, M-PSK, M-QAM), los anchos de bandas de estas modulaciones, radio digital, técnicas de multiplexación y la Modulación espectro-espectro, entre otros temas. Teniendo en cuenta esto, la programación en Python facilita el desarrollo de gráficas de este estilo permitiendo visualizar las características que se requieren (como la amplitud, la frecuencia, los diagramas de constelación, ruido e interferencias, etc), el cálculo de parámetros de los sistemas digitales, logrando satisfacer estas necesidades y replicar los procedimientos experimentales aplicados conforme a los apartados teóricos en telecomunicaciones debido a que esta tecnología puede ser implementada para la ejecución de las diferentes prácticas de laboratorios mediante un lenguaje de programación más ligero y práctico para el estudiante.

Con lo anterior, se puede destacar que en Python una de las principales ventajas es su lenguaje de programación moderno, el cual se puede obtener de manera gratuita por internet y en vista de esto, da seguridad al usuario para realizar sus proyectos e investigaciones sin necesidad de preocuparse por la adquisición de la licencia del software [10]. Actualmente en la asignatura

de Sistemas de Comunicaciones II no se han realizado prácticas, guías de laboratorio o proyectos académicos en el área de las Telecomunicaciones en donde se use el lenguaje de código abierto como Python. Teniendo en cuenta lo anterior surge la siguiente pregunta:

¿Cómo desarrollar una herramienta computacional para el estudio y análisis de Sistemas de Comunicaciones II utilizando lenguaje de programación Python?

3. Justificación

En vista del problema planteado, fue necesario crear una herramienta computacional que tiene los siguientes beneficios:

Teniendo en cuenta los beneficios, con la herramienta desarrollada, estudiantes, docentes e investigadores contarán con un elemento importante para un mejor desarrollo académico e investigativo, máxime en la crisis actual provocada por el covid-19, en la cual se es necesario contar con herramientas que apoyen el desarrollo académico de manera virtual.

3.1 Beneficios Tecnológicos

Se llevó a cabo el desarrollo de una herramienta computacional de software libre que permitirá un excelente aprendizaje en la asignatura de Sistemas de Comunicaciones II que contiene una interfaz gráfica interactiva.

Contribuye a la formación de profesionales en nuevas tecnologías y estimula a que la institución, por medio de los estudiantes, logre beneficiarse en cada avance e innovación que se esté realizando.

3.2 Beneficios Económicos

En el transcurso de su creación, no se incurrieron en gastos de licencia.

El desarrollo de esta herramienta virtual de software libre permite un mayor acceso a todos los estudiantes de la asignatura teniendo en cuenta que al ser gratuita no se requiere de la licencia de la universidad para poder utilizarla y los estudiantes pueden aprovechar este software en sus casas o espacios libres.

Al emplear el software libre, no se incurrieron en costos al momento de realizar ajustes debido a actualizaciones de sistemas operativos.

3.3 Beneficios Sociales

Al permitir trabajar estándares abiertos con distintos sistemas, facilita implementar servicios electrónicos y da lugar a un acceso seguro para los estudiantes obteniendo un impacto positivo ante la comunidad universitaria incentivando de esta manera a la investigación y desarrollo de software en diferentes áreas de la ingeniería para la solución ante las problemáticas que se encuentren actualmente.

4. Alcances

La herramienta es un proyecto tipo descriptivo que permite que los estudiantes de Sistemas de Comunicaciones II puedan realizar una mejora en las simulaciones de sus prácticas de laboratorio por medio de un diseño de software libre.

4.1 Tipo de proyecto

El proyecto fue de carácter investigativo debido a que se fundamenta en el uso de Python que busca el desempeño como lenguaje de programación y realizar aplicaciones en el área de sistemas de comunicaciones digitales. Por este modo, es una investigación aplicada donde se diseñaron y evaluaron proyectos o prácticas de laboratorio para los estudiantes de la asignatura Sistemas de Comunicaciones II del programa de Ingeniería Electrónica.

5. Limitaciones y Delimitaciones

5.1 Limitaciones

El proyecto se realizó mediante un entorno de desarrollo integrado (IDE) y esto se encontró limitado a las características que dispone el programa al momento de implementar los temas pertenecientes a sistemas de comunicaciones digitales.

En este proyecto de investigación se implementó software libre, bibliotecas o cajas de herramientas para producir guías de laboratorio a las estudiantes del programa de ingeniería electrónica de la asignatura sistemas de comunicaciones digitales.

5.2 Delimitaciones

Espacial: el proyecto se realizó dentro de las limitaciones académicas, las pruebas tuvieron lugar en los laboratorios de electrónica de la universidad Francisco de Paula Santander en su sede central y la comuna dos de la ciudad de San José de Cúcuta, por ser la comuna en la que se encuentra ubicado el barrio Colsag.

Temporal: El proyecto tuvo una duración de 12 meses

Conceptual: Se delimitó a Sistemas de comunicaciones digitales, procesamiento digital de señales, herramienta computacional educativa.

6. Objetivo general

Desarrollar una herramienta educativa para el aprendizaje de sistemas de comunicaciones II utilizando software libre.

7. Objetivos Específicos

- 7.1** Revisar el estado del arte con respecto al uso del software libre, su funcionamiento y aplicación de proyectos.
- 7.2** Realizar el algoritmo que establece el funcionamiento de la herramienta computacional
- 7.3** Codificar el algoritmo utilizando Python mediante un entorno de desarrollo que permita crear una interfaz para la herramienta.
- 7.4** Evaluar el funcionamiento del algoritmo por medio de prácticas de laboratorio para la asignatura Sistemas de Comunicaciones II.

8. Marco referencial

8.1 Antecedentes

Inicialmente, Flavio Morales Arévalo y Jairo Wladmir Congo Pastrana en “Aplicaciones del Software Libre Python para Prácticas de Laboratorio Aplicado a la Asignatura de Tratamiento Digital de Señales de la Universidad Tecnológica Israel”. En este proyecto hacen énfasis en el empleo de software libre Python con el fin de presentar una alternativa actualizada de las herramientas necesarias para un profesional en Telecomunicaciones dividiéndolo en 5 etapas: la primera trata promover Python como software para crear un modelo de laboratorio de Tratamiento Digital de Señales, la segunda parte contiene los aspectos conceptuales necesarios para interpretar las señales y los procesos matemáticos que deben emplearse en los circuitos que funcionan usando algoritmos para tratar señales, la tercera parte trata sobre los aspectos sugeridos por el software para emplearse en el laboratorio mostrando características como la interfaz gráfica, la cuarta parte del proyecto contiene el formato de las prácticas del laboratorio y la quinta parte contiene las conclusiones y recomendaciones que sugieren conforme al proyecto [11].

Además, Huitrón Mendoza y José Antonio en “Método para la enseñanza de software libre en ciencias socioeconómicas”. En este artículo se presenta un método de enseñanza-aprendizaje utilizando la tecnología de software libre para alumnos del área de las ciencias socioeconómicas. Esta herramienta computacional es llevada a cabo para estudiantes que no están especializados en la programación en vista de que este método favorece en ayudar a fomentar dichas

habilidades compartiendo sus conocimientos en el aula o en un espectro más amplio como realizando sus resultados fundamentales por medio de internet [12].

Consecutivamente, Amaury Pérez Torres, en su artículo “Criterios Considerados en la selección de software libre para la enseñanza y el aprendizaje de conceptos de computación gráfica”. En este artículo tratan el tema del software no libre y del software libre. Aplican criterios para la selección de software libre que sustituyen los programas no libres teniendo en cuenta criterios didácticos, la posibilidad de que soporte los criterios teóricos fundamentados en la asignatura, que su licencia gratuita evita gastos para la institución de manera que no se ocurran ilegalidades y también la facilidad de las instalaciones de las herramientas [13].

Seguidamente, Nilo Alfredo Poveda Cisneros en su proyecto de grado “La enseñanza de matemáticas en la recuperación pedagógica fundamentada en los tics de software sobre, aplicado en talleres interactivos de radiación y potenciación”. En este proyecto hablan sobre las actividades de recuperación pedagógica utilizando los tics como software libre en donde se procura ampliar y mejorar las condiciones del proceso de aprendizaje y optimizar la calidad de la educación. Por medio del uso del software libre se evidenció que produce un buen rendimiento de recuperación pedagógica en el aprendizaje causando curiosidad en las acciones avanzadas en el salón de clases [14].

Finalmente, Daniela Cepeda y Jorge Bacca en “Aplicación móvil para la enseñanza de la programación sobre el lenguaje Python”. Este artículo trata sobre una investigación introduciendo una aplicación móvil para la enseñanza de la programación de Python. Se explica

una metodología desarrollada en la aplicación móvil propuesta y se expresan los trabajos seleccionados que han servido de inspiración en donde se permiten tener fundamentos teóricos y prácticos. Con base a todo esto, se logra establecer el desarrollo de la aplicación, es decir, el entorno gráfico, el proceso de enseñanza, entre otras cosas, presentando de esta manera los resultados obtenidos de la aplicación [15].

8.2 Marco teórico

8.2.1 Sistemas de Comunicaciones Digitales

Es importante comprender primero que todo, que las señales que deseamos comunicar están originalmente en formato analógico. Un ejemplo puede ser la señal de voz que es captada por el micrófono de un teléfono móvil de primera generación y que, modulada en FM, se transmite por radio hasta la estación base. Por lo tanto, las comunicaciones digitales, en busca de una solución económica, permite la incorporación de terminales más baratos y pequeños que emplean tratamientos digitales que dotan a los sistemas de nuevas funcionalidades, entre los que destacan la flexibilidad. Los sistemas digitales permiten construir fácilmente mecanismos de multiplexación temporal y encaminamiento, que resultarían muy complejos con sistemas analógicos [16].

8.2.2 Matlab para el estudio de Sistemas de Comunicaciones Digitales

Matlab es un sistema interactivo cuyo elemento básico de datos es una matriz que no requiere dimensiones y esto permite resolver problemas numéricos en una fracción de tiempo; todo esto ha evolucionado en los últimos años en colaboración de gran variedad de usuarios. En entornos universitarios, Matlab se ha convertido en una herramienta de enseñanza estándar para cursos de introducción como para cursos avanzados. El entorno de Matlab se extiende para resolver clases particulares de problemas como: Procesamiento de señales, Diseño de sistemas de control, Simulación de sistemas dinámicos, Identificación de sistemas, Redes neuronales y otros [17].

La aparición de plataformas hardware tipo Arduino y Raspberry Pi favorece el desarrollo de entornos de prácticas que pueden ser utilizadas en casi todos los entornos. Estos tipos de dispositivos pueden ser programados en sus propios entornos o bien usando entornos de generación automáticos de código como los incorporados en Matlab/Simulink [18]. Simulink es un elemento que se encuentra incorporado en Matlab y, utilizando Matlab y Simulink de manera conjunta, en realidad se está realizando una combinación tanto textual como gráfica para diseñar un sistema en un entorno de simulación.

8.2.3 Lenguaje de programación Python

Python es un lenguaje de programación que interpreta fundamentalmente la legibilidad del código. También es un lenguaje en donde conviene de forma nativa aspectos imperativos, funcionales y orientada a objetos [19].



Figura 1 Logo de Python

Python cuenta con herramientas que permiten el buen desarrollo, estudio y análisis de los sistemas de comunicaciones digitales. Estas herramientas permiten graficar con precisión, estudiar la modulación, desarrollar figuras interactivas y realizar análisis en un medio computacional. Estas herramientas son tales como Spyder, Matplotlib y GNU Radio.

8.2.4 Spyder IDE

Spyder (anteriormente Pydee) es un potente entorno de desarrollo integrado, multiplataforma de código abierto (IDE) y un entorno interactivo para el lenguaje Python. Posee funciones avanzadas de edición, pruebas interactivas, depuración e introspección y un entorno informático numérico [20].



Figura 2 Logo de Spyder

8.2.4 Matplotlib

Matplotlib es un proyecto que permite desarrollar gráficos, figuras interactivas, permite el funcionamiento personalizado proporcionado por paquetes de terceros. Es una biblioteca para crear visualizaciones estáticas, animadas e interactivas en Python [21].



Figura 3 Logo de Matplotlib

8.2.5 GNU Radio

GNU Radio es una herramienta de desarrollo tipo GLP (General Public License), esto permite que el procesamiento, la construcción y simulación de señales se logre a nivel de simulación o a nivel de implementación on la ayuda de radios programables cuyo trabajo es realizar esto en radio frecuencia mientras el procesamiento en banda base de la señal se realiza mediante software [22].

8.3 Marco Legal

En primer lugar, se encuentra la Ley No. 1341 del 30 de Julio de 2009 en la cual se definen principios y conceptos sobre la sociedad de la información y la organización de las tecnologías de la información y las comunicaciones (TIC).

En el Artículo 2 habla sobre los principios orientados en donde aclaran que la investigación y el desarrollo de las Tecnologías de la Información y las Comunicaciones son una política del estado que involucra todos los sectores públicos y de la sociedad para contribuir con el desarrollo educativo, cultural, económico, social y político de tal manera que involucren la competitividad y el respeto a las personas. Los principios orientados de esta ley son: 1. Prioridad al acceso y uso de las Tecnologías de la Información y las Comunicaciones: Todos los sectores en general deberán colaborar, dentro del marco de sus obligaciones, para priorizar el acceso y uso de las TIC's en la producción de bienes y servicios, en condiciones no discriminatorias en la conectividad, la educación, los contenidos y la conectividad. 2. Libre Competencia: Esto significa que el estado creará escenarios libres y leales que ayuden a incentivar la inversión actual y futura en el sector de las TIC's, que se logre involucrar en el mercado de carácter de libre competencia, bajos precios y en condiciones de igualdad, sin dar oportunidad de otorgar condiciones ni privilegios distintos dentro del mercado, priorizando una sana competencia. 3. Protección de los derechos de los usuarios: El Estado velará por la adecuada protección de los derechos de los usuarios de las Tecnologías de la Información y de las Comunicaciones, así como por el cumplimiento de los derechos y deberes derivados del Habeas Data, asociados a la prestación del servicio. 4. El Derecho a la comunicación, la información y la educación y los servicios básicos de las TIC: Según el desarrollo de los artículos 20 y 67 de la constitución Nacional, el estado dará prioridad a todo colombiano, el derecho al acceso de las Tecnologías de la Información y de las Comunicaciones básicas, dando lugar a la libertad de expresión y de difundir sus pensamientos y opiniones, la de informar y recibir información verás e imparcial, la educación y el acceso al conocimiento, a la ciencia, a la técnica, y a los demás bienes y valores de la cultura. [23].

Por otra parte, el marco legal del software libre cuenta con la Ley 11723 que está compuesta por 89 artículos conocida como "Ley de Propiedad Intelectual" o también como "Ley de Propiedad Científica, Literaria y Artística". Su última reforma data de noviembre de 1998, cuando por Ley 25036 se le introdujeron modificaciones referidas al software, para darle fin a las discusiones doctrinarias y jurisprudenciales sobre la cuestión de si el software estaba o no bajo el amparo de esta ley. En su Artículo 1 establece que las obras científicas, literarias y artísticas comprenden los escritos de toda naturaleza y extensión, entre ellos los programas de computación fuente y objeto; las compilaciones de datos o de otros materiales. Esto establece la prioridad de usar software libre en todas las dependencias de la Administración Pública Nacional, salvo excepciones. Entre estos aspectos se destaca el económico (por el costo de la licencia y la libertad que otorga el software libre de copiar), el moral (es conocido en todos los ámbitos la descarga de manera ilegal del software, ya sea por cuestiones del costo de la licencia, negligencia, etc.; el cual coloca al Estado como uno de los principales infractores a la ley 11723), el cultural, el educativo, el de seguridad nacional, etc. [24].

9. Diseño Metodológico

A continuación, se presentan los procesos que se realizarán para el cumplimiento de los objetivos específicos planteados.

9.1 Objetivo 1: Revisar el estado del arte con respecto al uso del software libre, su funcionamiento y aplicación de proyectos.

9.1.1 Conocimientos Previos sobre Python

Metodología: Se buscó información relevante por medio de artículos, libros, informes técnicos u otras publicaciones sobre elementos importantes relacionados con Python

9.1.2 Búsqueda de Referencias

Metodología: Se investigó sobre proyectos realizados desde el año 2017 en donde se usó Python como parte de su desarrollo por medio de conferencias, páginas web oficiales y revistas científicas.

9.1.3 Análisis Preliminar

Metodología: Se analizó el funcionamiento de las diferentes herramientas de Python y la forma en la que éstas podrían aplicarse al proyecto.

9.2 Objetivo 2: Realizar el algoritmo que establece el funcionamiento de la herramienta computacional

9.2.1 Crear una metodología de análisis

Metodología: Se Realizó un método de análisis que incluye los temas de la asignatura y con base en el algoritmo de la herramienta.

9.2.2 Representar el método mediante un diagrama de flujo

Metodología: Se diseñó un algoritmo que representa el proyecto por medio de un diagrama de flujo, el cual contiene las aclaraciones del diseño a través de pasos sucesivos de una manera precisa y ordenada.

9.3 Objetivo 3: Codificar el algoritmo utilizando Python mediante un entorno de desarrollo que permita crear una interfaz para la herramienta.

9.3.1 Elección del Entorno de desarrollo

Metodología: Se investigó sobre los diferentes entornos de desarrollo de Python para elegir el más adecuado que permite crear una interfaz interactiva a la herramienta.

9.3.2 Escritura del Código de programación de Python

Metodología: Se procedió a escribir el código de programación en Python con base al diagrama de flujo elaborado.

9.4 Objetivo 4: Evaluar el funcionamiento del algoritmo por medio de prácticas de laboratorio para la asignatura Sistemas de Comunicaciones II.

9.4.1 Funcionamiento del algoritmo

Metodología Se Realizan las pruebas del diseño por medio de ejemplos de temáticas abordadas utilizando Matlab y la herramienta desarrollada para llevar a cabo un análisis comparativo de carácter cualitativo y cuantitativo, calculando el error promedio de algunas variables entre las dos herramientas.

9.4.2 Implementar algunas pruebas simuladas

Metodología: Se toman de forma aleatoria algunos ejercicios o diseños de libros que logran explicar este campo y se implementan para posteriormente evaluar la precisión obtenida a través de los datos simulados.

11. Presupuesto

11.1 Gasto Global

Tabla 2 Presupuesto global del proyecto (en miles \$)

RUBROS	FUENTES				TOTAL
	ESTUDIANTES		UFPS		
	Efectivo	Especie	Efectivo	Especie	
Personal	0	5600	0	4800	10400
Equipo	0	1000	0	250	1250
Materiales y suministros	0	31	0	441	472
Software	0	0	0	0	150000
Imprevistos	0	663,1	0	549,1	1212,1
TOTAL	0	7294,1	0	6040,1	163334,2

11.2 Presupuesto del Personal

Tabla 3 Descripción de los gastos de personal (en miles \$)

PERSONAL	FUNCIÓN DENTRO DEL PROYECTO	DEDICACIÓN HORAS/ SEMANA	\$/h	# de meses	FUENTES		TOTAL
					Estudiantes	UFPS	
Angelica Rocio Mejia Serrano	Tesista	35	5	8	5600	0	5600
Karla Cecilia Puerto López	Director	4	25	8	0	800	3200
Luis Eduardo Ramírez Carvajal	Codirector	2	25	8	0	800	1600
TOTAL					56400	4800	10400

11.3 Gastos de Equipo

Tabla 4 Descripción de los gastos de equipo (en miles de \$)

DESCRIPCIÓN	CANTIDAD	JUSTIFICACIÓN	FUENTES		TOTAL
			Estudiantes	UFPS	
Computador	1	Necesario para investigación, simulación y desarrollo del proyecto.	1000	0	1000
Impresora	1	Necesario para la impresión de los documentos de investigación.	0	250	250
TOTAL			1000	250	1250

11.4 Gastos de Materiales e Insumos

Tabla 5 Descripción de los gastos de materiales e insumos (en miles de \$)

DESCRIPCIÓN	CANTIDAD	VALOR UNIDAD	FUENTES		TOTAL
			Estudiantes	UFPS	
Internet (horas)	315	1.4.	0	441	441
Resma de papel tamaño carta	1	1.5	15	0	15
Tinta	2	5	10	0	10
Carpetas	6	1	6	0	6
TOTAL			31	4411	472

11.5 Gastos de Software

Tabla 6 Descripción de los gastos de software (en miles de \$)

SOFTWARE	CANTIDAD	VALOR UNIDAD	FUENTES		TOTAL
			Estudiantes	UFPS	
Matlab	1	0	0	0	0
LabVIEW	1	0	0	0	0
Web of Science	1	0	0	150000	0
TOTAL			0	0	150000

12. Resultados

En este capítulo, como se muestra en la figura 4, se detalla cada uno de los resultados en cada actividad realizada para poder cumplir con el objetivo en el que se encuentran implícitas. Se documentó información previa y pruebas realizadas para determinar la validez de la herramienta computacional desarrollada.

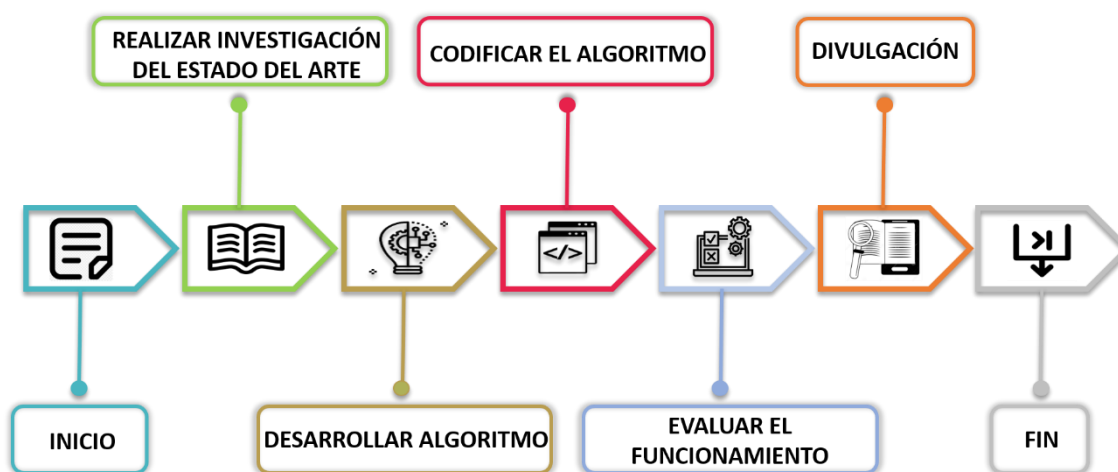


Figura 4 Diagrama de metodología usada en el proyecto

12.1 REVISAR EL ESTADO DEL ARTE

12.1.1 Conocimientos Previos de Python

Torres Gómez, Camilo y Torres Sandoval, Jeyson Steven en su proyecto de “Seguridad en Redes con Python”, tiene como finalidad identificar, proyectar y promover la utilidad y eficiencia de esta herramienta, encontrando los principales beneficios de la misma, y demostrar por qué este elemento es vital para ciberseguridad hoy en día. Por otra parte, se hicieron demostraciones del funcionamiento de Python por medio de videos donde se explica paso a paso

uno de las tantas utilidades que puede tener este complemento informático en un hogar o en una empresa [25].

Jesús Alberto Díaz Navarro, Mauricio Ricaurte Vargas y Miryam Johanna Salcedo Ramírez en su tesis de pregrado titulado como “Fases de diseño de arquitectura, construcción de Backend y pruebas para la migración al lenguaje de programación Python del proyecto “web cliente” en una compañía de asistencias”, tiene como finalidad realizar una migración de un desarrollo ejecutado en un lenguaje de programación C Sharp bajo el framework .Net el cual es un lenguaje de programación de propiedad de Microsoft que se desarrolla solo en sistemas operativos licenciados (Windows), a un lenguaje de programación conocido como Python, el cual es desarrollado bajo sistemas operativos libres (GNU Linux, Ubuntu, Debian, entre otros) [26].

Flores Pérez, José en su proyecto de grado titulado “Detector de escenas acústicas en Python”, consiste en la creación de un detector de escenas acústicas en lenguaje de programación Python, haciendo uso de tecnologías relacionadas con el ámbito del procesado y tratamiento de señales digitales, así como de redes neuronales. El proyecto se aborda en una primera instancia, con el objetivo de encontrar unos parámetros que sean característicos a la hora de representar un fichero de audio, como una escena determinada y poder ser diferenciada con respecto a otras [27].

Cristian L. Vidal-Silva, Aurora Sánchez-Ortiz, Jorge Serrano y José M. Rubio en su proyecto “Experiencia académica en desarrollo rápido de sistemas de información web con

Python y Django”, presenta la experiencia académica, tanto de profesores como estudiantes, en un curso de último año para el desarrollo de sistemas de información web mediante el uso del marco de trabajo Django del lenguaje de programación Python. Django facilita el desarrollo de aplicaciones web con el uso del patrón de desarrollo Modelo-Plantilla-Vista (MTV, en inglés). Por la simplicidad sintáctica y práctica de Python y Django, los estudiantes de una carrera del área de administración y ciencias de la información fueron capaces de desarrollar prototipos de sistemas de información [28].

12.1.2 Búsqueda de Referencias

Python es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. La elegante sintaxis de Python y su tipado dinámico, la extensa librería estándar se encuentran disponibles libremente en código fuente y de forma binaria para la mayoría de las plataformas desde la Web de Python, y se pueden distribuir libremente [29].

Gissella Danelia Grillo Moreno, Luis Armando Lezama Barrientos y Ernesto de Jesús Ordóñez González, en su proyecto de grado titulado “Prácticas de laboratorio para el desarrollo de aplicaciones de escritorio implementando el lenguaje de programación Python 3 para el Departamento de Computación de la UNAN-León” consta de una serie de prácticas de laboratorio desarrolladas paso a paso para el mejor entendimiento de los estudiantes y docentes que quieran utilizarlo como material de apoyo al momento de impartir alguna sesión de clases en la que se vea involucrado el lenguaje de programación Python, orientado al desarrollo de aplicaciones de escritorio [30].

Roldán Blay, Carlos, en su proyecto titulado “Configuración de Spyder para programar aplicaciones de escritorio con Tkinter en Python” explica que para poder desarrollar aplicaciones de escritorio que se puedan exportar a directorios y archivos con extensión *.exe, se puede usar la librería Tkinter de Python. Además, en el desarrollo de aplicaciones se pueden necesitar librerías científicas o de otros tipos [31].

En el artículo “IRJET- Comparative Analysis of Python and Java for Beginners”, los autores incluyen una comparación de los dos lenguajes de programación más populares, mejor clasificados e indemnes: Python y Java. Una breve de Python y Java, incluyendo sus características, aplicaciones, ventajas y desventajas. La comparación de los dos lenguajes se basa en comparación de la sintaxis y las características [32].

12.1.3 Análisis Preliminar

Python como lenguaje de programación tienen diferentes enfoques y usos, dependiendo de los requerimientos y herramientas que se utilizan con el mismo, al ser un lenguaje de código abierto, enfocado a objetos, permite la creación de código legible y fácil de implementar.

Uno de los principales usos que tiene Python es el desarrollo web y la ciencia de datos, en el desarrollo web se usan Frameworks como Django y Flask, permitiendo integración de protocolos y minimizando tiempo en desarrollo.

Sin embargo, no se descarta el uso y la implementación del lenguaje en la creación de interfaces de escritorio, permitiendo el diseño de aplicaciones en el sector empresarial, una de las principales herramientas para la creación de interfaces con Python es:

Tkinter, esta es la interfaz por defecto de Python para implementar el kit de herramientas de GUI Tk, disponible en la mayoría de plataformas Unix y Windows, este paquete es mantenido por ActiveState [33].

En el procesamiento numérico se tiene la librería Numpy, especializada en la computación científica y el análisis de datos, especialmente cuando es un volumen elevado de información, esta librería usa el concepto de arrays, que agrupa datos de un mismo tipo en las dimensiones requeridas permitiendo su manipulación [34].

Para el procesamiento y presentación de gráficos, Python cuenta con una librería especializada que se denomina Matplotlib, esta permite la creación de gráficos en dos dimensiones, entre los más comunes están:

- Diagramas de barras
- Histogramas
- Mapas de color, entre otros
- El módulo que permite la manipulación y configuración es pyplot y permite la manipulación de subgráficos necesarios [35].

En el caso del proyecto se vio la necesidad de implementar el módulo random, que permite la creación de funciones y datos aleatorios, con ciertas condiciones para poder continuar con el procesamiento y codificación, se puede mencionar que los datos generados son pseudoaleatorios, debido a que son seleccionados al azar, pero con parámetros definidos que deben cumplir [36].

12.2 REALIZAR EL ALGORITMO

12.2.1 Crear un método de análisis

Para poder desarrollar una herramienta computacional que sea apoyo pedagógico en la enseñanza de los sistemas de comunicaciones digitales se definieron las temáticas que se incluyen en el proyecto, así como la manera en que se relaciona cada una con la herramienta computacional.

Entre los conocimientos que se deben adquirir e impartir en la asignatura Sistemas de Comunicaciones Digitales se pueden tomar dos bloques importantes: Modulación digital y modulación en banda base (conocida también como códigos de línea).

La modulación digital tiene como propósito la transmisión de información mediante la amplitud de una señal portadora sinusoidal, el símbolo debe ser transmitido en un tiempo establecido, este tiempo implementado es el inverso de la tasa de símbolo. En el caso de la modulación binaria, la información se transmite por símbolos equivalentes cada uno a un bit, mientras que en las modulaciones M-arias, cada símbolo especificado hace referencia a $\log_2 M$ bits.

Las técnicas de modulación digital que se implementan en este proyecto, corresponden a BFSK (Binary Frequency Shift Keying), BPSK (Binary Phase Shift Keying) y QPSK (Quadrature Phase Shift Keying), en adelante en el libro solo se hará referencia por sus siglas en inglés.

12.2.1.1 Modulación Digital

BFSK:

En la figura 5 se observa que en las modulaciones FSK existe un cambio en la frecuencia de salida cada que cambia la condición lógica de la señal binaria de entrada: En consecuencia, la rapidez de cambio de salida es igual a la rapidez de cambio de la entrada. A esta rapidez de cambio se le denomina frecuencia de bits o rapidez de transferencia de bits. Sus unidades son bits por segundo. A la rapidez de cambio de la salida del modular se le llama baudio [37].

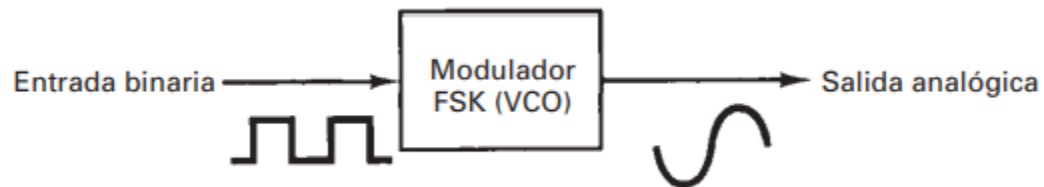


Figura 5 Entrada Binaria, Entrada Analógica

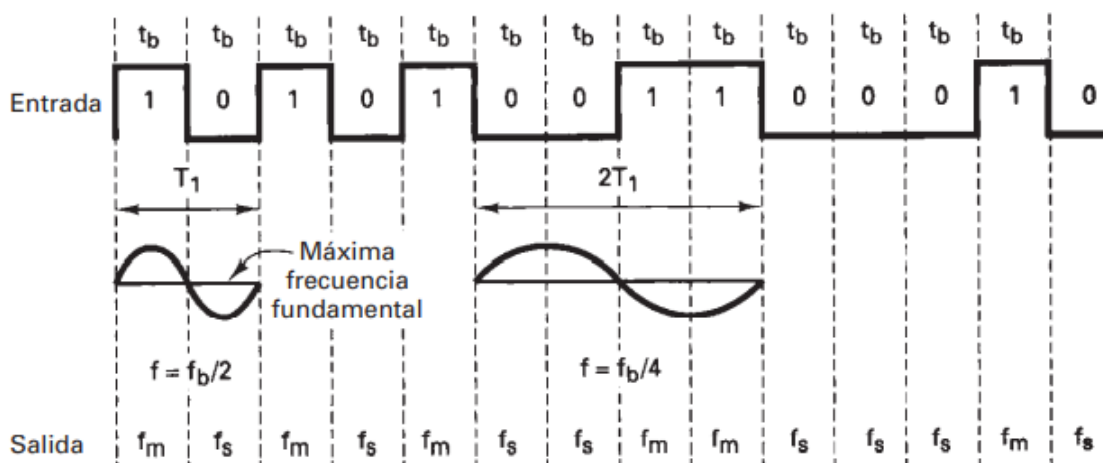


Figura 6 Imágenes tomadas del capítulo 12 del libro de Tomasi

En la BFSK, la amplitud de la señal de entrada solo puede contener dos valores: uno para una condición de 1 lógico y uno para una condición de cero lógico como se aprecia en la figura 6. Debido a esto, la desviación máxima en la frecuencia no varía, y obtiene su valor máximo.

BPSK:

Esta es una manera de modulación digital angular con amplitud constante. En este caso la señal de entrada es una señal digital binaria, y se puede tener una cantidad limitada de fases de salida como se demuestran en las figuras 7 y 8, al tratarse de modulación binaria, solo son posibles dos fases de salida para una única frecuencia portadora. Una fase de salida representa un 1 lógico, y la otra un cero lógico. La fase de la portadora varía entre dos ángulos desfasados 180° .

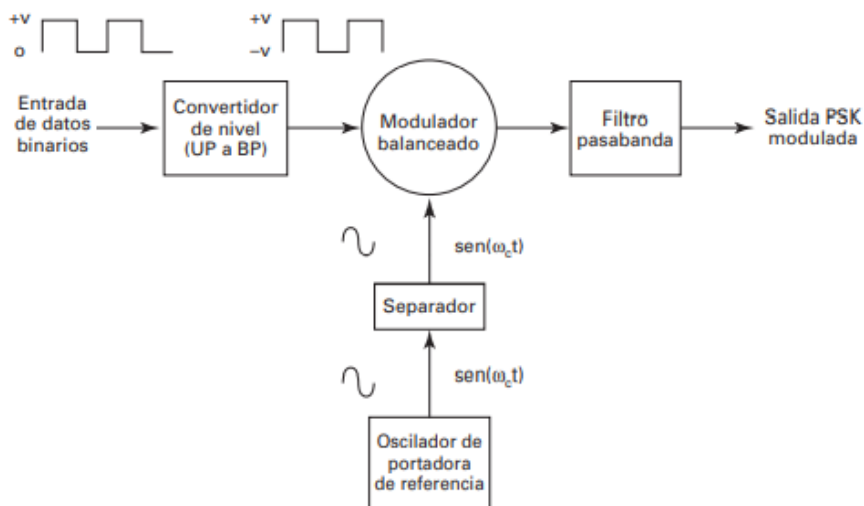


Figura 7 Representación de un transmisor BPSK – imagen de Tomasi

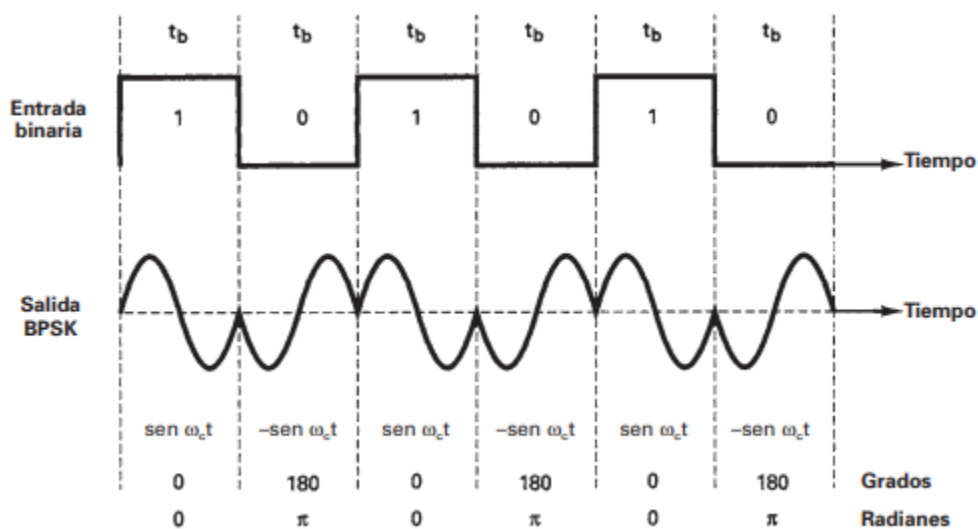


Figura 8 Relación en función del tiempo para una modulador BPSK (Tomasi)

QPSK:

Esta es otra forma de modulación digital angular y con amplitud constante. Así es, entonces una técnica M-aria de codificación en la que $M = 4$. Con este tipo de codificación, se permiten cuatro fases de salida para una frecuencia de portadora. Al existir cuatro posibles salidas hay cuatro condiciones de entrada como se observa en la tabla 7 siendo esto demostrado en la figura 9.

Tabla 7 Datos Fase de Salida

Entrada binaria		Fase de salida QPSK
Q	I	
0	0	-135°
0	1	-45°
1	0	+135°
1	1	+45°

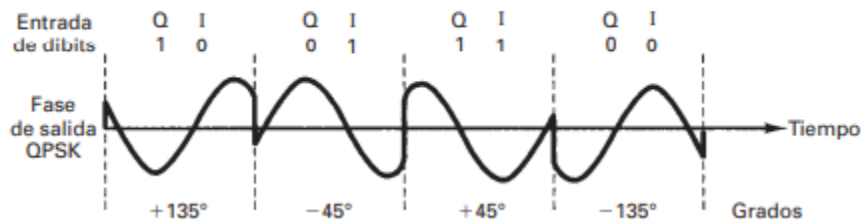


Figura 9 Fase de salida en función del tiempo para un modulador QPSK

12.2.1.2 Modulación Banda Base

La figura 10 explica que un modulador digital se identifica como un bloque que realiza una transformación de señales y que puede representarse de forma simplificada, su señal de entrada es discreta, con valores binarios. Su velocidad se mide en bits por segundos, tal como se definió anteriormente. La señal de salida es una onda continua en tiempo y transporta esta información binaria de la secuencia.

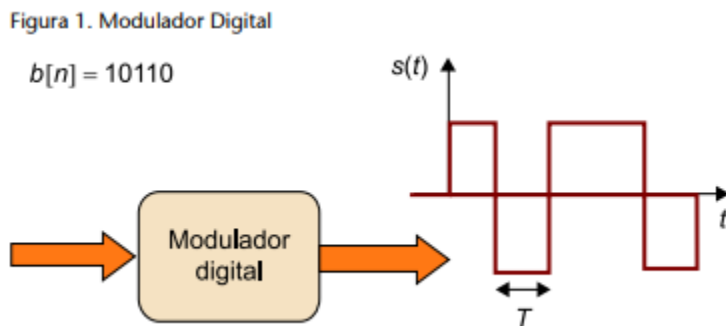


Figura 10 Figura tomada del libro de la universidad cataluya

A los sistemas que implementan estas modulaciones en banda base se les denomina también códigos de línea, estos transmiten la señal demodulada digitalmente a una frecuencia baja, sin trasladar la frecuencia en una banda específica.

Por tanto, son utilizados para conexiones directas entre equipos de comunicaciones para tratamientos digital de señal y la representación de la señal modulada, representa la tensión presente en los cables físicos de conexión de estos equipos.

MODULACIONES NRZ Y RZ

Por sus siglas Non Return Zero y Return Zero, son fundamentales en la transmisión de información en banda base.

La figura 11 explica que para el caso de la modulación NRZ se asigna un nivel de tensión positivo cuando el bit a transmitir es 1 y un nivel de tensión negativo cuando el bit a transmitir es un 0. Para la modulación RZ se asigna de la misma manera que NRZ, pero, una vez transmitido el bit, debe retornar a la tensión de 0 antes de producir el bit siguiente.

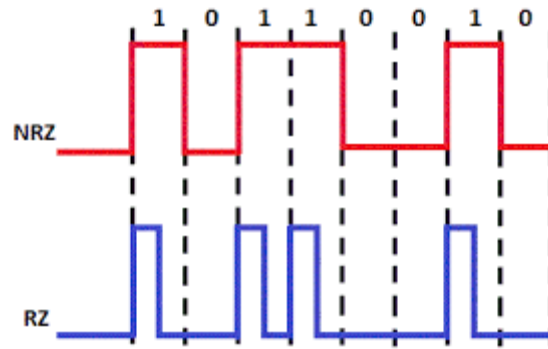


Figura 11 Modulaciones NRZ y RZ

MODULACIONES AMI

En la figura 12 explica una modulación bipolar también denominada marca alternada AMI (Aternate Mark Inversión), se asigna una tensión cero a un bit cero y tensión positiva y negativa, cuando se transmite un bit 1.

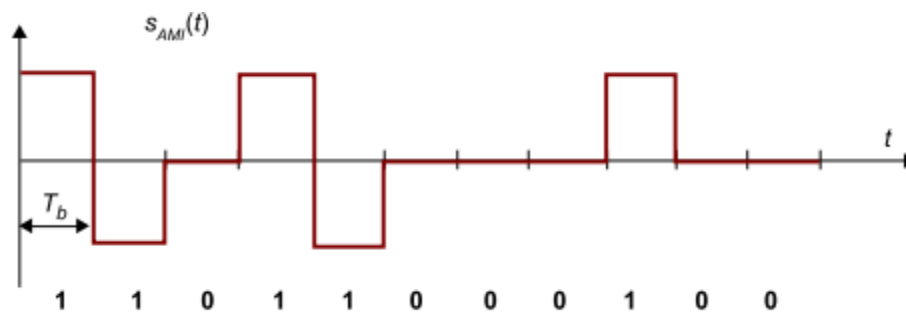


Figura 12 Modulación AMI

Para la codificación de línea se recomiendan seis factores a considerar para seleccionar un formato para la modulación:

- Voltajes de transmisión y componente CD
- Ciclo de trabajo
- Ancho de banda
- Recuperación de reloj

- Detección de errores
- Facilidad de detección y decodificación

MODULACIÓN BIFÁSICA DIGITAL:

En este caso hace referencia al código Manchester demostrado en la figura 13, en donde se produce un componente sólido en la sincronización, sin variación en CD y facilitando la recuperación de reloj.

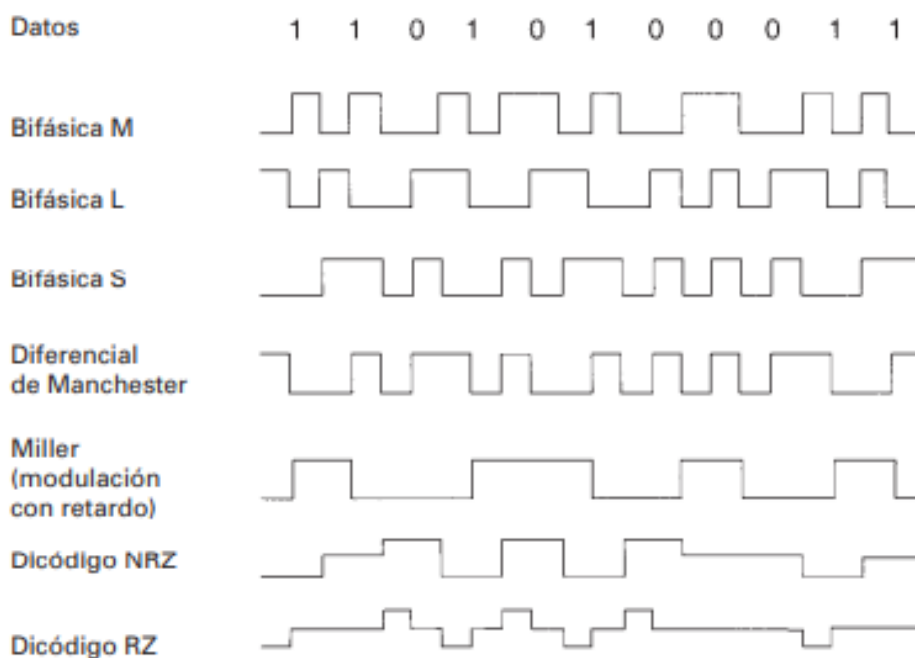


Figura 13 Ejemplos de códigos de línea tomasi

12.2.2 Representar el método mediante un diagrama de flujo

El algoritmo se compone de dos módulos principales uno para modulación digital y otro para la modulación en banda base; a nivel general ambos se componen de ingreso de data de valores binarios para su procesamiento y posteriormente observar en la interfaz gráfica las

señales que se espera obtener en cada tipo de tratamiento de datos. El algoritmo final que describe el comportamiento del sistema es representado por el diagrama de flujo que se observa en la siguiente figura 14.

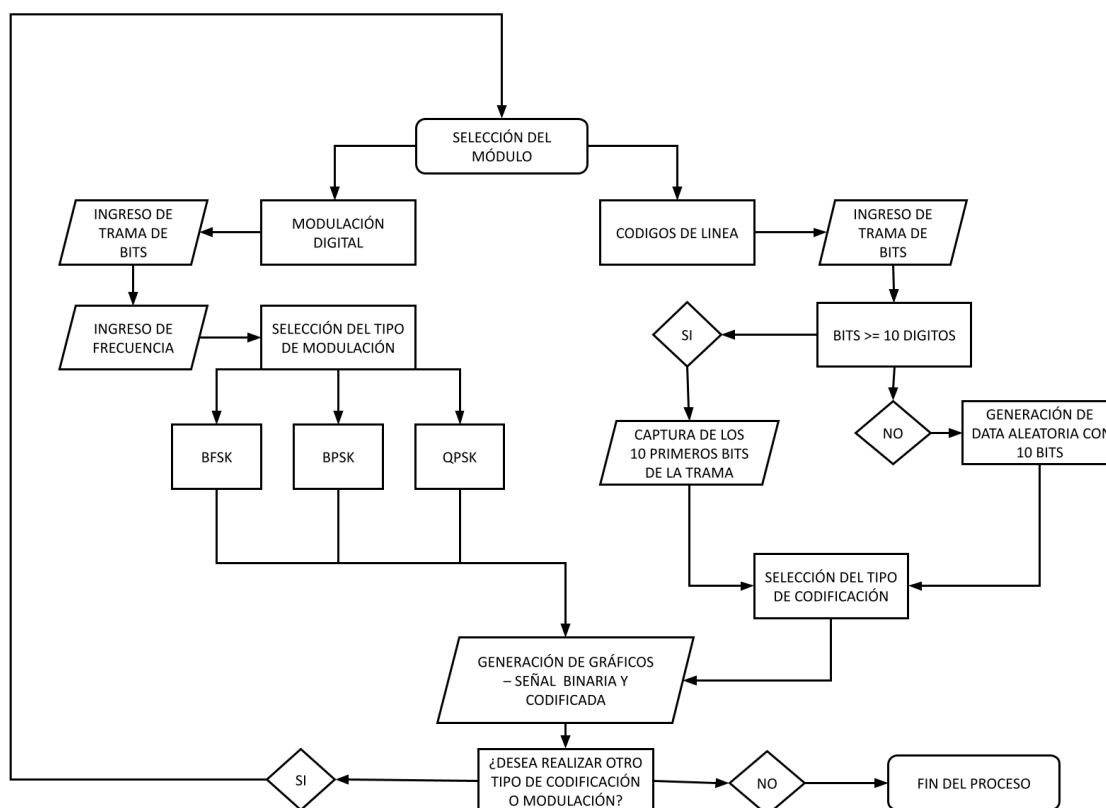


Figura 14 Diagrama de flujo

12.3 CODIFICAR EL ALGORITMO

12.3.1 Elección del entorno de desarrollo

Para seleccionar la interfaz de desarrollo adecuada para el proyecto se realizó una explicación en la tabla 8. En esta se plantearon los Entornos de desarrollo integrados que pudieran ser útiles, según documentación de personas con experiencia en el área y estado. Los parámetros evaluados en la selección del entorno integrado de desarrollo fueron los siguiente, se

les dio una ponderación de 1 a 5, desde un valor mínimo a máximo respectivamente, dependiendo del nivel de desarrollo tenga con el ítem:

- Especialidad del lenguaje de programación, se le dio una ponderación de un 30% de peso a este ítem, puesto que se procuró usar un entorno enfocado al lenguaje seleccionado que es Python.
- Procesamiento de señales, tramas de bits, puesto que se trabaja procesamiento de datos ingresados por usuario o generados aleatoriamente, se asignó un peso del 35% a este parámetro.
- Visualización de gráficos, al ser una interfaz para el aprendizaje y avance en la asignatura de comunicaciones digitales, se procuró el IDE permita de manera clara y práctica el avance en el cada código implementado sin necesidad de ventanas emergentes, 15% de peso al ítem.
- Librerías, contar con librerías especializadas ayudan en el procesamiento y tratamiento de los datos, puesto que agilizan procesos y tiempos de respuesta, ponderación del 10%
- Licencia, se toma por último este ítem, permitiendo la modificación de código sin problema alguno.

Tabla 8 Elaboración propia

IDE Parámetro	Desarrollado para python	Peso	Procesamiento de trama de bits - Señales binarias	Peso	Visualización de gráficos	Peso	Librerías	Peso	Licencia	Peso	Total
Spyder	4	30,00%	4	35,00%	4,5	15,00%	4	10,00%	5	10%	4,175
Pycharm	4	30,00%	4	35,00%	4	15,00%	4	10,00%	4	10%	4
Sublime Text	4	30,00%	3	35,00%	4	15,00%	4,5	10,00%	5	10%	3,8
Atom	4	30,00%	3	35,00%	4	15,00%	3,5	10,00%	4	10%	3,6
Visual Studio Code	3	30,00%	3	35,00%	4,5	15,00%	4	10,00%	5	10%	3,525

12.3.2 Escritura del Código de programación de Python

Se explican los conceptos matemáticos y el área de las Comunicaciones digitales plasmadas en el algoritmo de Python desarrollado con el IDE Spyder y la implementación de los modulos Tkinter, NumPy, Matplotlib y Random entre otras. Se desarrolla el código para Modulación Banda Base (Anexo 1) y Modulación Digital (Anexo 2).

En la modulación digital se requiere una entrada digital de cierta trama de datos para poder aplicar el tipo de modulación requerido, para el proyecto desarrollado se permitió la generación de bits aleatoriamente o que esta sea ingresada por parte del usuario de la interfaz.

En la ecuación 1 se muestra una de las consideraciones importantes que es el ancho de banda:

$$\eta = \frac{\textit{bit}}{\frac{\textit{segundo}}{\textit{Hz}}} \quad (1)$$

$$\eta \rightarrow \textit{entre mayor el valor mejor en la modulación} \quad (2)$$

LIMITE DE SHANNON DE CAPACIDAD DE LA INFORMACIÓN:

La capacidad de información de un sistema de comunicación representa la cantidad de símbolos independientes que pueden transportarse en un sistema por unidad de tiempo. Esto corresponde a bit, el cual es el más básico digito binario, por tanto, la frecuencia expresa la capacidad de la información en bits por segundo.

LEY DE HARTLEY:

$$I \propto B \times T \quad (3)$$

en donde I=Capacidad de información (Bits por segundo)

B=Amplitud de banda (hertz)

T=Tiempo de transmisión (segundo)

Por tanto, la capacidad de la información es una función lineal de la amplitud de banda y del tiempo de transmisión, siendo directamente proporcional a ambos.

Probabilidad de error: Esto es una expectativa que tiene de cierto resultado, entonces, al indicar una probabilidad de error en el orden de los: 10^{-7} → indica que se puede tener un error de un bit cada 10.000.000 bits transmitidos

Por otra parte, cuando se refiere a Bit Error Rate, cuando indica que es de 10^{-7} , hace referencias a que se ha cometido un bit de error por cada 10.000.000 transmitidos.

MANIPULACIÓN POR DESPLAZAMIENTO DE LA FRECUENCIA - FSK

La ecuación 4 define la modulación FSK binaria y su definición de observa en la ecuación 5, 6, 7, 8 y 9:

$$v_{fsk}(t) = v_c \cos\{2\pi[f_c + v_m(t)\Delta f]t\} \quad (4)$$

$$v_{fsk}(t) = \text{forma e onda binaria fsk} \quad (5)$$

$$v_c = \text{amplitudde la portadora} \quad (6)$$

$$f_c = \text{frecuencia central de la portadora} \quad (7)$$

$$\Delta f = \text{desviación máxima de frecuencia} \quad (8)$$

$$v_m(t) = \text{señal moduladorade entrada binaria } (\pm 1) \quad (9)$$

La ecuación 10 muestra que la señal moduladora puede tomar valor diferente si es un 1 al ingreso o un cero lógico (-1) como se define en la ecuación 11.

$$v_{fsk} = v_c \cos[2\pi(f_c + \Delta f)t] \rightarrow \text{cuando el valor } + 1 \quad (10)$$

$$v_{fsk} = v_c \cos[2\pi(f_c - \Delta f)t] \rightarrow \text{cuando el valores } - 1 \quad (11)$$

Estos valores binarios ingresados por usuario requieren de las siguientes librerías para su manipulación:

La captura de esta información inicial de datos binarios se realiza con el método `get` de Python implementado en Tkinter [38].

Se definen dos tipos de Arrays mediante el uso de Numpy, los cuales contienen ceros o unos según requiera en la codificación, cada uno de estos tendrá un tamaño que es igual al inverso de la velocidad de bits que se ingresa por parte del usuario. Estas secuencias por facilidad en el procesamiento de los datos se les denomina `s1`, `s0`

Cada iteración almacenará en una lista las secuencias obtenidas, así se diferenciaría la modulación en frecuencia de la señal.

Se realiza la implementación de la librería `matplotlib.pyplot` [39] para la creación de gráficos como se aprecia en la figura 15. Se debe crear un array con la librería `numpy` [40] del tamaño de la data de secuencias indexadas para poder representar visualmente el procesamiento obtenido, indicando los límites en el eje de las abscisas y el de las coordenadas, observando estos resultados en las figuras 16 y 17.

VIRTUAL LABORATORY OF DIGITAL COMMUNICATIONS SYSTEMS

DIGITAL MODULATION
INPUT IN BINARY SYSTEM THE NUMBER TO CONVERT

INPUT MODULATION FREQUENCY

INPUT MODULATE AMPLITUDE

Figura 15 Ingreso de datos para modulación digital

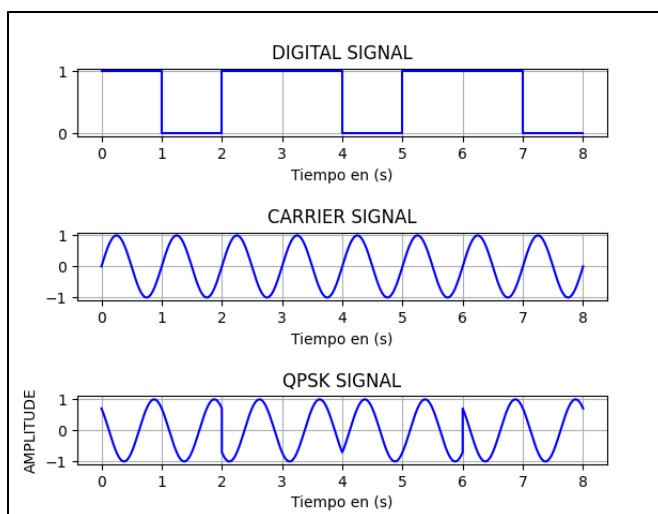


Figura 16 Modulación por desplazamiento cuaternario de fase

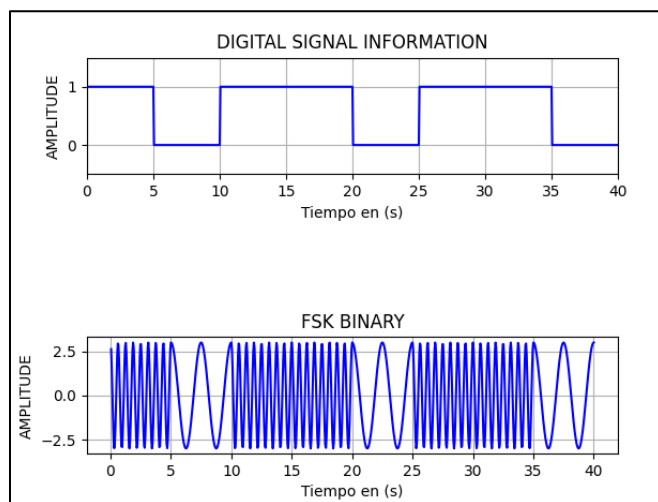


Figura 17 Modulación por desplazamiento de frecuencia binario

Para el caso de la Modulación en banda base, se importan las librerías, Numpy, Pyplot y se hizo sobre la misma interfaz gráfica basada en Tkinter, a diferencia de la modulación se pide al usuario el ingreso de una secuencia binario de 10 bits, en caso de ser contrario se genera aleatoriamente esa trama de bits por parte de la librería Random en Python y se solicita el bit rate o bits por segundo.

La selección de los tipos de códigos de línea corresponde a los temas que se observan en las figuras 18, 19, 20, 21 y la figura 22 es la representación del ingreso de datos incompletos:

AMI RZ – AMI NRZ

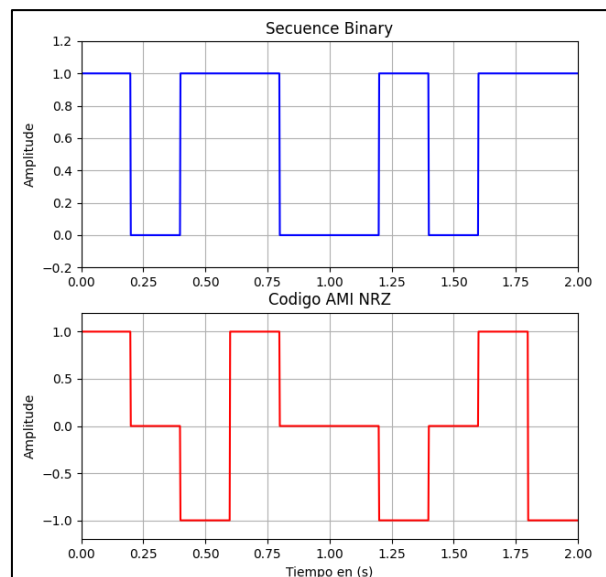


Figura 18 AMI RZ – AMI NRZ

Bipolar RZ – Bipolar NRZ

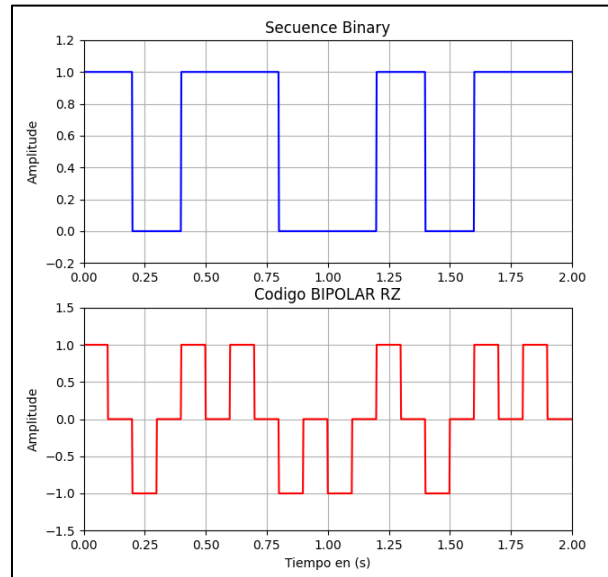


Figura 19 Bipolar RZ – Bipolar NRZ

Unipolar RZ – Unipolar NRZ

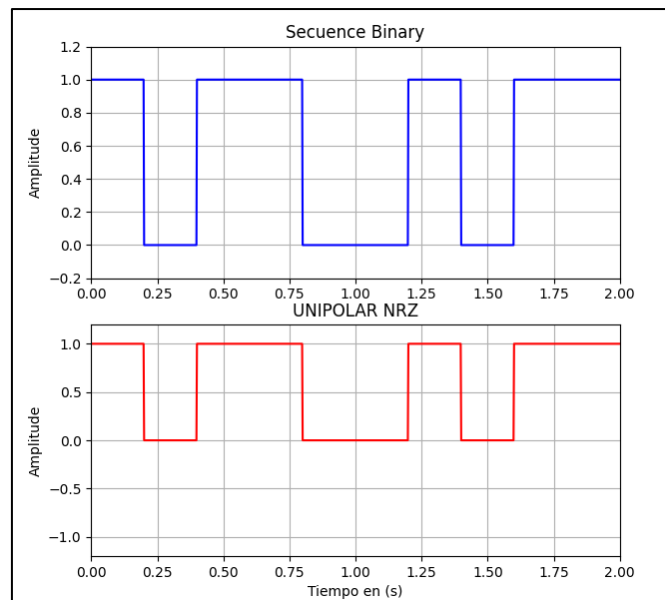


Figura 20 Unipolar RZ – Unipolar NRZ

Manchester

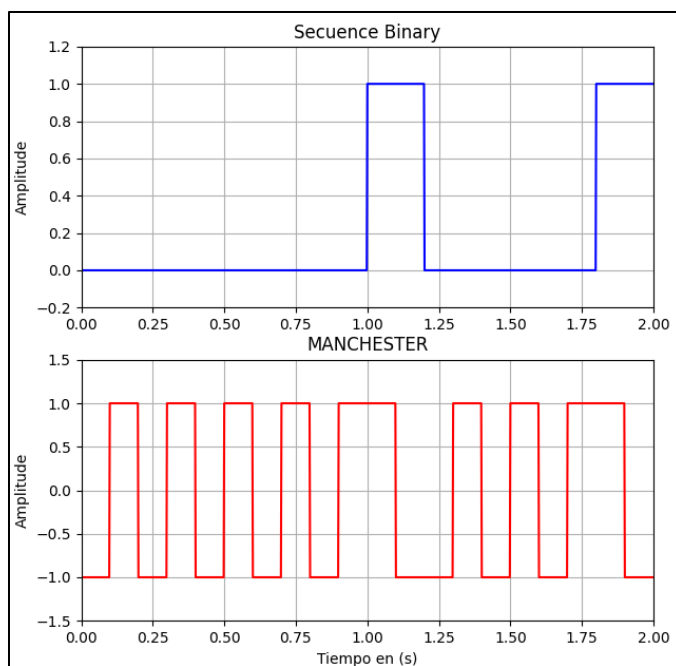


Figura 21 Unipolar RZ – Unipolar NRZ

Mensaje generado al no ingresar la trama de datos completa.

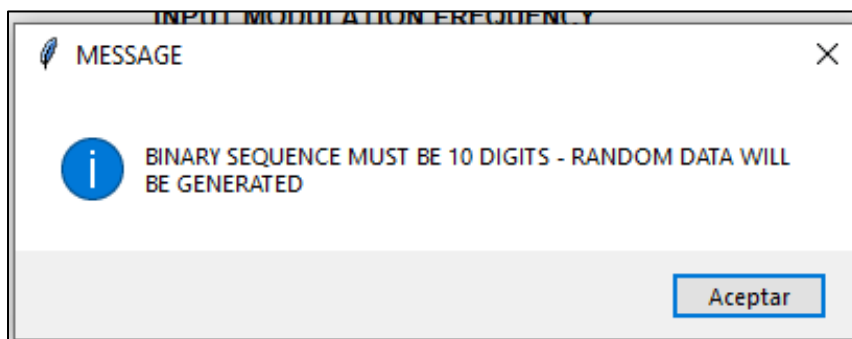


Figura 22 Mensaje generado por la interfaz

12.4 EVALUAR EL FUNCIONAMIENTO DEL ALGORITMO

12.4.1 Funcionamiento del Algoritmo. Para la demostración de su funcionamiento se realiza una comparación con el programa que se encuentra en funcionamiento en la asignatura.

12.4.1.1 AMI RZ. Su funcionamiento se observa en la figura 23, siendo la figura 24 su comparación

- 1100101000

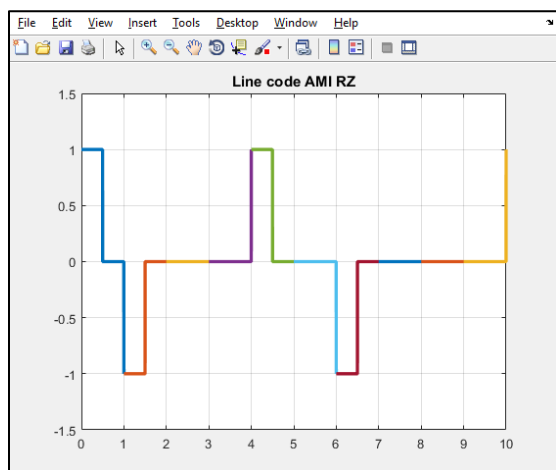


Figura 24 AMI RZ Matlab

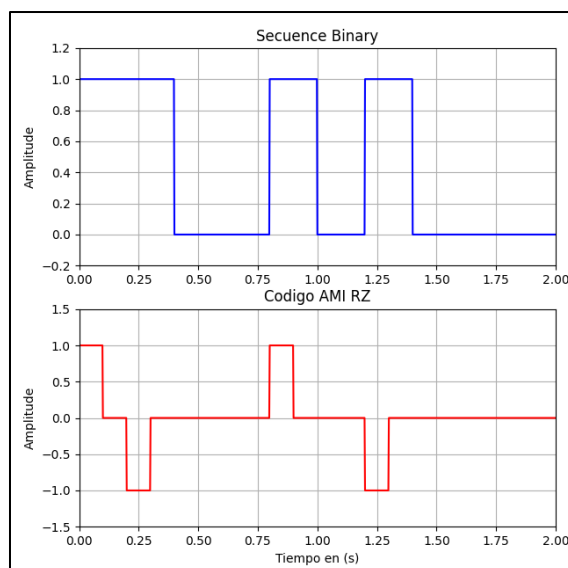


Figura 23 AMI RZ Python

12.4.1.2 AMI NRZ. Su funcionamiento se observa en la figura 26, siendo la figura 25 su comparación.

- 1100101000

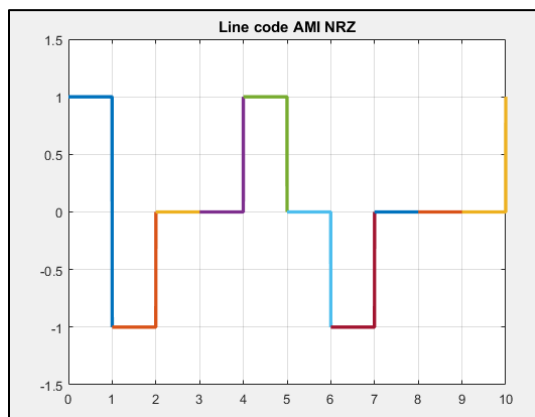


Figura 25 AMI NRZ Matlab

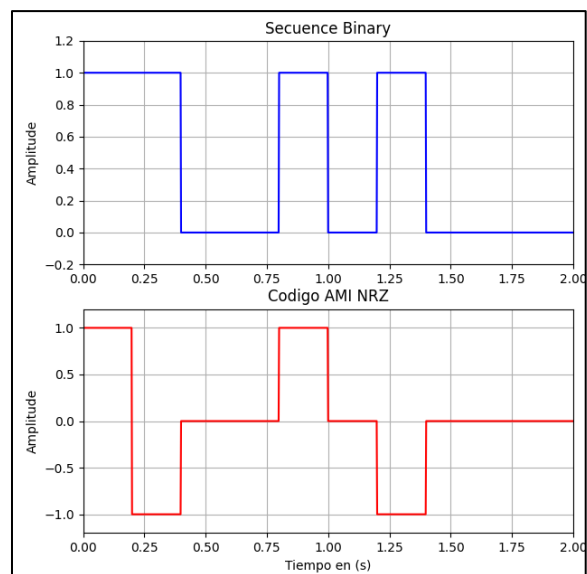


Figura 26 Python

12.4.1.3 BRZ. Su funcionamiento se observa en la figura 28, siendo la figura 27 su comparación.

- 1100101000

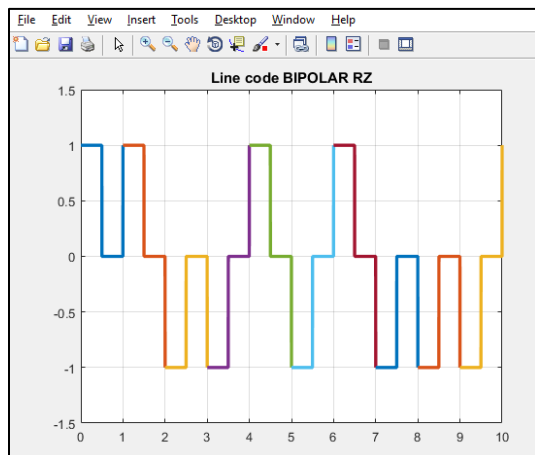


Figura 27 BRZ Matlab

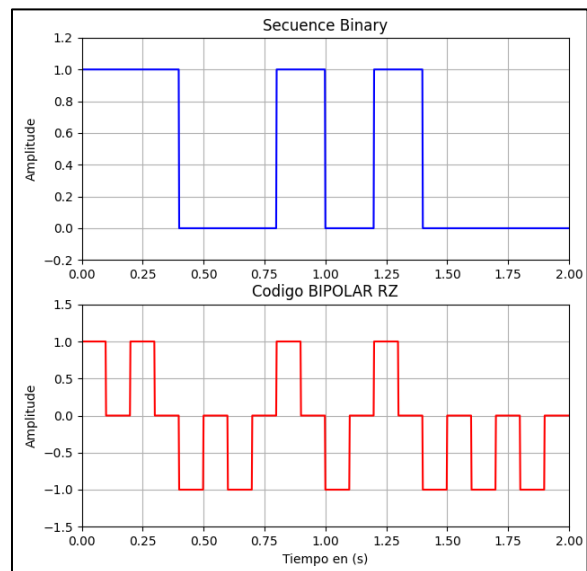
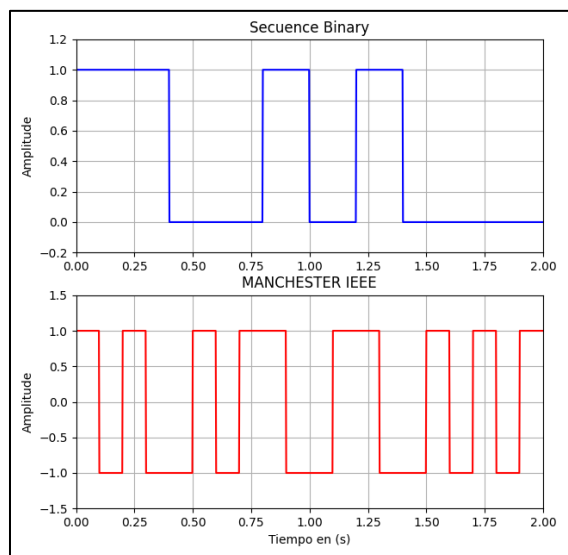
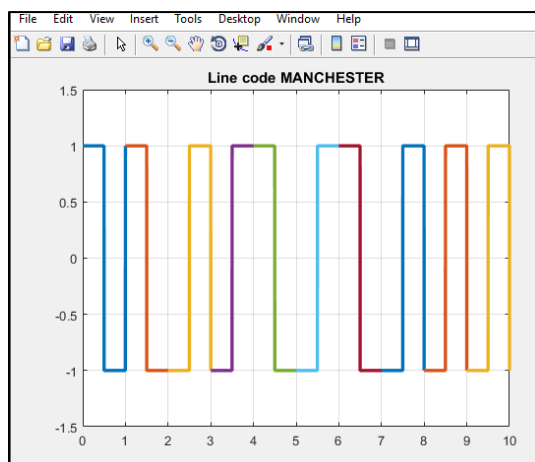


Figura 28 BRZ Python

12.4.1.4 MANCHESTER. Su funcionamiento se observa en la figura 29, siendo la figura 30 su comparación

- 1100101000



[Figura 29 Manchester Python](#)

12.4.1.5 URZ. Su funcionamiento se observa en la figura 32, siendo la figura 31 su

[Figura 30 Manchester Matlab](#)

comparación

- 1100101000

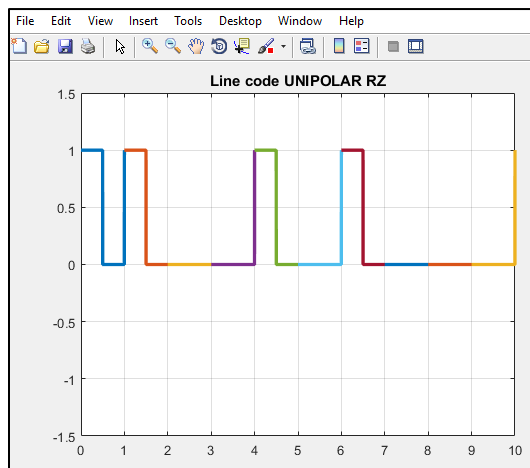


Figura 31 URZ Matlab

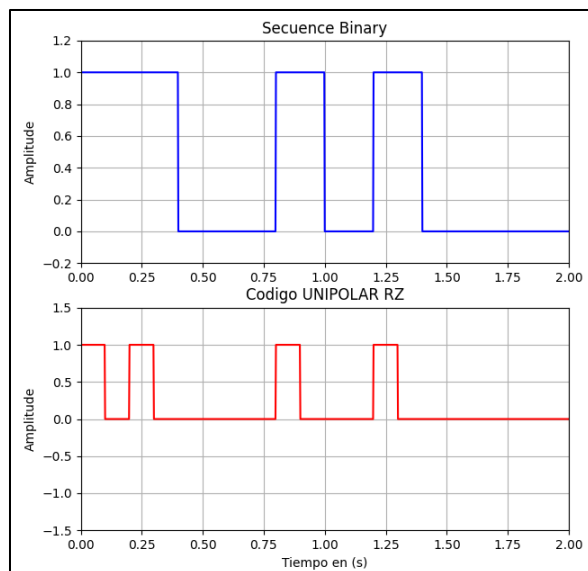


Figura 32 URZ Python

12.4.1.6 UNRZ. Su funcionamiento se observa en la figura 34, siendo la figura 33 su comparación

- 1100101000

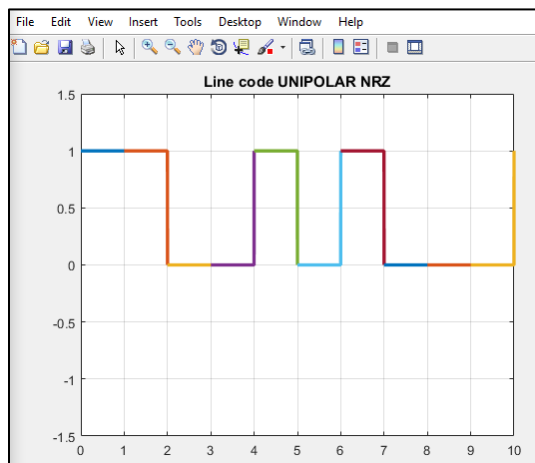


Figura 33UNRZ Matlab

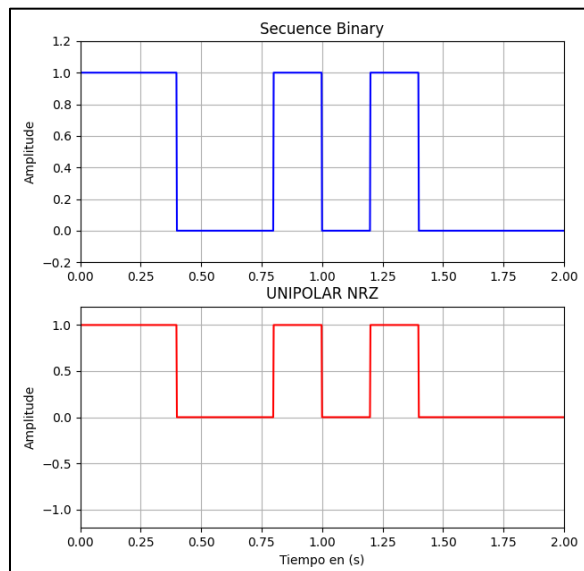


Figura 34 UNRZ Python

12.4.1.7 BFSK. Su funcionamiento se observa en la figura 36, siendo la figura 35 su comparación

- **1 0 1 0 1 0 0 1 1 0**
- **Frecuencia (Hz): 5**
- **Amplitud(V): 10**

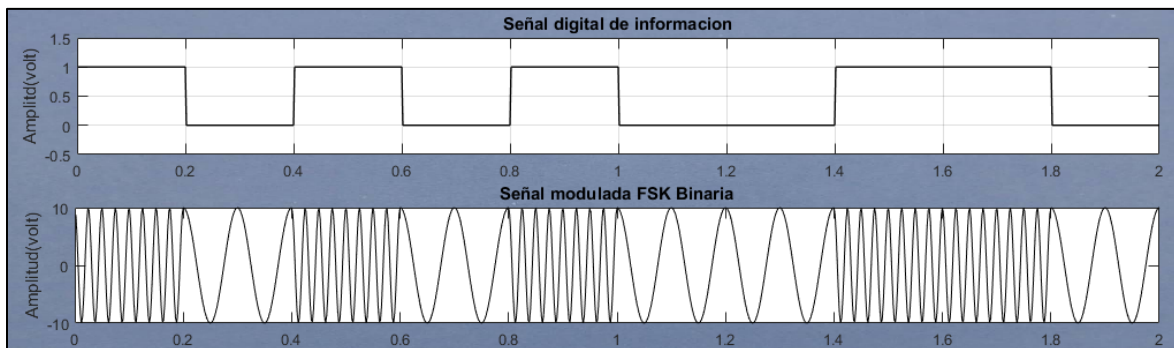


Figura 35 BFSK Matlab

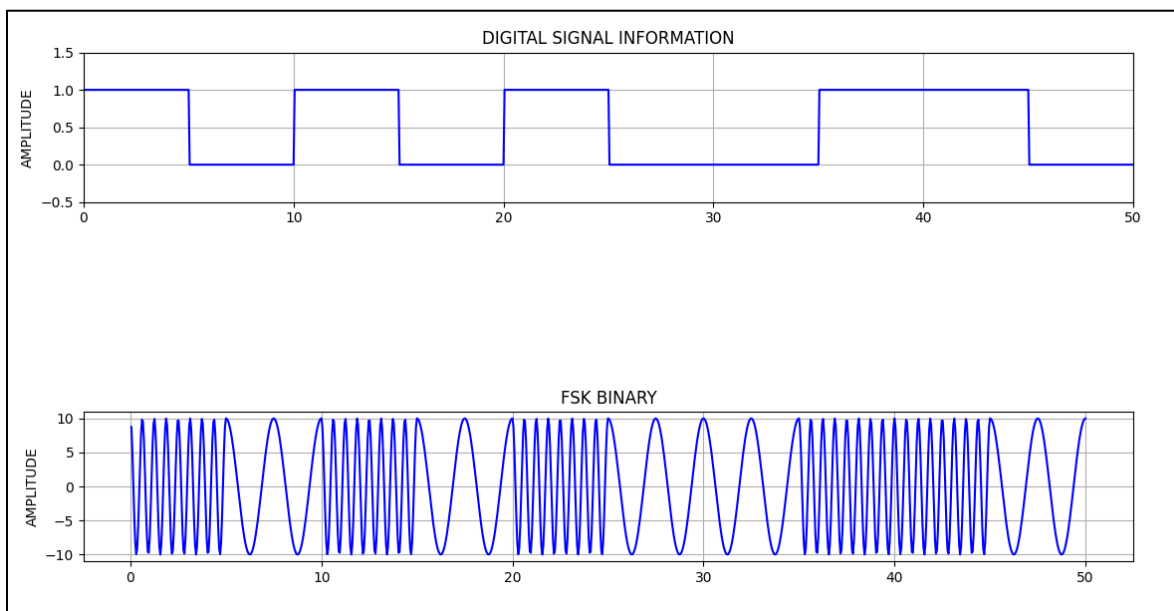


Figura 36 BFSK Python

12.4.1.8 BPSK. Su funcionamiento se observa en la figura 38, siendo la figura 37 su comparación

- **1 0 1 0 1 0 0 1 1 0**
- **Frecuencia (Hz): 5**
- **Amplitud(V): 10**

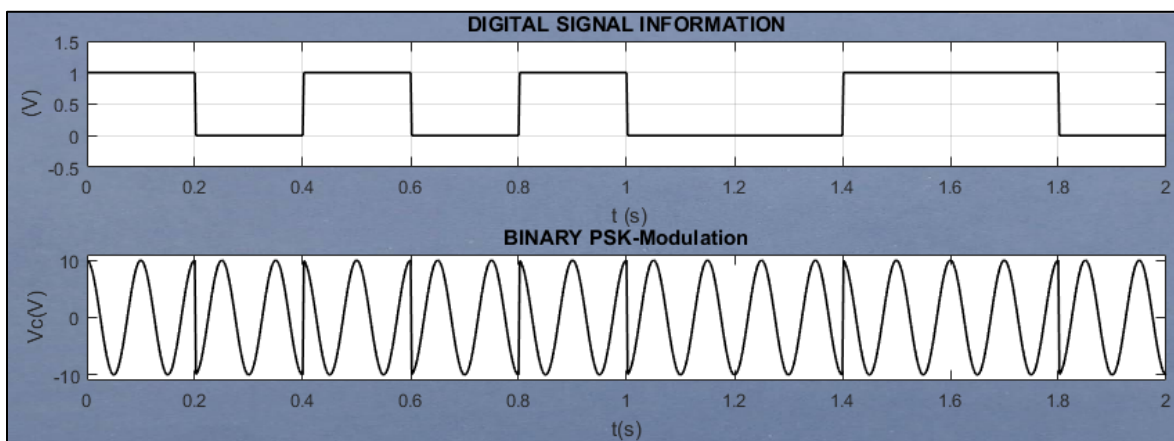


Figura 37 BPSK Matlab

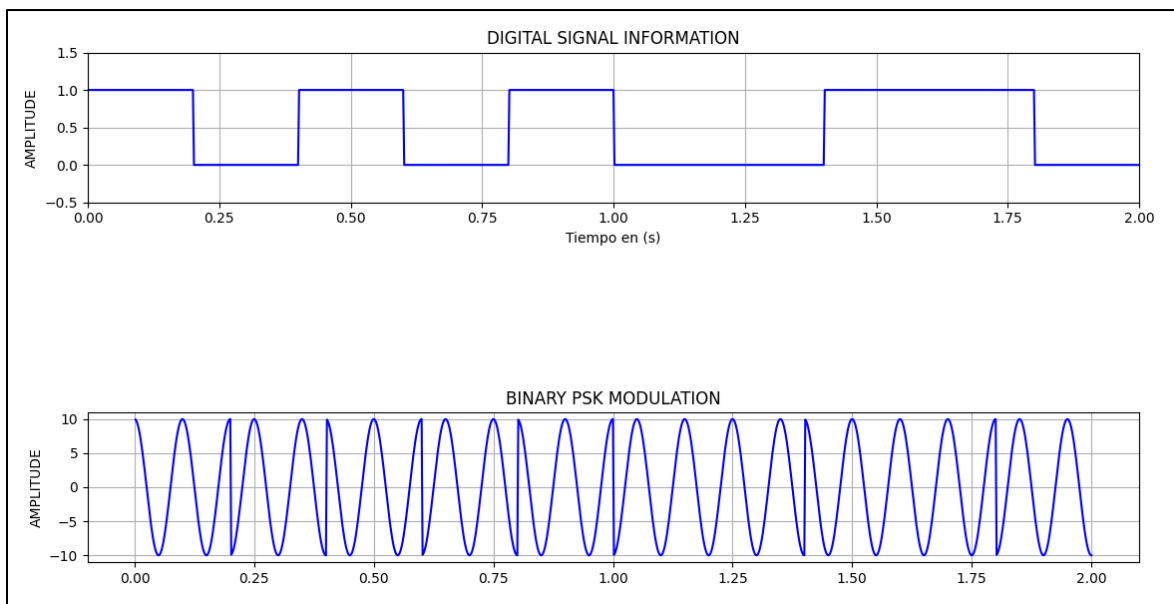


Figura 38 BPSK Matlab

12.4.1.9 QPSK. Su funcionamiento se observa en la figura 40, siendo la figura 39 su comparación

- **1010100110**

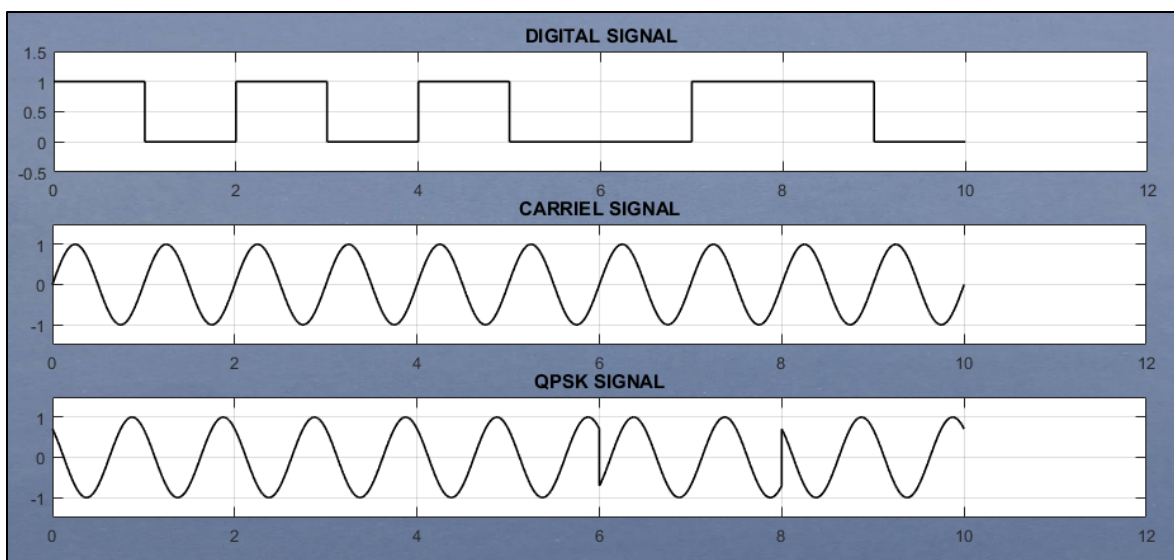


Figura 39 QPSK Matlab

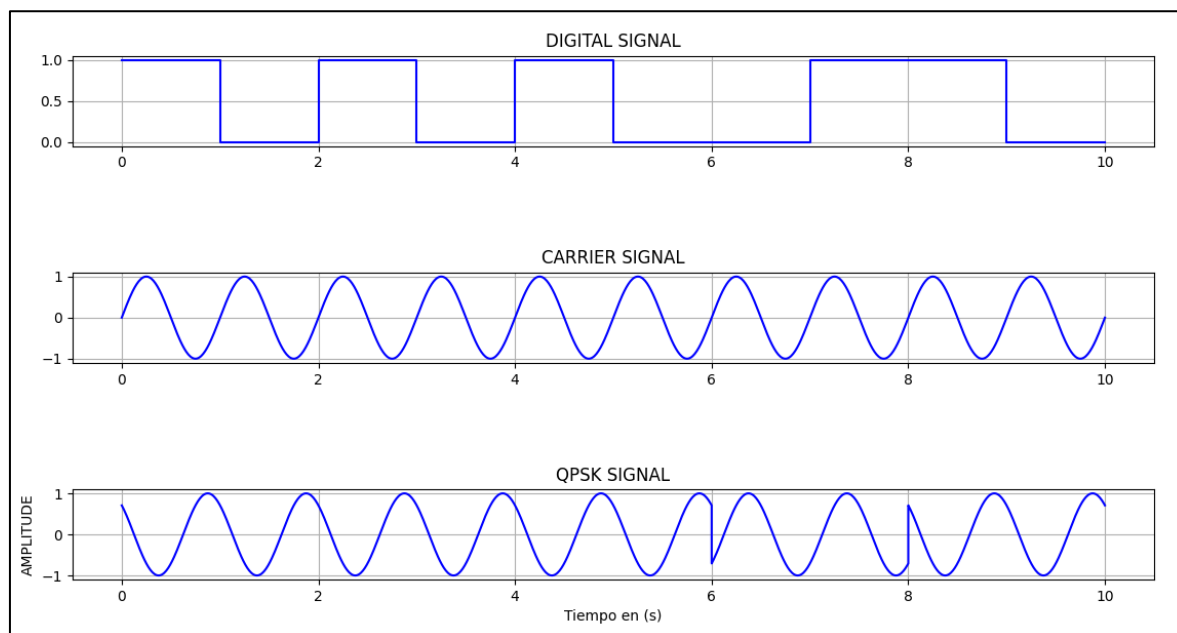


Ilustración 40 QPSK Python

12.4.2 Implementar algunas pruebas

Se toman ejercicios planteados, algunos desarrollados y otros como retos de trabajo en el libro de Tomasi, se abordan los temas de Modulación en banda base (Códigos de línea) y Modulación Digital (BFSK, BPSK, QPSK).

12.4.2.1 AMI RZ. Los resultados de este código de línea se aprecian en las figuras 41, 42, 43 y 44.

- 1010100110

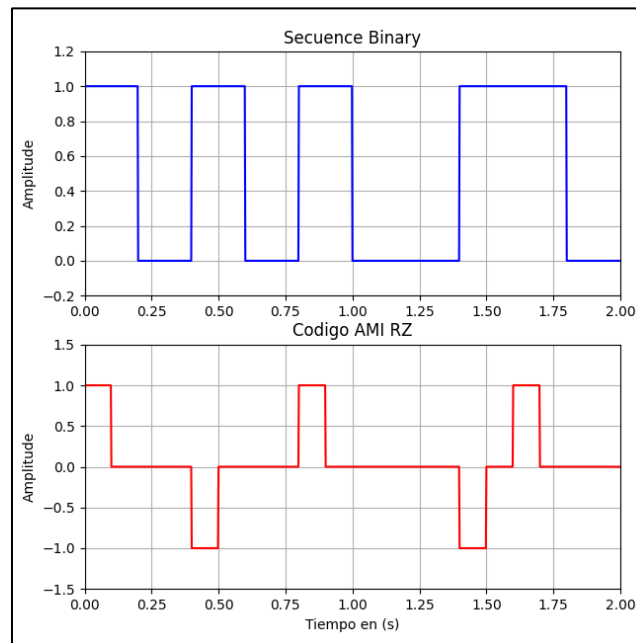


Figura 41 Prueba AMI RZ 1010100110

- 1011001010

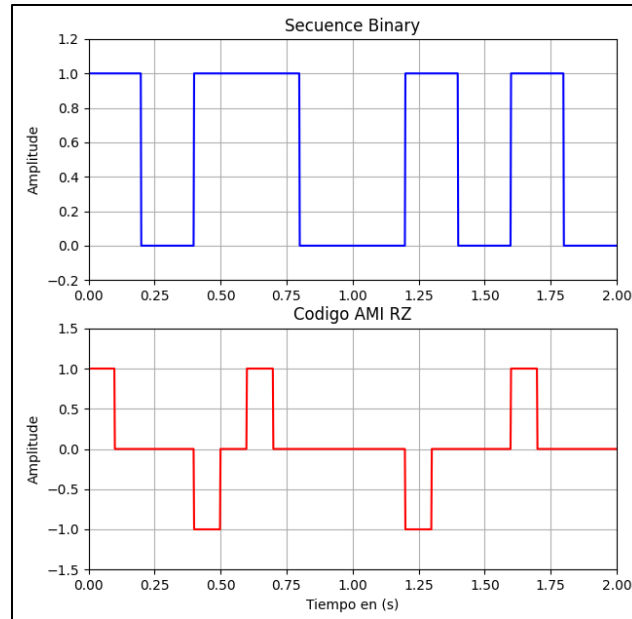


Figura 42 Prueba AMI RZ 1011001010

- 1101100010

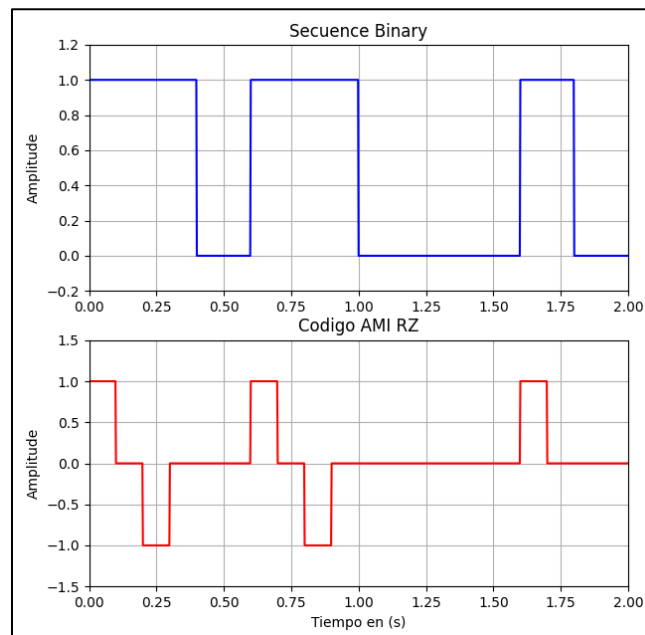


Figura 43 Prueba AMI RZ 1101100010

- 1101010001

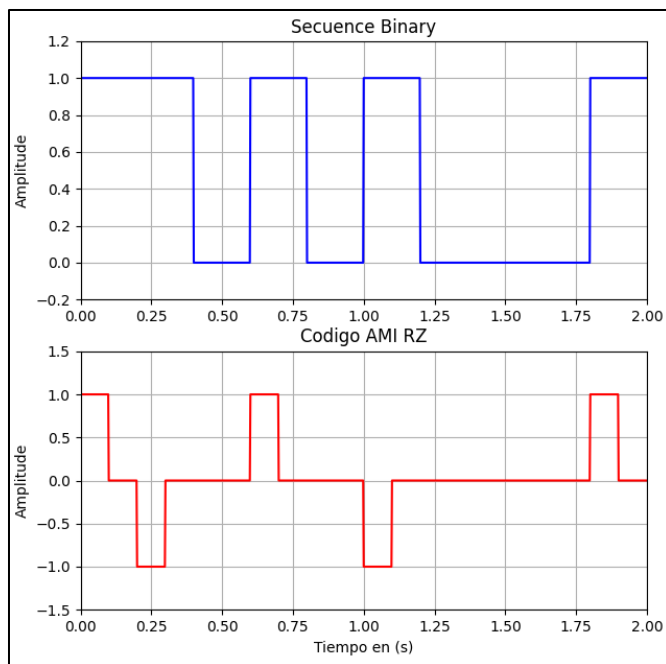


Figura 44 Prueba AMI RZ 1101010001

12.4.2.2 AMI NRZ. Los resultados de este código de línea se aprecian en las figuras 45, 46, 47 y 48.

- 1010100110

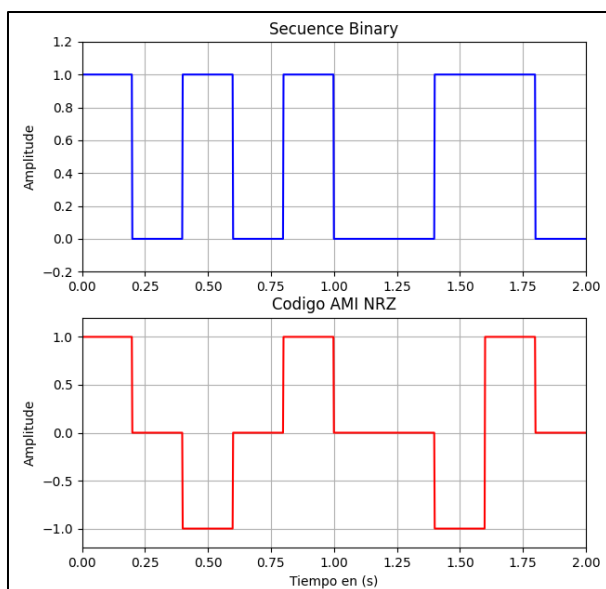


Figura 45 Prueba AMI NRZ 1010100110

- 1011001010

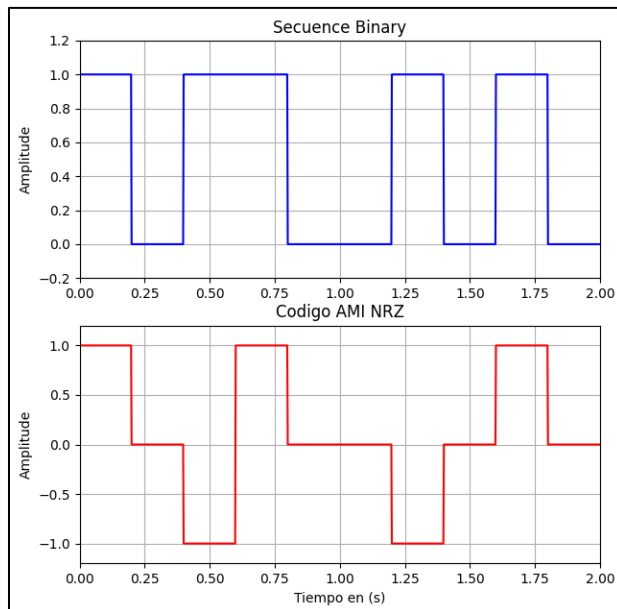


Figura 46 Prueba AMI NRZ 1011001010

- 1101100010

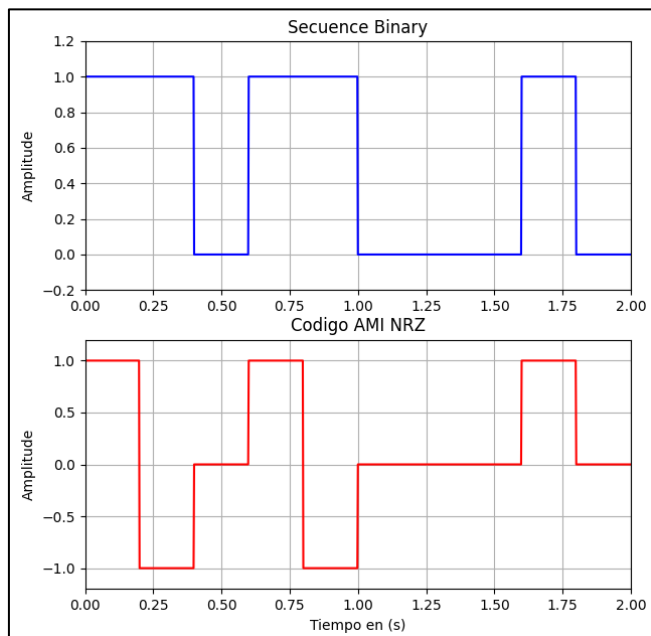


Figura 47 Prueba AMI NRZ 1101100010

- 1101010001

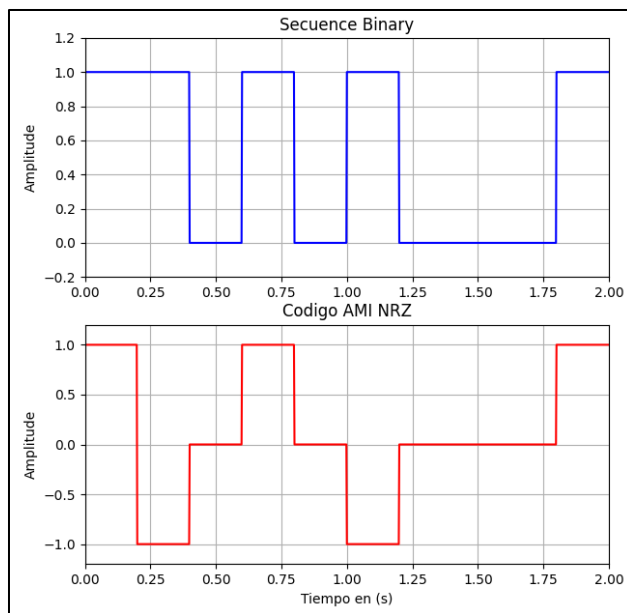


Figura 48 Prueba AMI NRZ 1101010001

12.4.2.3 BRZ. Los resultados de este código de línea se aprecian en las figuras 49, 50, 51 y 52.

- 1010100110

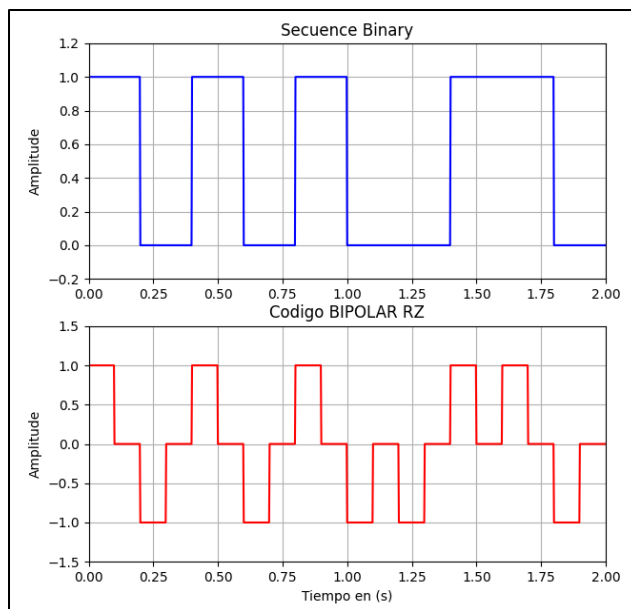


Figura 49 BRZ Prueba BRZ 1010100110

- 1011001010

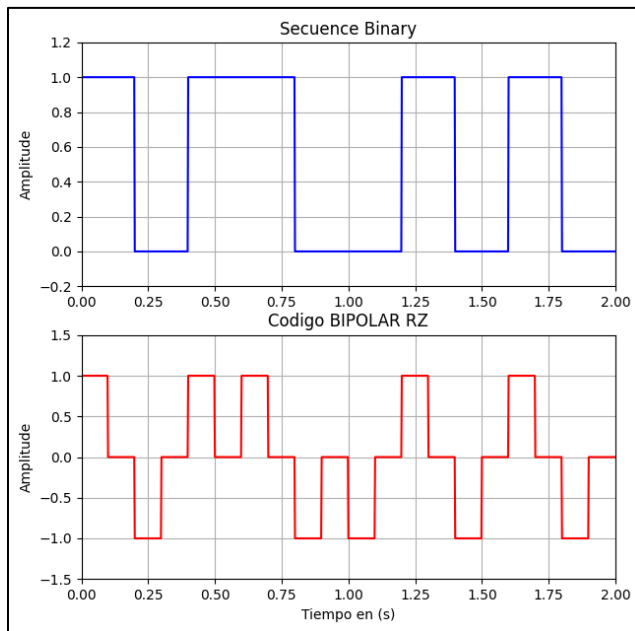


Figura 50 Prueba BRZ 1011001010

- 1101100010

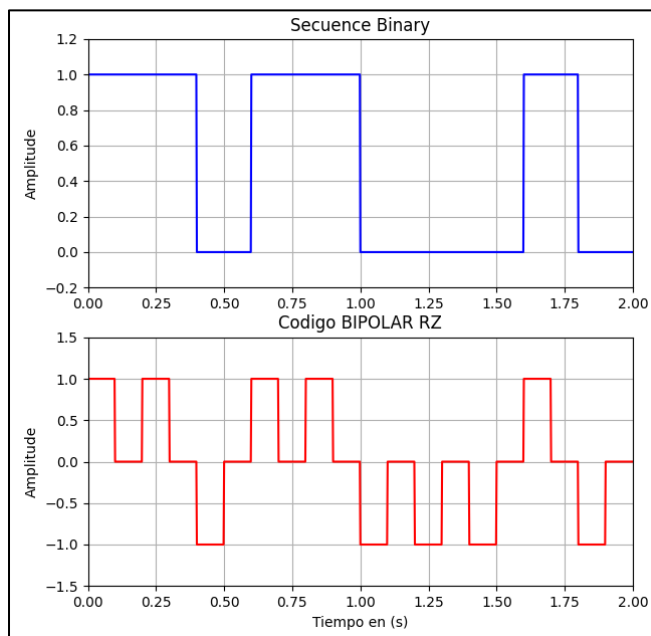


Figura 51 Prueba BRZ 1101100010

- 1101010001

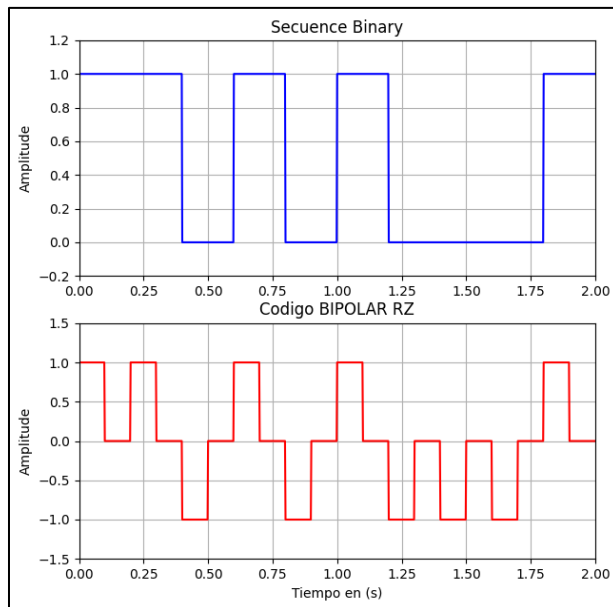


Figura 52 Prueba BRZ 1101010001

12.4.2.4 MANCHESTER. Los resultados de este código de línea se aprecian en las figuras 53, 54, 55 y 56.

- 1010100110

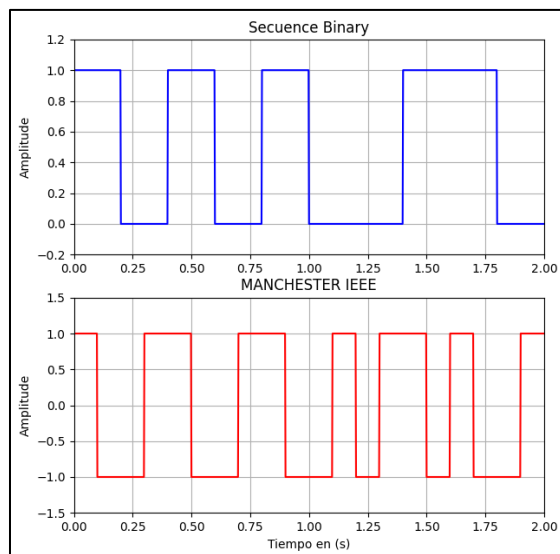


Figura 53 Prueba Manchester 1010100110

- 1011001010

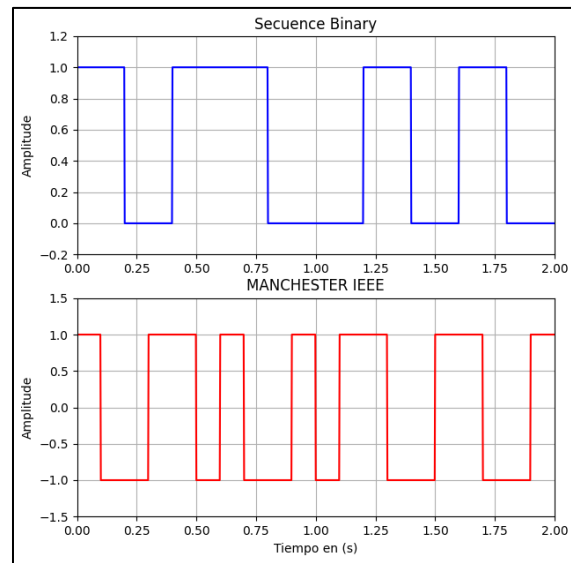


Figura 54 Prueba Manchester 1011001010

- 1101100010

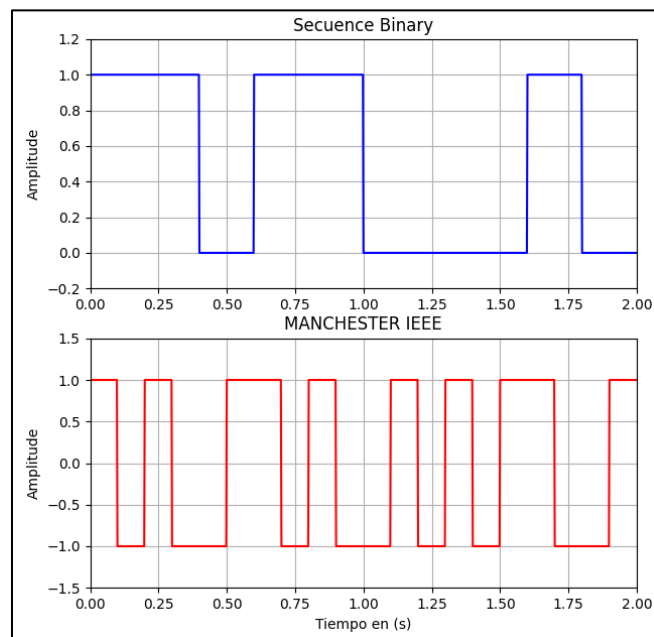


Figura 55 Prueba Manchester 1101100010

- 1101010001

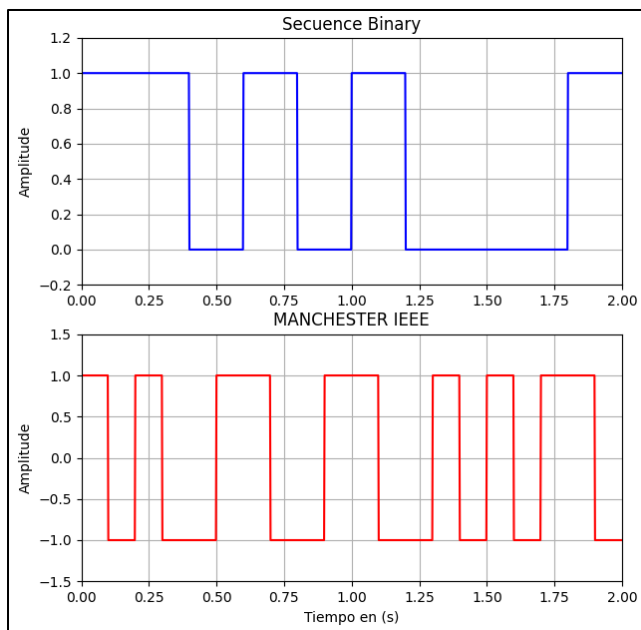


Figura 56 Prueba Manchester 1101010001

12.4.2.5 URZ. Los resultados de este código de línea se aprecian en las figuras 57, 58, 59 y 60.

- 1010100110

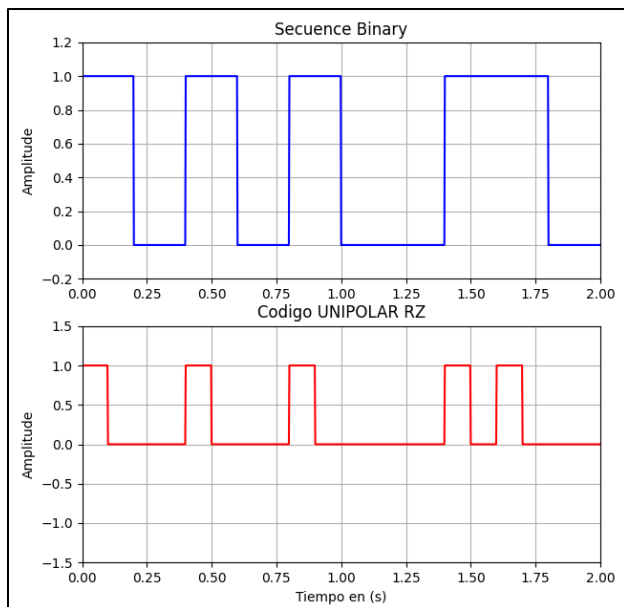


Figura 57 Prueba URZ 1010100110

- 1011001010

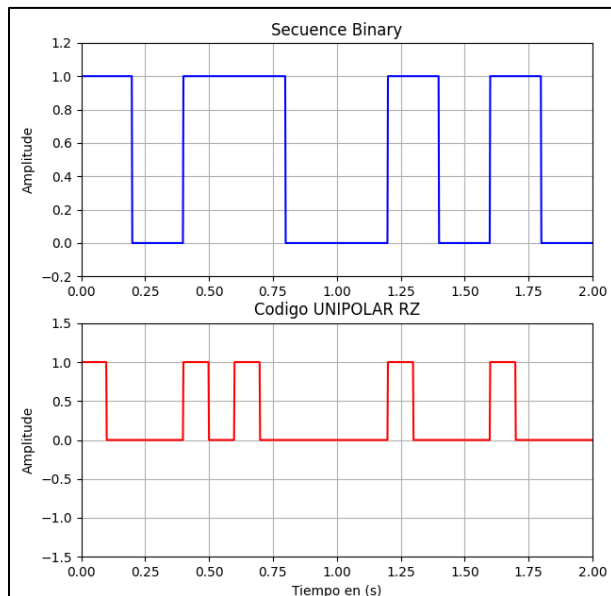


Figura 58 Prueba URZ 1011001010

- 1101100010

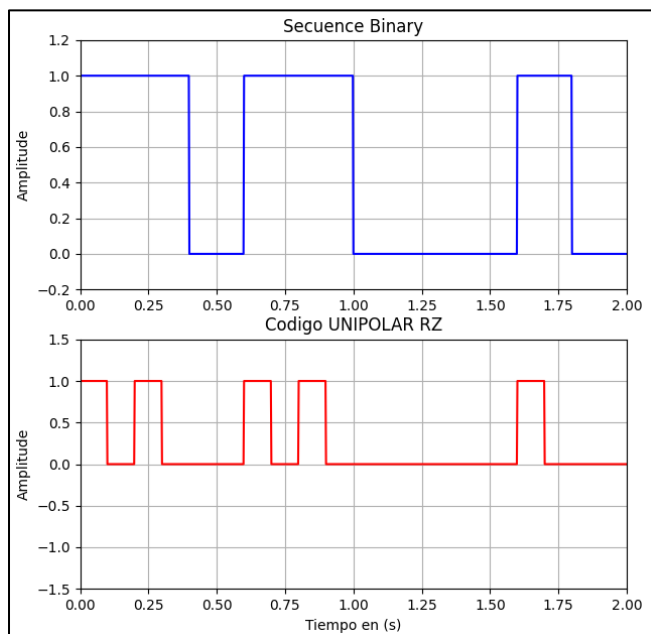


Figura 59 Prueba URZ 1101100010

- 1101010001

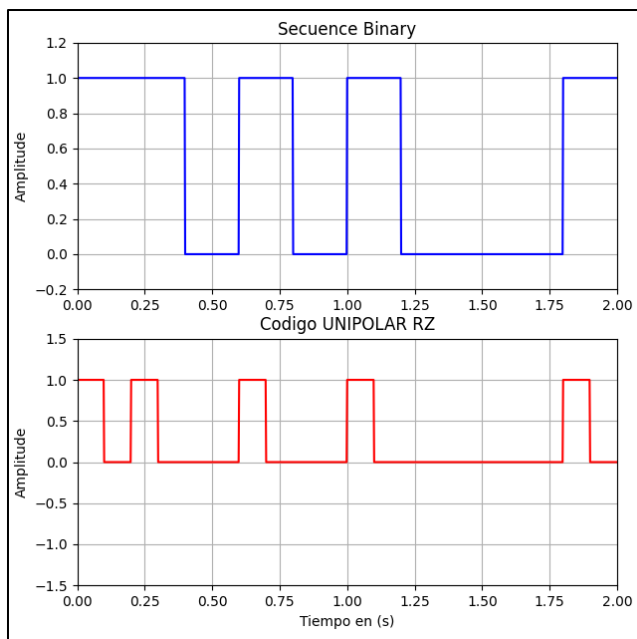


Figura 60 Prueba URZ 1101010001

12.4.2.6 UNRZ. Los resultados de este código de línea se aprecian en las figuras 61, 62, 63 y 64.

- 1010100110

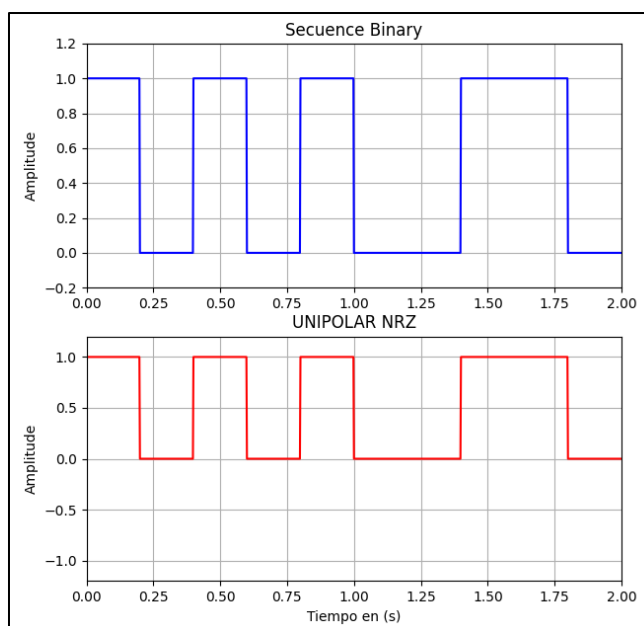


Figura 61 Prueba UNRZ 1010100110

- 1011001010

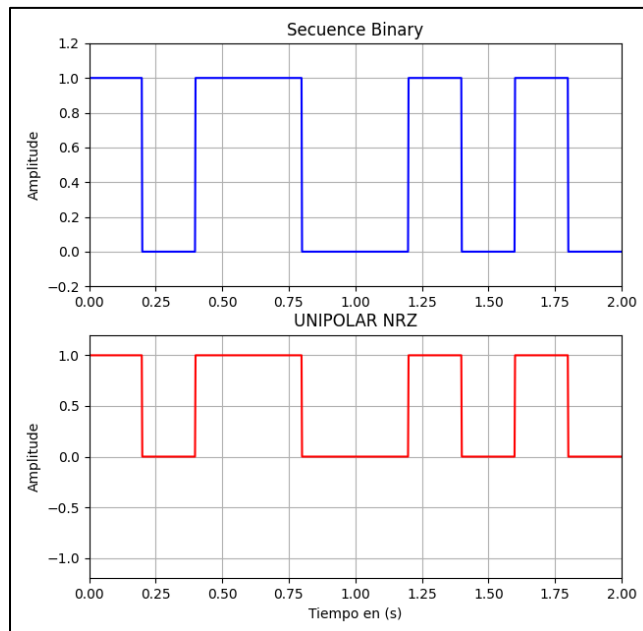


Figura 62 Prueba UNRZ 1011001010

- 1101100010

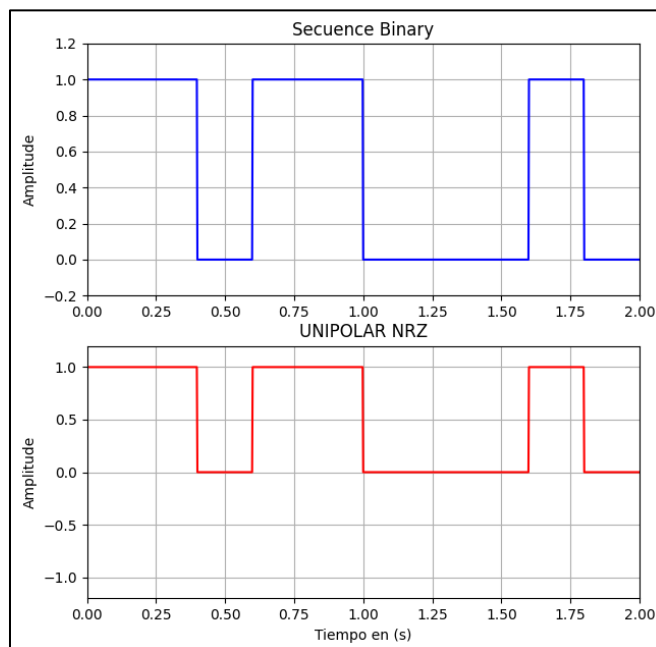


Figura 63 Prueba UNRZ 1101100010

- 1101010001

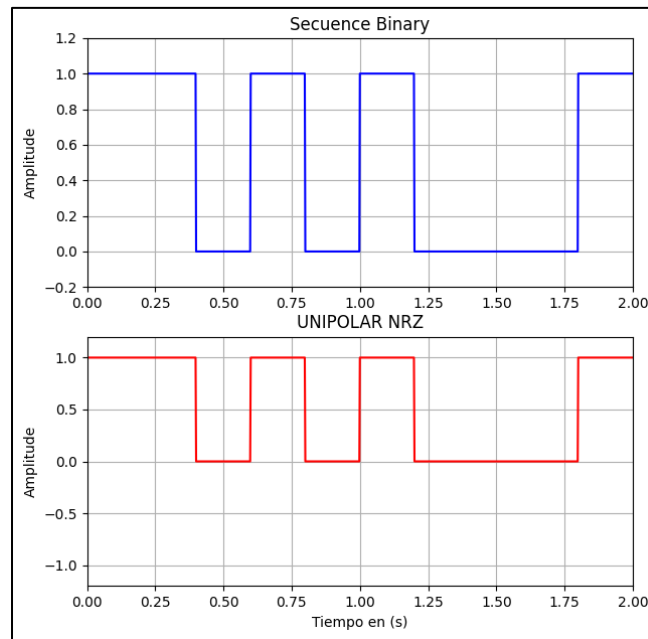


Figura 64 Prueba UNRZ 1101010001

12.4.2.7 BFSK. Los resultados de esta modulación digital se aprecian en las figuras 65, 66 y 67.

- **1 0 1 1 0 0 1 0 1 0**
- **Frecuencia (Hz): 5**
- **Amplitud(V): 10**

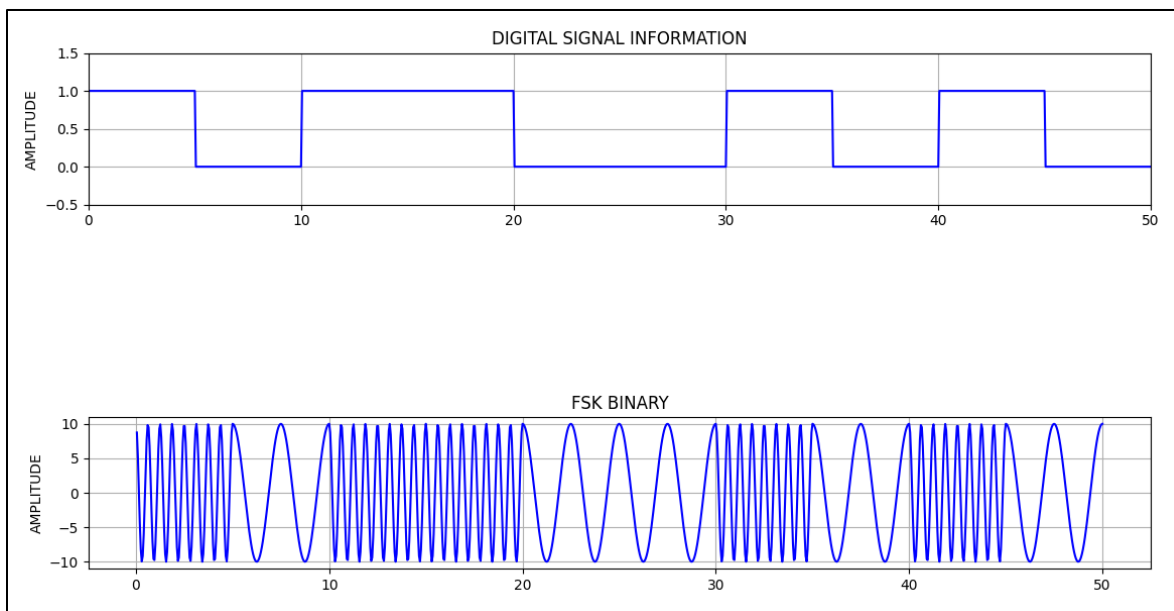


Figura 65 Prueba BFSK 1011001010

- **1 1 0 1 1 0 0 0 1 0**
- **Frecuencia (Hz): 5**
- **Amplitud(V): 10**

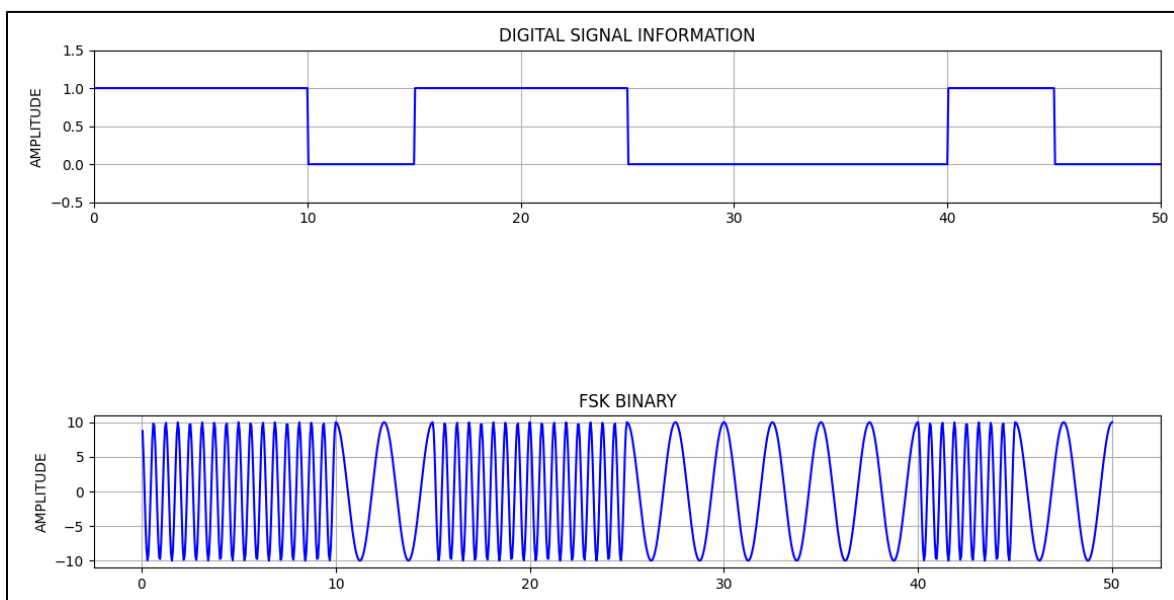


Figura 66 Prueba BFSK 1101100010

- **1 1 0 1 0 1 0 0 0 1**
- **Frecuencia (Hz): 5**
- **Amplitud(V): 10**

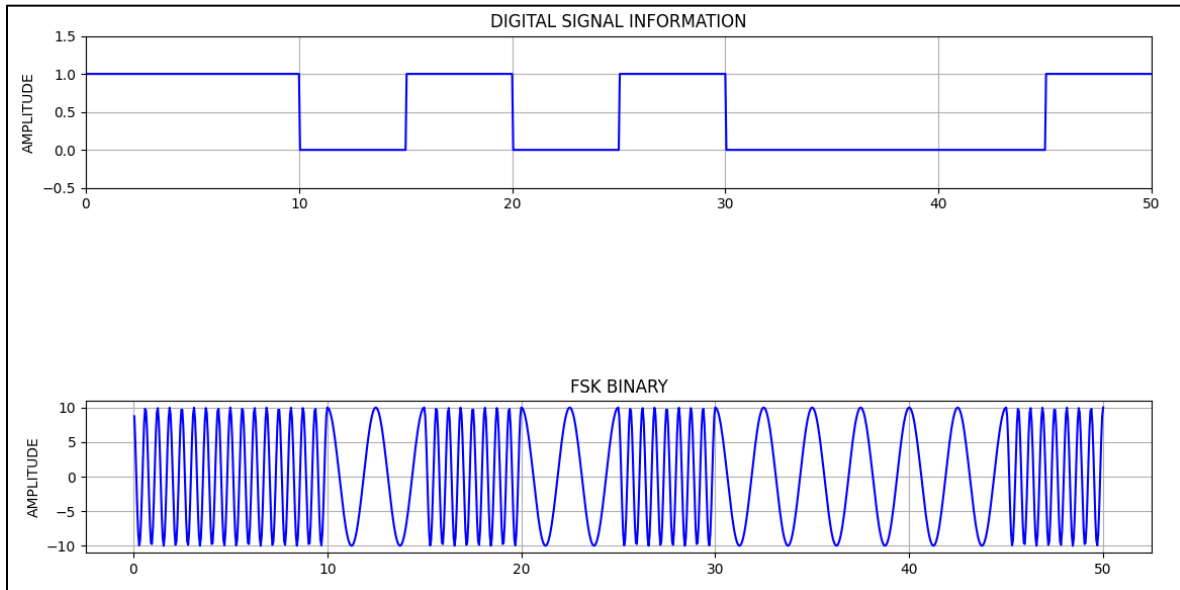


Figura 67 Prueba BFSK 1101010001

12.4.1.8 BPSK. Los resultados de esta modulación digital se aprecian en las figuras 68, 69 y 70.

- **1 0 1 1 0 0 1 0 1 0**
- **Frecuencia (Hz): 5**
- **Amplitud(V): 10**

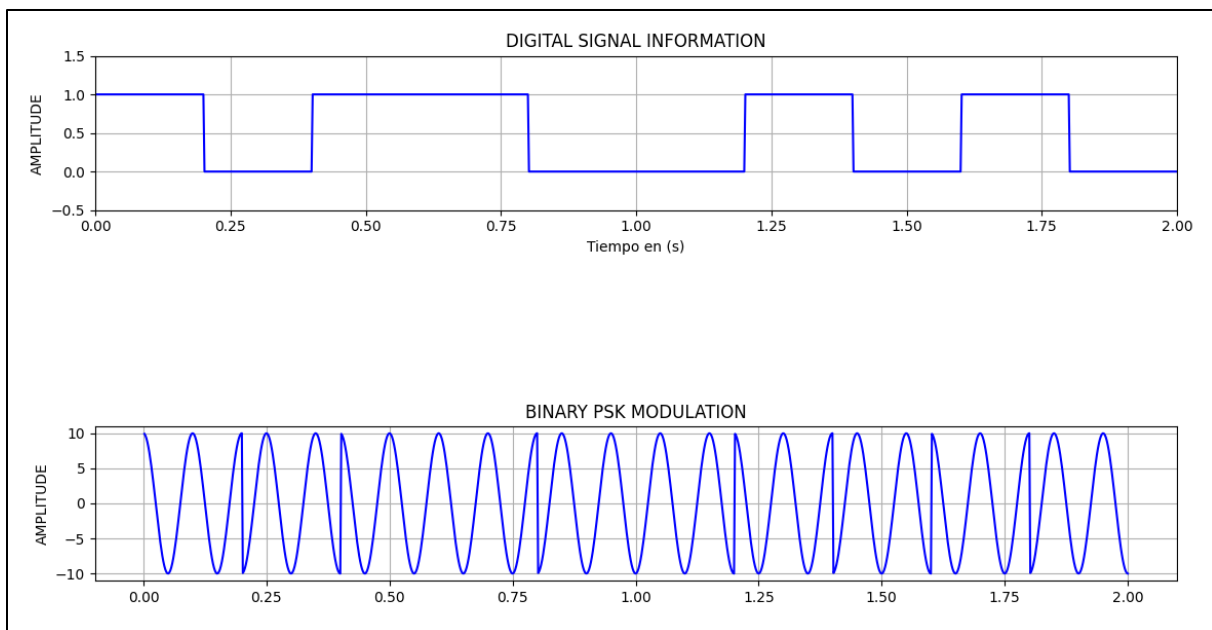


Figura 68 Prueba BPSK 1011001010

- **1 1 0 1 1 0 0 0 1 0**
- **Frecuencia (Hz): 5**
- **Amplitud(V): 10**

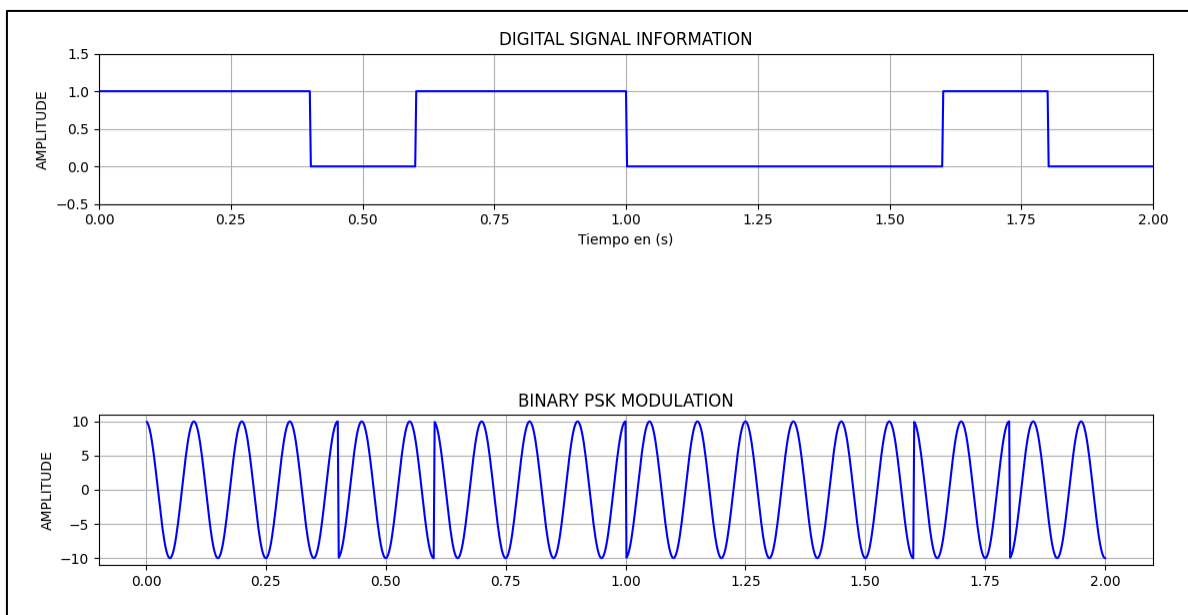


Figura 69 Prueba BPSK 1101100010

- **1 1 0 1 0 1 0 0 0 1**
- **Frecuencia (Hz): 5**
- **Amplitud(V): 10**

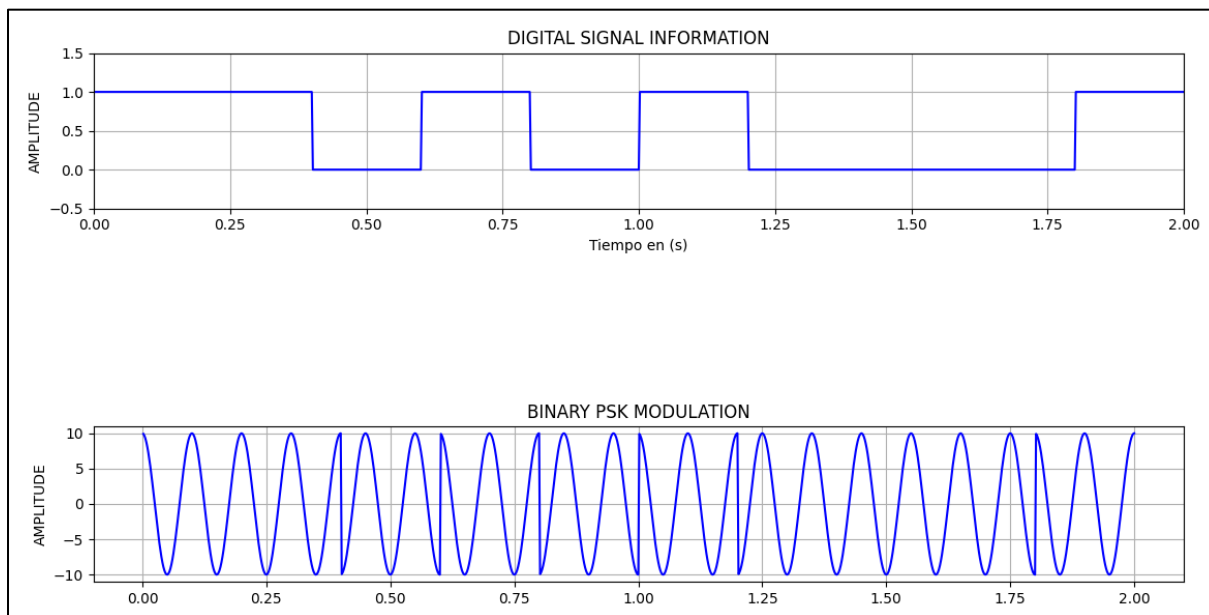


Figura 70 Prueba BPSK 1101010001

12.4.2.9 QPSK. Los resultados de esta modulación digital se aprecian en las figuras 71, 72, 73 y 74.

- **1010100110**

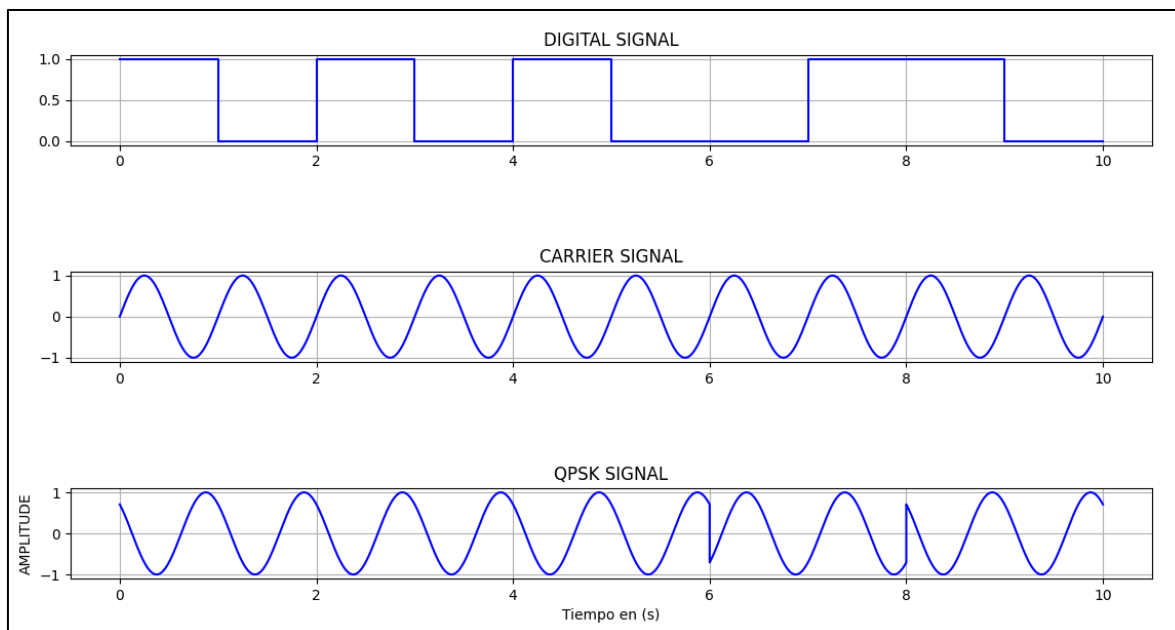


Figura 71 Prueba QPSK 1010100110

- **1011001010**

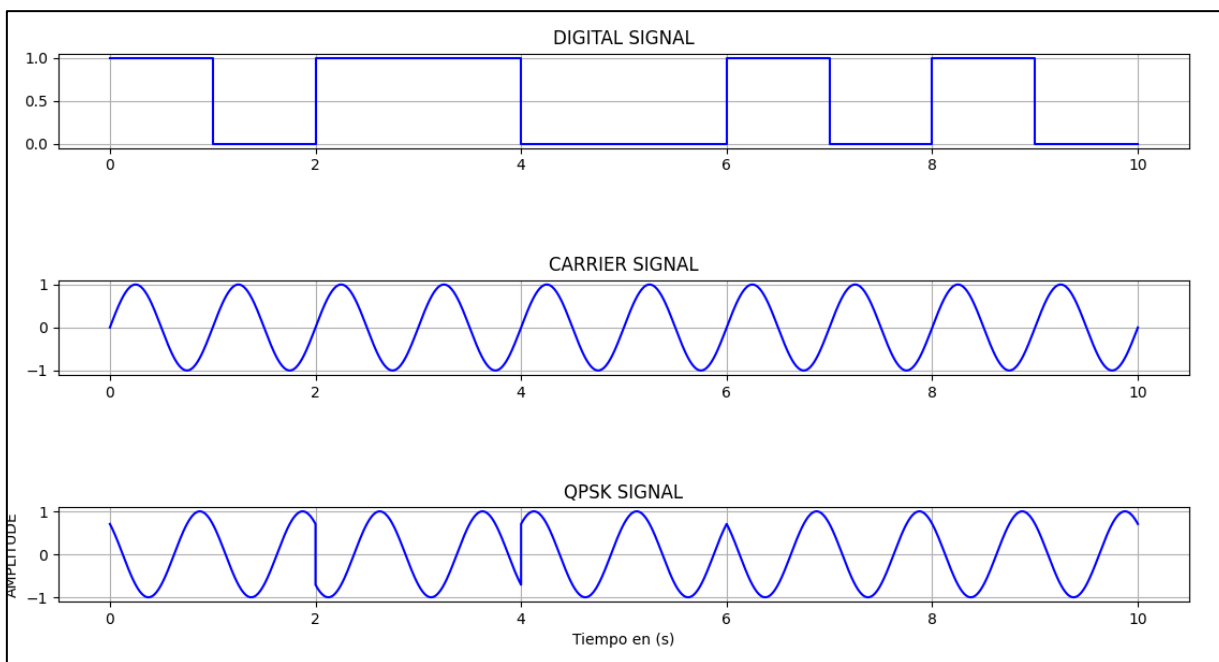


Figura 72 Prueba QPSK 1011001010

- **1101100010**

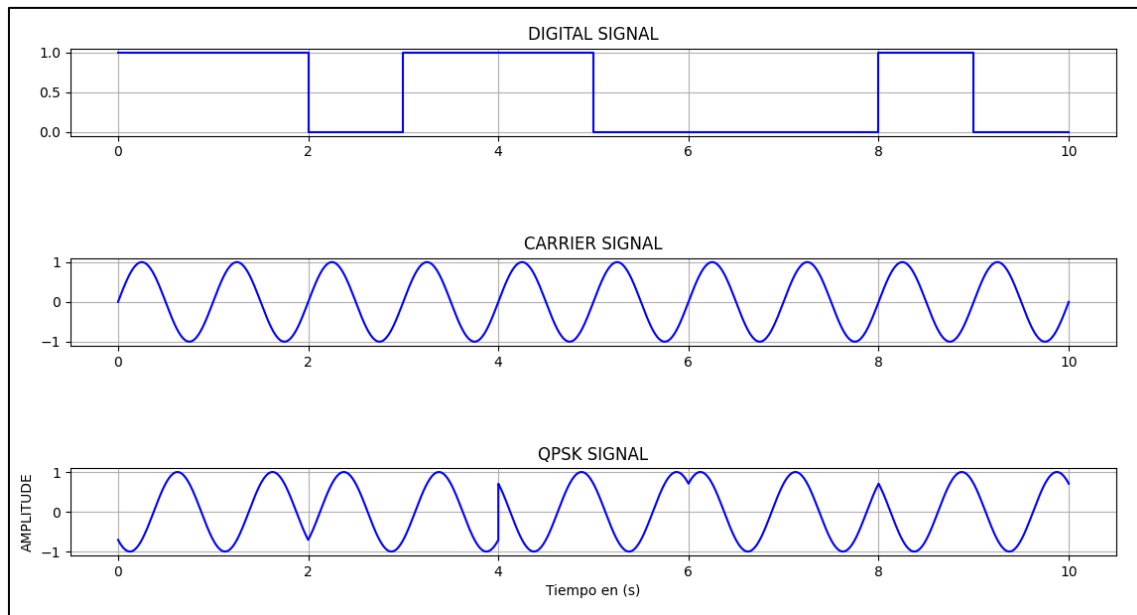


Figura 73 Prueba QPSK 1101100010

- **1101010001**

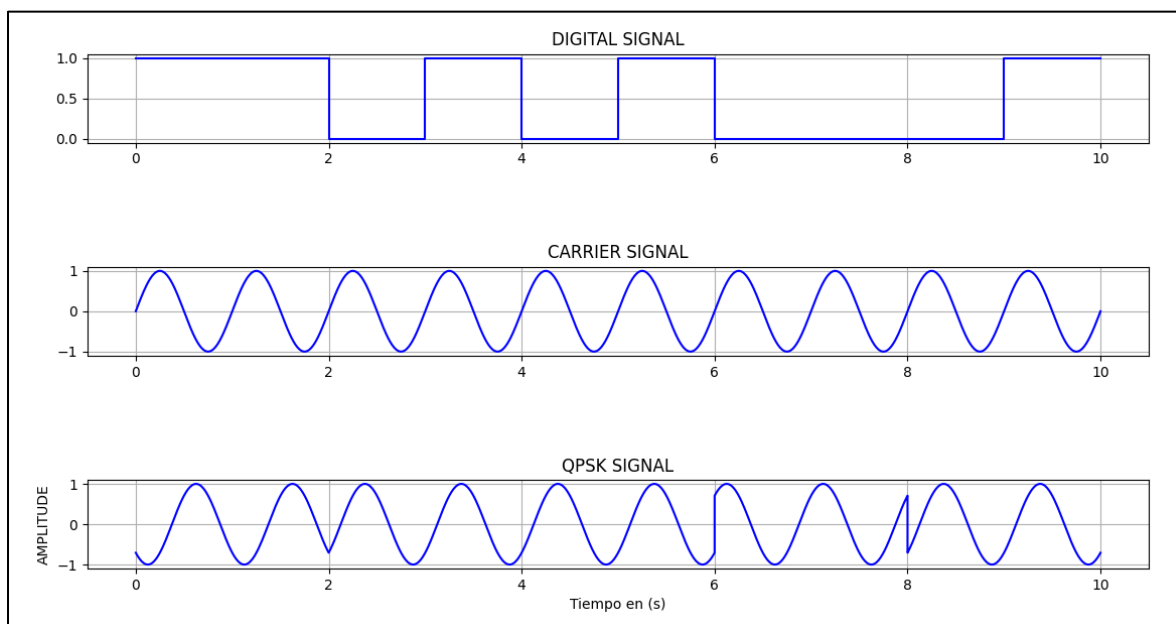



Figura 74 Prueba QPSK 1101010001


12.5 Productos

12.5.1 Guías de laboratorio

En las figuras 75, 76, 77, 78 y 79 se desarrolla una guía modelo con el propósito de que sea considerada para la renovación de las prácticas de laboratorio utilizando la herramienta desarrollada.



**Universidad Francisco
de Paula Santander**
Vigilada Mineducación



**Ingeniería
Electrónica**

Guía de laboratorio – MODULACIÓN BANDA BASE

Nombre: _____ Fecha: _____

1. Objetivo

Conocer el concepto de modulación banda base también llamado códigos de línea y los tipos de codificación que se encuentran, utilizando la herramienta computacional VIRTUAL LABORATORY OF DIGITAL COMMUNICATIONS SYSTEMS (VLDS).

2. Descripción

VLDS es una herramienta computacional que se desarrolló con el uso del lenguaje de programación Python, implementado en el Ide Spyder, entorno científico escrito en Python para Python, enfocado en el análisis de datos, permitiendo una visualización agradable de los procesamientos gráficos implementados. La interfaz gráfica se compone de dos módulos principales, uno dedicado a la codificación en línea y otro a la modulación de señales digitales.

2.1 Modulación banda base

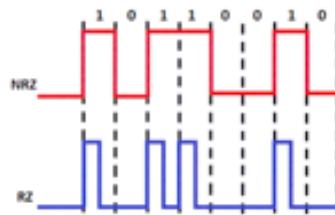
Los códigos de línea son técnicas que se aplican sobre una señal binaria con la finalidad de hacerla compatible con el medio alámbrico. Esto implica a que la señal binaria deba pasar por un proceso de modulación. Estos códigos son una forma de transmisión banda base, en donde "1" y "0" se pueden representar en diversos formatos de señalización serial de bit [1].

2.2 Modulaciones NRZ Y RZ:

Por sus siglas Non Return Zero y Return Zero, son fundamentales en la transmisión de información en banda base.

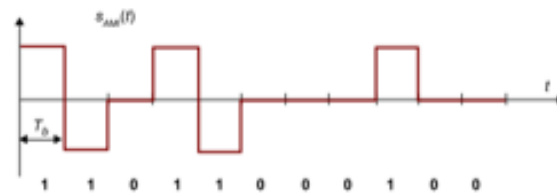
Para el caso de la modulación NRZ se asigna un nivel de tensión positivo cuando el bit a transmitir es 1 y un nivel de tensión negativo cuando el bit a transmitir es un 0. Para la modulación RZ se asigna de la misma manera que NRZ, pero, una vez transmitido el bit, debe retornar a la tensión de 0 antes de producir el bit siguiente.

Figura 75 Guía de laboratorio



2.3 Modulación AMI:

Es una modulación bipolar también denominada marca alternada AMI (Alternate Mark Inversión), se asigna una tensión cero a un bit cero y tensión positiva y negativa, cuando se transmite un bit 1.



3. Trabajo Previo

1. Definir las siguientes funciones y explique cómo se construye cada señal binaria.

- AMI RZ
- AMI NRZ
- B RZ
- B NRZ
- U RZ
- U NRZ
- MANCHESTER

2. ¿Qué es BIT RATE?

Figura 76 Guía de laboratorio

4. Funcionamiento de la Interfaz

La interfaz gráfica contiene los módulos de Modulación Digital y Códigos de línea (modulación banda base). Antes de seleccionar la codificación de la señal que se quiere visualizar, se deben ingresar los datos

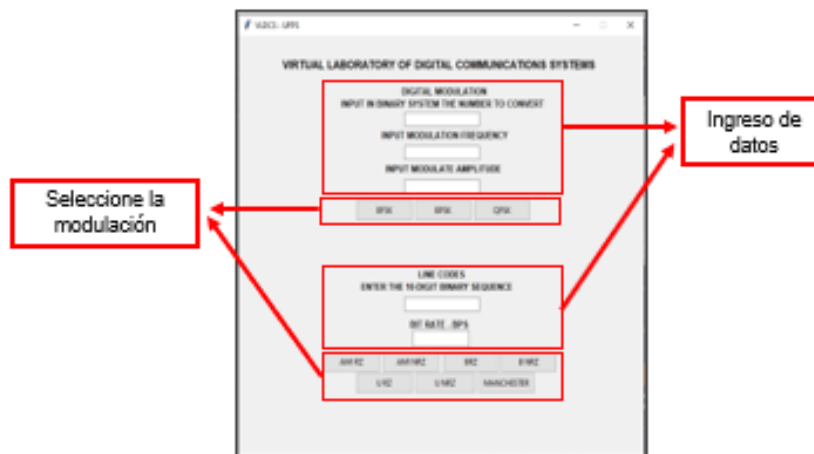


Ilustración 1 Interfaz gráfica de la Herramienta desarrollada

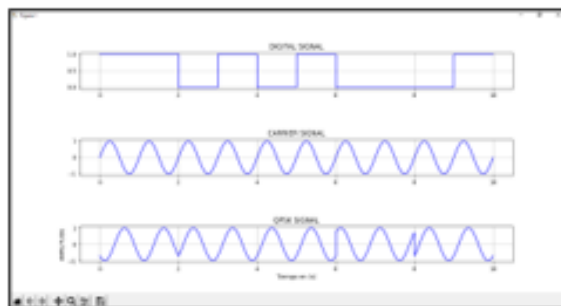


Ilustración 2 Visualización de gráficas

Figura 77 Guía de laboratorio

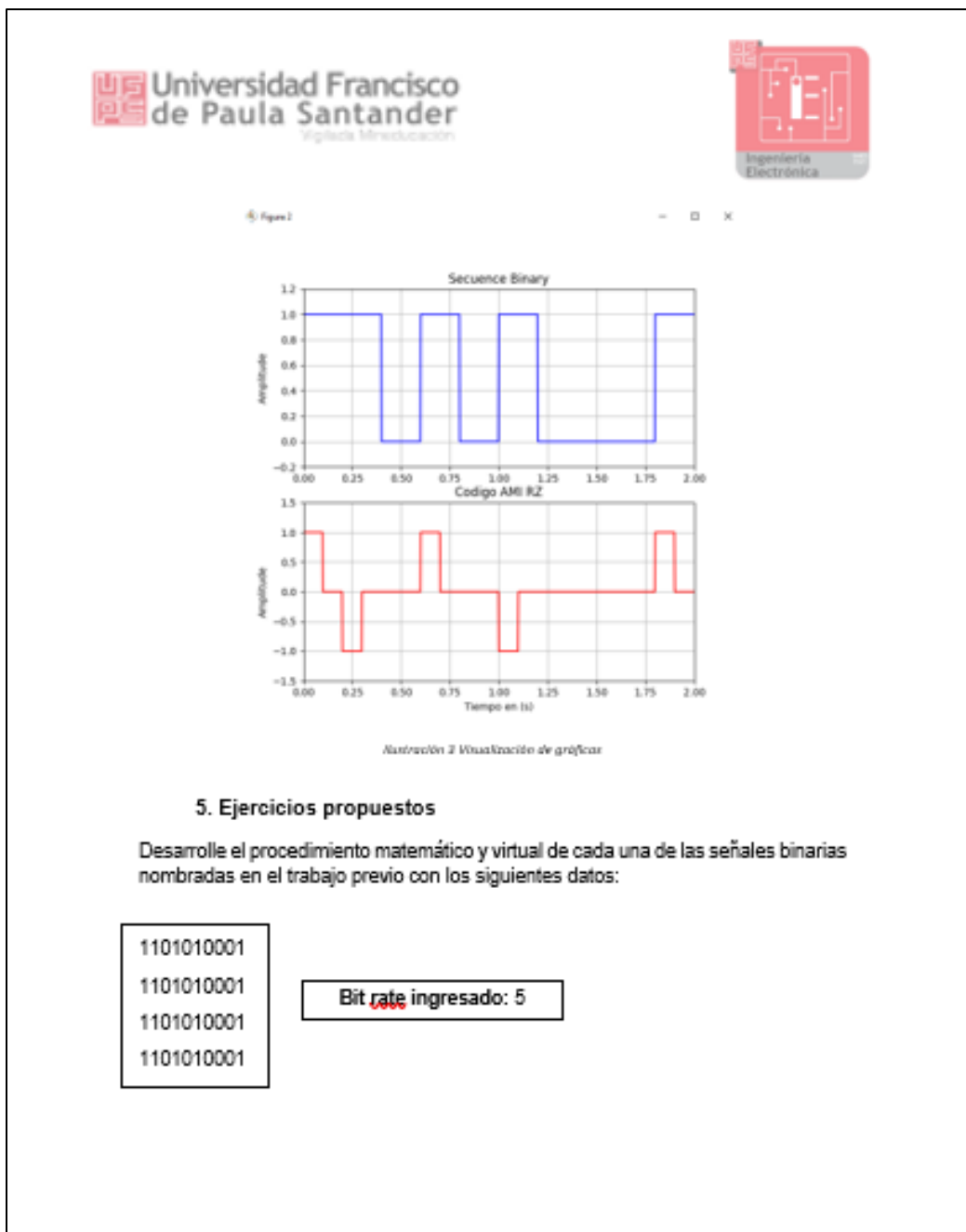


Figura 78 Guía de laboratorio

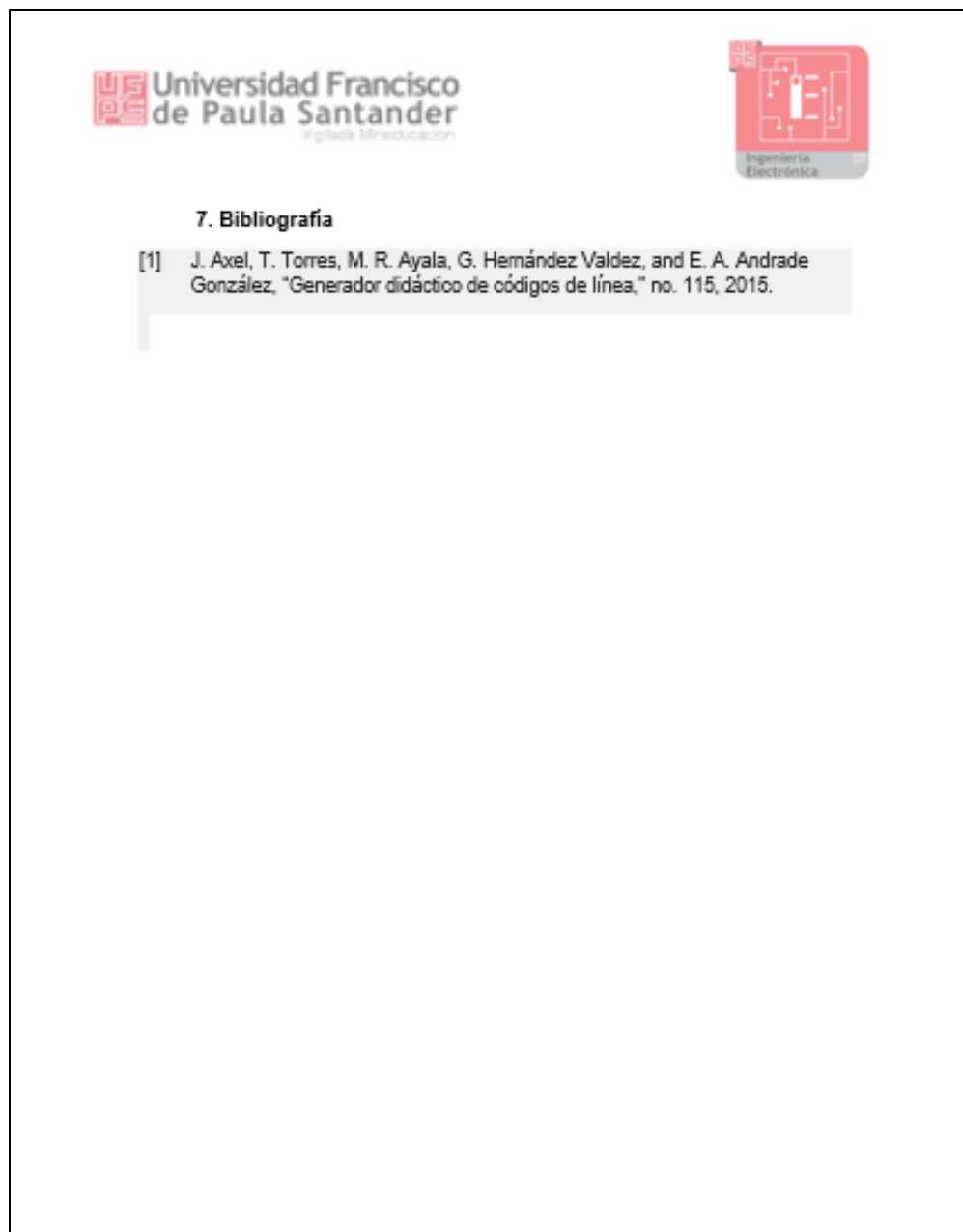


Figura 79 Guía de laboratorio

12.5.2 Divulgaciones

- Ponencia 1: VI Encuentro Internacional de Educación Matemáticas 19 Y 20 de mayo 2022. Esta ponencia es enfocada en la metodología. Su certificado de participación se observa en la figura 80.



Figura 80 Certificado Potencia 1

- Ponencia 2: VII Encuentro Regional de Semilleros de Investigación, realizado los días 25, 26, 27 y 28 de mayo 2022. La segunda ponencia se enfocó en la interfaz de la herramienta, la cual se socializó en el encuentro regional de semilleros de investigación donde se obtuvo un puntaje de 90 con el cual se otorgó la oportunidad de asistir al encuentro nacional. Su certificado de participación se observa en la figura 81.



Figura 81 Certificado Ponencia 2

- Ponencia 3: XXV Encuentro Nacional y XXIV Encuentro Internacional de Semilleros de Investigación Fundación REDCOLSI del 12 al 15 de octubre del 2022. En este evento se participo con la interfaz gráfica completa y en funcionamiento. Su certificado de participación se observa en la figura 82.



LA FUNDACIÓN RED COLOMBIANA DE SEMILLEROS DE INVESTIGACIÓN RedCOLSI

Del 12 al 15 de octubre de 2022

Hace constar que,

ANGELICA ROCIO MEJIA SERRANO

Con número de identificación **1090467406**

Participó en calidad de **PONENTE**

Con el proyecto **HERRAMIENTA EDUCATIVA PARA EL APRENDIZAJE DE SISTEMAS DE COMUNICACIONES II UTILIZANDO SOFTWARE LIBRE.**


LUZ MERY HERRERA GALEANO
 Coordinadora Nacional de RedCOLSI


MARTHA LUCIA MONSALVE PERDOMO
 Secretaria Nacional de RedCOLSI


DANIEL CARVAJAL TABARES
 Coordinador Nodo Antioquia RedCOLSI

ORGANIZA:



ANFITRIONES:



ALIANZA INTERNACIONAL:



Figura 82 XXV Encuentro Nacional REDCOLSI

13. Conclusiones

Existe avance en la aplicación de herramientas computacionales para la educación, sin embargo, estas se enfocan en medir el grado de aprendizaje y la afectación cognitiva que podría traer en los estudiantes, en este proyecto se priorizó la facilidad de un laboratorio virtual contra las prácticas físicas que acarrearán mayor gasto en logística, personal e inversión.

La herramienta computacional permite la apropiación de conocimientos teórico – prácticos en modulación digital y modulación en banda base, permitiendo avanzar en temas que requieran más consumo del recurso intelectual humano en el área de las comunicaciones digitales.

La implementación de un lenguaje de desarrollo de software libre como Python y la implementación de librerías que son diseñadas y asistidas por una comunidad global de científicos e investigadores, permite la adecuación y configuración de la herramienta computacional en requerimientos futuros.

Los tiempos de respuesta son distintos cuando se ejecuta en un IDE especializado y la máquina computacional que aloja la aplicación de escritorio, la notación big O indica variaciones de aproximadamente 1,5 hasta 2,7 segundos de diferencia en el procesamiento y generación de gráficos.

14. Recomendaciones y trabajo futuro

La herramienta computacional fue diseñada para el aprendizaje práctico de Modulación por desplazamiento de frecuencia, desplazamiento de fase y desplazamiento por cuadratura, sin embargo, no se incluyeron módulos para el diseño de sistemas de comunicaciones, por tanto, estos pueden implementarse en trabajos futuros.

Las pruebas con la aplicación de escritorio (Laboratorio Virtual Para Comunicaciones Digitales) se realizaron con las siguientes características técnicas: procesador Intel Core i5 con dos núcleos de procesamiento, frecuencia básica de operación 2,5 GHz, disponibilidad de 15,9 Gb de memoria RAM y en sistema operativo Windows 10 de 64 bits.

El desarrollo del laboratorio virtual basado en Python, permitiría su migración a aplicación web, facilitando el acceso y aprendizaje de quienes lo requieran.

Para las modulaciones digitales M-arias, se cuenta con funciones específicas en lenguajes como Matlab, sin embargo, se recomienda incluir en el diseño de la herramienta en Python.

El análisis espectral puede ser implementado para una mayor apreciación en las variaciones de frecuencia y sus afectaciones en modulación.

15. Referencias

- [1] R. E. Briones and Y. M. S. Claudio, “Regional telecommunications infrastructure: A multivariate analysis,” *Trimestre Economico*, vol. 85, no. 340, pp. 765–799, Oct. 2018, doi: 10.20430/ete.v85i340.537.
- [2] K. Puerto, D. Roperro-Torres, and D. Guevara, “Evaluación de un sistema de comunicación óptico empleando modulación por desplazamiento de fase QPSK, 8PSK y 16PSK utilizando la técnica FDM,” *Aibi revista de investigación, administración e ingeniería*, pp. 76–83, Aug. 2020, doi: 10.15649/2346030x.709.
- [3] M. Cabrera, F. Tarrés, R. Revisión, F. Rey, M. Francesc, and T. Ruiz, “Comunicaciones digitales en banda base.”
- [4] M. A. Rodríguez Hernandez, “Modulación digital binaria FSK (BFSK),” Feb. 2011, Accessed: Oct. 17, 2022. [Online]. Available: <https://riunet.upv.es:443/handle/10251/9332>
- [5] I. J. Roberto, S. Sandoval, and I. E. Chanto, “Aplicación de la virtualidad en la enseñanza de la ingeniería: Caso de estudio Ingeniería en Telecomunicaciones en la UNED de Costa Rica Application of virtuality in engineering education: Case study Telecommunications Engineering at UNED in Costa Rica.”
- [6] V. F. Barrera Rea and A. Guapi Mullo, “La importancia del uso de las plataformas virtuales en la educación superior,” *Atlante Cuadernos de Educación y Desarrollo*, no. julio, Jul. 2018.
- [7] E. D. carmen Cruz Rodriguez, “Importancia del manejo de competencias tecnológicas en las prácticas docentes de la Universidad Nacional Experimental de la Seguridad (UNES),” *Revista Educación*, vol. 43, pp. 196–218, 2018, doi: 10.15517/revedu.v43i1.27120.
- [8] R. Stallman, “La definición de Software libre The definition of Free Software,” pp. 151–154, 2020.
- [9] J. M. Ruiz, C. D. Pacifico, and M. M. Pérez, “Clasificación y Evaluación de Métricas de Mantenibilidad Aplicables a Productos de Software Libre,” *In XIX Workshop de Investigadores en Ciencias de la Computación (WICC)*, pp. 460–464, 2017.

- [10] R. Villarroel Flores, M. Tetlalmatzi Montiel, and E. E. Rodríguez Torres, “Visualización de series de tiempo en Python,” *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, vol. 6, no. 11, p. 5, 2018, doi: 10.29057/icbi.v6i11.3021.
- [11] R. Berrones, “Trabajo De Titulación En Opción Al Grado De : Ingeniero En Electrónica Digital Y,” p. 167, 2019.
- [12] H. Mendoza and J. Antonio, “Método para la enseñanza de software libre en ciencias socioeconómicas Actividades Semestre 1 Metodología de investigación Actividades Semestre 2 Metodología de investigación Actividades Semestre 3 Actividades profesor : Metodología de investigación . de ,” pp. 1–12.
- [13] C. A. P. Torres, “CRITERIA CONSIDERED IN THE SELECTION OF FREE SOFTWARE FOR TEACHING AND,” vol. 5, no. 2, pp. 46–54, 2020.
- [14] N. Poveda, “La enseñanza de matemáticas en la recuperación pedagógica fundamentada en las Tics de software libre, aplicado en talleres interactivos de radicación y potenciación,” 2017.
- [15] D. Cepeda-Holguin and J. Bacca, “Aplicación móvil para la enseñanza de la programación sobre el lenguaje Python,” *Revolución en la Formación y la Capacitación para el Siglo XXI (Vol. II)*, no. June, pp. 925–933, 2019, doi: 10.5281/zenodo.3524363.
- [16] A. Artés Rodríguez, F. Pérez González, R. López Valcarce, C. Mosquera Nartallo, F. Pérez Cruz, and J. Cid Sueiro, “Comunicaciones digitales,” pp. 1–790, 2012.
- [17] R. M. DE ANDA-LÓPEZ, “MatLab y sus toolboxes como herramientas IoT para facilitar el desarrollo y diseño de sistemas mecatrónicos: Migrando hacia la I 4.0,” *CIERMMI Mujeres en la ciencia T.I*, pp. 15–27, 2019, doi: 10.35429/h.2019.1.15.27.
- [18] E. Lerma, R. Griñó, R. Costa-Castelló, and C. Sanchis, “Implementación de controladores en Arduino mediante Simulink,” pp. 158–164, 2020, doi: 10.17979/spudc.9788497497565.0158.
- [19] A. Marzal Varó, I. Gracia Luengo, and P. García Sevilla, “Introducción a la programación con Python 3,” *Introducción a la programación con Python 3*, no. November, 2014, doi: 10.6035/sapientia93.
- [20] “Spyder, un potente entorno de desarrollo interactivo para Python | Uibunlog.” <https://ubunlog.com/spyder-entorno-desarrollo-python/> (accessed Feb. 23, 2021).

- [21] “Matplotlib: trazado de Python - documentación de Matplotlib 3.3.4.” <https://matplotlib.org/stable/index.html> (accessed Feb. 23, 2021).
- [22] “SDR y GNU Radio como plataforma para un laboratorio de comunicaciones digitales.” <https://www.redalyc.org/jatsRepo/5122/512261374007/html/index.html> (accessed Feb. 23, 2021).
- [23] Congreso de la República de Colombia, “Ley No.1341 30 julio,” *Ley No.1341 30 julio*, pp. 1–34, 2009.
- [24] “1. Ley De Software Libre En Colombia - Unicesar-legislación.” <https://sites.google.com/site/unicesarlegislacion/Inf/home/Ley-De-Software-Libre-En-Colombia> (accessed Feb. 25, 2021).
- [25] C. Torres Gómez and J. S. Torres Sandoval, “Seguridad en Redes con Python,” 2022, Accessed: Nov. 27, 2022. [Online]. Available: <http://repositorio.uts.edu.co:8080/xmlui/handle/123456789/8414>
- [26] J. Alberto, D. Navarro, M. Ricaurte, V. Miryam, and J. Salcedo Ramírez, “‘Fases de diseño de arquitectura, construcción de Backend y pruebas para la migración al lenguaje de programación Python del proyecto ‘web cliente’ en una compañía de asistencias’ Autores”.
- [27] J. Flores Pérez, “Detector de escenas acústicas en Python,” 2021.
- [28] C. L. Vidal-Silva, A. Sánchez-Ortiz, J. Serrano, and J. M. Rubio, “Experiencia académica en desarrollo rápido de sistemas de información web con Python y Django Academic experience in rapid development of web information systems with Python and Django,” vol. 14, no. 5, pp. 85–94, 2021, doi: 10.4067/S0718-50062021000500085.
- [29] “El tutorial de Python — documentación de Python - 3.11.0.” <https://docs.python.org/es/3/tutorial/index.html> (accessed Nov. 27, 2022).
- [30] B. Gissella Danelia Grillo Moreno, B. Luis Armando Lezama Barrientos, and B. Ernesto de Jesús Ordóñez González, “Universidad Nacional Autónoma de Nicaragua ‘A la Libertad por la Universidad.’”
- [31] C. Roldán Blay, “Configuración de Spyder para programar aplicaciones de escritorio con Tkinter en Python,” 2022, Accessed: Nov. 27, 2022. [Online]. Available: <https://riunet.upv.es/handle/10251/185012>
- [32] I. Journal, M. S. Khoirom, M. Sonia, B. Laikhuram, J. Laishram, and D. Singh, “IRJET- Comparative Analysis of Python and Java for Beginners Cite this paper Comparative

- Analysis of Python and Java for Beginners,” *International Research Journal of Engineering and Technology*, 2020, Accessed: Nov. 27, 2022. [Online]. Available: www.irjet.net
- [33] Mark. Roseman, “Modern Tkinter for busy Python developers : quickly learn to create great looking user interfaces for Windows, Mac and Linux using Python’s standard GUI toolkit,” p. 257, 2020.
- [34] “NumPy - Community.” <https://numpy.org/community/> (accessed Nov. 27, 2022).
- [35] “matplotlib.pyplot — Matplotlib 3.5.3 documentation.” https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html (accessed Nov. 27, 2022).
- [36] “random — Generar números pseudoaleatorios — documentación de Python - 3.11.0.” <https://docs.python.org/3/library/random.html> (accessed Nov. 27, 2022).
- [37] Wayne. Tomasi, G. Mata Hernández, and V. González Pozo, *Sistemas de comunicaciones electrónicas*. Pearson Educación, 2003.
- [38] Mark. Roseman, “Modern Tkinter for busy Python developers : quickly learn to create great looking user interfaces for Windows, Mac and Linux using Python’s standard GUI toolkit,” p. 257, 2020.
- [39] “matplotlib.pyplot — Matplotlib 3.5.3 documentation.” https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html (accessed Nov. 11, 2022).
- [40] “Creación de arreglos — NumPy v1.23 Manual.” <https://numpy.org/doc/stable/user/basics.creation.html> (accessed Nov. 11, 2022).

16. Anexos

- **ANEXO 1: MODULACIÓN EN BANDA BASE**

#AMI-NRZ

```
def Amirz():

    inicio = time.time()

    #Ingresamos parametros para la señal

    #velocidad binaria

    Vb = int(bitRate.get())

    A = 1 # Amplitud es 1

    tb=1/int(Vb);#Tiempo de bit

    tau=tb;#Ancho de pulso tau

    BWs=1/tau;#Ancho de banda

    paso=tb/100;#Tiempo de paso

    t=np.arange(0,tb,paso);

    s1=np.ones(len(t),dtype=int);

    s0=np.zeros(len(t));

    #Secuencia aleatoria de bits

    n=10;

    #x=np.random.randint(2,size=n);

    x1 = Entrada.get()

    x2 = []

    for i in range(0,len(x1),1):

        x2.append(int(x1[i:i+1]))
```



```

if len(x2) < 10:
    messagebox.showinfo(message = "BINARY SEQUENCE MUST BE 10 DIGITS -
RANDOM DATA WILL BE GENERATED", title ="MESSAGE")

x = x=np.random.randint(2,size=n)

secuencia = [];

for i in range(len(x)):
    if x[i]==1:
        secuencia=np.append(secuencia,s1*int(A));
    else:
        secuencia=np.append(secuencia,s0);

t1=(len(secuencia))*paso;
tt=np.arange(0,t1,paso);
plt.figure(figsize=(7,8));
plt.subplot(2,1,1)
plt.plot(tt,secuencia,color='blue')
plt.axis([0,t1,-0.2*int(A),1.2*int(A)]);
plt.grid(linestyle='-')
plt.title('Secuence Binary');
#plt.xlabel("Tiempo en (s)");
plt.ylabel("Amplitude")

#Codigo de linea AMI RZ

```

```

tbami=tb;#Tiempo de bit para AMI NRZ igual al tiempo de bit de la señal inicial
tauami=tb/2;#Ancho de pulso tau
BWsami=1/tauami;#Ancho de banda
tbm=tb/2;
tm=np.arange(0,tbm,paso);
secuenciaM=[];
j=0;
for i in range(n):
    if x[i]==1:
        j=j+1;
        if j%2==0:
            s1M=-int(A)*np.ones(len(tm));
            s1M=np.append(s1M,np.zeros(len(tm)));
        else:
            s1M=int(A)*np.ones(len(tm));
            s1M=np.append(s1M, np.zeros(len(tm)));
        secuenciaM=np.append(secuenciaM,s1M);
    else:
        s0M=np.zeros(len(tm));
        s0M=np.append(s0M,np.zeros(len(tm)));
        secuenciaM=np.append(secuenciaM,s0M);
t2=(len(secuenciaM))*paso;
tt2=np.arange(0,t2,paso);

```

```
plt.subplot(2,1,2);

plt.plot(tt2,secuenciaM,color='red')

plt.axis([0,t1,-1.5*int(A),1.5*int(A)]);

plt.grid(True)

plt.title('Codigo AMI RZ');

plt.xlabel('Tiempo en (s)');

plt.ylabel("Amplitude")

plt.show()

else:

    x = x2[0:10]

    #print("Secuencia binaria: " + str(x));

    secuencia = [];

    for i in range(len(x)):

        if x[i]==1:

            secuencia=np.append(secuencia,s1*int(A));

        else:

            secuencia=np.append(secuencia,s0);

    t1=(len(secuencia))*paso;

    tt=np.arange(0,t1,paso);

    plt.figure(figsize=(7,8));

    plt.subplot(2,1,1)
```

```

plt.plot(tt,secuencia,color='blue')

plt.axis([0,t1,-0.2*int(A),1.2*int(A)]);

plt.grid(linestyle='-')

plt.title('Secuence Binary');

#plt.xlabel('Tiempo en (s)');

plt.ylabel("Amplitude")

#Codigo de linea AMI RZ

tbami=tb;#Tiempo de bit para AMI NRZ igual al tiempo de bit de la señal inicial

tauami=tb/2;#Ancho de pulso tau

BWsam=1/tauami;#Ancho de banda

tbm=tb/2;

tm=np.arange(0,tbm,paso);

secuenciaM=[];

j=0;

for i in range(n):

    if x[i]==1:

        j=j+1;

        if j%2==0:

            s1M=-int(A)*np.ones(len(tm));

            s1M=np.append(s1M,np.zeros(len(tm)));

        else:

            s1M=int(A)*np.ones(len(tm));

```

```

        s1M=np.append(s1M, np.zeros(len(tm)));
    secuenciaM=np.append(secuenciaM,s1M);
else:
    s0M=np.zeros(len(tm));
    s0M=np.append(s0M,np.zeros(len(tm)));
    secuenciaM=np.append(secuenciaM,s0M);
t2=(len(secuenciaM))*paso;
tt2=np.arange(0,t2,paso);
plt.subplot(2,1,2);
plt.plot(tt2,secuenciaM,color='red')
plt.axis([0,t1,-1.5*int(A),1.5*int(A)]);
plt.grid(True)
plt.title('Codigo AMI RZ');
plt.xlabel('Tiempo en (s)');
plt.ylabel("Amplitude")

plt.show()

fin = time.time()

tiempoej = fin-inicio

print(tiempoej)

```

- **ANEXO 2: MODULACIÓN POR DESPLAZAMIENTO DE FRECUENCIA**

```
def Bfsk():
```

```
    inicio= time.time()
```

```
valor = BinarioModular.get()
x = [int (a) for a in str(valor)]
bp = int(Frecuencia.get())
A = int(AmplitudModular.get())

bit = []

for n in x:
    if n == 1:
        se = np.ones(100, dtype=int)
    else:
        se = np.zeros(100, dtype=int)
    bit = np.append(bit, se)

t1 = np.linspace(bp/100, bp*len(x), 100*len(x))

plt.figure(figsize=(7,8));
plt.subplot(3,1,1)
plt.plot(t1,bit,color='blue')
plt.axis([0, bp*len(x), -0.5, 1.5]);
plt.grid(linestyle='-')
plt.title('DIGITAL SIGNAL INFORMATION');
plt.xlabel('Tiempo en (s)');
```

```
plt.ylabel("AMPLITUDE")

br = 1/bp

f1 = br*8

f2 = br*2

t2 = np.linspace(bp/99,bp,99)

ss = len(t2)

m = []

for i in x:

    if i == 1:

        y = A*np.cos(2*np.pi*f1*t2)

    else:

        y = A*np.cos(2*np.pi*f2*t2)

    m = np.append(m,y)

t3 = np.linspace(bp/99,bp*len(x),99*len(x))

#plt.figure(figsize=(7,8));

plt.subplot(3,1,3)

plt.plot(t3,m,color='blue')

plt.grid(linestyle='-')

plt.title('FSK BINARY');

plt.xlabel('Tiempo en (s)');
```

```
plt.ylabel("AMPLITUDE")

mn = []

for n in range(ss,len(m)+1,ss):

    t = np.linspace(bp/99,bp,99)

    y1 = np.cos(2*np.pi*f1*t)

    y2 = np.cos(2*np.pi*f2*t)

    mm =np.multiply(y1,m[n-(ss):n])

    mmm = np.multiply(y2,m[n-(ss):n])

    t4 = np.linspace(bp/99,bp,99)

    z1 = np.trapz(t4,mm)

    z2 = np.trapz(t4,mmm)

    zz1 = np.round(2*z1/bp)

    zz2 = np.round(2*z2/bp)

    if (zz1 > A/2):

        a =1

    else:

        a = 0

    mn = np.append(mn,a)

plt.show()

fin = time.time()

print(fin-inicio)
```