



**UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE INGENIERÍA INDUSTRIAL
DEPARTAMENTO ACADÉMICO DE GRADUACIÓN**

**TRABAJO DE TITULACIÓN
PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN TELEINFORMÁTICA**

**ÁREA
TECNOLOGÍA DE LAS TELECOMUNICACIONES**

**TEMA
“IMPLEMENTACIÓN EN SOFTWARE DE LOS SISTEMAS
DE MODULACIONES DIGITALES”**

**AUTOR
WILA ESPINOZA HENRY ALEXANDER**

**DIRECTOR DEL TRABAJO
ING. TELECOM. NEISER STALIN ORTÍZ MOSQUERA, MG.**

GUAYAQUIL, SEPTIEMBRE 2018

Declaración de autoría

“La responsabilidad del contenido de este Trabajo de Titulación, me corresponde exclusivamente; y el patrimonio Intelectual del mismo a la Facultad de Ingeniería Industrial de la Universidad de Guayaquil”

WILA ESPINOZA HENRY ALEXANDER

C.C 0927015628

Dedicatoria

A Dios, por otorgarme la vida, sabiduría y salud para llegar hasta aquí y alcanzar uno de mis objetivos.

A mi madre por haberme dado la vida, creer en mí, ser mi guía y siempre entregarme sus palabras.

A mis hermanos por ser mi motivación cada día.

A mi tío Manuel Vivas por su apoyo incondicional, y ser un ejemplo de superación y lucha.

A mi amor por estar presente en cada momento de mi vida, y estar pendiente con su amor, comprensión y ternura.

Agradecimiento

A Dios, por permitirme alcanzar cada uno de mis objetivos planteados y bendecirme en cada momento de la vida.

A mi Madre, por permitirme llegar a este mundo y junto con sus palabras y consejos hacer de mí una persona de principios y valores.

A mis compañeros, por su apoyo en los momentos difíciles.

A la Master Jeannette Escalante, quien se ha convertido en una persona a la cual admiro y llevaré siempre en mi pensamiento cada una de sus palabras y consejos.

A mis docentes, por compartir sus conocimientos y guiarme en mi formación académica y profesional.

Índice General

N°	Descripción	Pág.
	Introducción	13

Capítulo I

El Problema

N°	Descripción	Pág.
1.1.	Planteamiento del Problema	2
1.1.1	Formulación del Problema	2
1.2.	Justificación	2
1.3.	Delimitaciones	2
1.4.	Hipótesis	3
1.4.1	Hipótesis General	3
1.4.2	Variable Independiente	3
1.4.3	Variable Dependiente	3
1.4.4	Variables Empericas de la Variable Independiente	3
1.4.5	Tipo de Estudio	3
1.5.	Operacionalización	4

Capítulo II

Marco Teórico

N°	Descripción	Pág.
2.1	Antecedentes	6
2.2	Marco Conceptual	8
2.2.1	Software	9
2.2.2	Software Educativo	9
2.2.3	Matlab	10
2.2.4	Sistema de Comunicación	15
2.2.5	Protocolo Tcp/Ip	16
2.3	Modulaciones Digitales	18

N°	Descripción	Pág.
2.3.1	Modulación por Desplazamiento de Amplitud (Ask)	19
2.3.2	Modulación por Desplazamiento de Frecuencia (Fsk)	19
2.3.3	Modulación Digital de Cuadratura en Fase (Qpsk)	20
2.3.4	Modulación Bpsk	21
2.3.5	Modulación Qam (8qam, 16qam, 32qam)	21
2.3.6.	Modulación por Multiplexión por División de Frecuencias Ortogonales (Ofdm)	22
2.4	Marco Legal	22

Capítulo III

Metodología

N°	Descripción	Pág.
3.1	Descripción del Proceso Metodológico	24
3.2	Diseño de la Investigación.	24
3.3	Enfoque de la Investigación.	24
3.3.1	Método Bibliográfico	25
3.3.2.	Método Experimental	25
3.3.3.	Método Deductivo	25
3.4	Estructuración de la Investigación	26
3.4.1	Población y Muestra	26
3.4.2	Selección de la Muestra	26
3.5	Técnicas e Instrumentos	27
3.5.1	Observación	28
3.5.2	Resultado de la Práctica	28
3.5.3	Encuesta	29
3.5.4	Entrevista	29

Capítulo IV

Desarrollo De La Propuesta

N°	Descripción	Pág.
4.1	Elaboración de la aplicación con la herramienta App Designer del Software Matlab	32
4.1.1	Panel Transmisor	34
4.1.2	Panel Transmisor-Sensores	35
4.1.3	Panel Receptor-Configuraciones	37
4.1.4	Panel Receptor- Sensores	38
4.2	Implementación de la aplicación de modulaciones digitales	40
4.2.1	Evaluación 1. Transmisión y Recepcion Ask a través de medio guiado (Cable Par Trenzado Utp Cat5)	40
4.2.2	Evaluación 2. Transmisión Y Recepcion Ask A Través De Medio No Guiado (Red Wi-Fi Con El Estándar Ieee 802.11n)	43
4.3	Resultado De La Práctica	46
4.4	Desarrollo De La Encuesta	47
4.5	Análisis De Encuesta	47
4.6	Análisis De La Hipótesis	49
4.7	Conclusiones	50
4.8	Recomendaciones	51
	Anexos	52
	Bibliografía	125

Índice de Tablas

N°	Descripción	Pág.
1	Operacionalización	4
2	Estudiantes matriculados por campo de conocimientos	9
3	Componentes de Herramienta App Designer	13
4	Componentes App Designer	13
5	Componentes de instrumentación	13
6	Propiedades de App Designer	14
7	Componentes del contenedor	15
8	Componentes de instrumentación	15
9	Técnicas e instrumentos aplicados en la metodología de la investigación	28
10	Resultados obtenidos proceso de transmisión y recepción	29
11	Controles Panel Transmisor	34
12	Controles Panel Transmisor-Sensores	36
13	Controles Panel Transmisor-Sensores	38
14	Controles Panel Receptor-Sensores	39
15	Resultados obtenidos proceso de transmisión y recepción	47
16	Porcentaje de opinión sobre la metodología teórica-práctica.	48
17	Porcentaje de opinión sobre conocimientos del software Matlab y App Designer	48
18	Tabla de verdad	49

Índice de Figuras

N°	Descripción	Pág.
1	Auto reporte IES públicas, autofinanciadas y cofinanciadas	8
2	Vista de Comandos Software Matlab	11
3	Vista de pantalla de la App Designe	12
4	Protocolo TCP/IP. Información Recuperada de IBM.	17
5	Flujo información de todas las capas del protocolo TCP/IP, 2017	17
6	Transmisión y recepción de información en forma simultánea, 2017	18
7	Esquema de modulaciones digitales	18
8	Ejemplo de modulación ASK	19
9	Esquema de modulación FSK	20
10	Modulación QPSK	20
11	Modulación BPSK	21
12	Modulación QAM.	21
13	Esquema de Diseño del Sistema de Modulaciones Digitales	32
14	Diagrama de diseño de la aplicación del sistema de modulaciones digitales.	33
15	Descripción del panel transmisor.	34
16	Descripción del panel transmisor-sensores.	35
17	Descripción del panel Receptor-Configuraciones	37
18	Descripción del panel receptor -sensores.	38
19	Transmisión y recepción ASK utilizando un canal de comunicación alámbrico	40
20.	Implementación del sistema de modulaciones digitales	40
21.	Panel 1: Transmisor/Configuraciones	41
22.	Panel 2: Transmisor-Sensores	42
23	Panel 3: Receptor	42
24	Panel 4: Receptor/sensores	43
25	Transmisión y recepción Digital utilizando un canal de comunicación inalámbrica	43
26	Panel 1: Transmisor/Configuraciones	44
27	Panel 2: Sensores	45
28	Panel 3: Receptor	45
29	Panel 4: Receptor/sensores	46

Índice de Anexos

N°	Descripción	Pág.
1	Manual de usuario	53
2	Preguntas post-encuesta	66
3	Código de la aplicación del sistema de modulaciones digitales	76
4	Formato de la encuesta	122
5	Fotos de la implementación del sistema de modulaciones digitales	124



Universidad de Guayaquil

**FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA DE INGENIERÍA EN TELEINFORMÁTICA**

UNIDAD DE TITULACIÓN

**“IMPLEMENTACIÓN EN SOFTWARE DE LOS SISTEMAS DE
MODULACIONES DIGITALES”**

Autor: Henry Wila Espinoza

Tutor: Ing. Neiser Stalin Ortiz Mosquera, MG

Resumen

La importancia de contar con espacios equipados con tecnología, en las instituciones de educación superior, permite el desarrollo de actividades prácticas, a través de las cuales se logran avances significativos y mejoras en las metodologías de enseñanza y la capacitación de profesionales para brindar soluciones efectivas en el aspecto laboral. El objetivo de este trabajo de investigación es implementar software en sistemas de modulación digital (ASK, FSK, BPSK, QPSK, QAM), en el dominio del tiempo, utilizando la herramienta de interfaz gráfica App Designer integrada en el software MATLAB, lo que demuestra la importancia de tener Laboratorios equipados, y un papel fundamental en la formación académica, que permita reforzar los conocimientos obtenidos en la clase, con lo que se facilita el proceso de aprendizaje. Este trabajo de titulación se encuentra en las metodologías descriptivas, bibliográficas y experimentales, con las cuales recopila información que permite realizar el sistema propuesto previamente y mediante la preparación de encuesta permite recabar las opiniones de los alumnos del 8° semestre 2018-2019. de la carrera de Ingeniería en Teleinformática. A través de esta investigación se determinó que a través del software se puede desarrollar un sistema de modulaciones digitales, evitando así el uso de hardware como las tarjetas de adquisición de datos, lo que determinó que la metodología teórico-práctica debería ser más utilizada en la enseñanza y el aprendizaje. Proceso dentro de la carrera de Ingeniería en Teleinformática.

Palabras Claves: Modulaciones Digitales, Software, Sistema, Adquisición.



FACULTY OF INDUSTRIAL ENGINEERING
CAREER OF ENGINEERING IN TELEINFORMATICS

UNIT OF TITULATION

**"IMPLEMENTATION THROUGH A SOFTWARE OF THE SYSTEMS OF
DIGITAL MODULATIONS"**

Author: Henry Wila Espinoza

Advisor: TE Neiser Stalín Ortiz Mosquera, MG

Abstract

The importance of having spaces equipped with technology, in higher education institutions, allows the development of practical activities, through which significant progress and improvement is achieved in teaching methodologies and the training of professionals to provide effective solutions in the labor aspect. The objective of this research work is to implement software in digital modulation systems (ASK, FSK, BPSK, QPSK, QAM), in the time domain, using the graphic interface tool App Designer integrated into the MATLAB Software, which demonstrates the importance of having equipped laboratories, and a fundamental role in academic training, allowing to reinforce the knowledge obtained in the class, with which the learning process is facilitated. This work of titulation is he in the descriptive, bibliographic and experimental methodologies, with which compiles information that allows develop the system previously proposed and by means of the preparation of survey allows collecting the opinions of the students of 8th semester 2018-2019 CII of the career engineering in Teleinformatics. Through this research it was determined that through the software a system of digital modulations can be developed, avoiding in this way the use of hardware such as data acquisition cards, this determined that the theoretical-practical methodology should be more used in the teaching and learning process within the career Engineering in Teleinformatics.

Keywords: Digital Modulations, Software, System, Acquistion

Introducción

El continuo desarrollo de las tecnologías que se produce día a día abre paso a que diferentes técnicas se centren en la creación de nuevas metodologías de enseñanza, con lo cual se busca fortalecer el proceso de enseñanza y aprendizaje, en el aspecto teórico, como a su vez en el aspecto práctico lo que permite alcanzar un mejor entendimiento de los estudiantes.

Mediante el uso de nuevas herramientas tecnológicas, en las aulas de clase, y el complemento de la enseñanza mediante los laboratorios que poseen las carreras de ingeniería en las diferentes universidades, se ha logrado que los estudiantes de las diferentes carreras puedan demostrar nuevas capacidades, lo que es de principal apoyo para su campo profesional, y de esta manera poder asumir nuevos retos en el aspecto laboral. Mediante la utilización de equipos, los estudiantes pueden simular problemas que se presentan en su desempeño laboral, esto permite que los estudiantes muestren un mayor interés sobre el desarrollo de las actividades prácticas que el docente propone en la enseñanza de la cátedra.

En el Art. 343 de la Constitución de la República del Ecuador (2008), se establece: “El sistema nacional de educación tendrá como finalidad el desarrollo de capacidades y potencialidades individuales y colectivas de la población, que posibiliten el aprendizaje, y la generación y utilización de conocimientos, técnicas, saberes, artes y cultura” lo que permite que se garantice el derecho que poseen los estudiantes de contar con una educación de calidad.

Asumir una reestructuración y repotenciación, es de vital importancia, con lo cual la Universidad de Guayaquil busca llegar a ser una institución de educación superior ubicada en la máxima categoría de excelencia, en todo el Ecuador.

El presente trabajo de investigación busca demostrar la importancia que asume realizar prácticas de laboratorio en la Facultad de Ingeniería Industrial, haciendo uso de equipos y software, con lo cual se optimiza el tiempo, que conlleva el desarrollo de las prácticas en las diferentes cátedras, como las materias relacionadas al estudio de las comunicaciones, permitiendo a su vez formar el perfil profesional de los próximos ingenieros de la carrera de Ingeniería en Teleinformática.

Capítulo I

El Problema

1.1. Planteamiento del Problema

El continuo desarrollo de las tecnologías que se producen día a día establece un universo en el cual se conjugan el conocimiento técnico y la gran importancia del aprendizaje teórico de los estudiantes de ingeniería.

Para lo cual se necesita un laboratorio equipado de herramientas tecnológicas y equipos de campo que facilitan el aprendizaje, los cuales son osciloscopios, generadores de señales, instrumentos de análisis de espectros, para los cuales se depende de una infraestructura para los equipos. El estudiante presenta la necesidad de conocer sobre la manipulación de las diferentes herramientas.

Es decir, se toma en cuenta que el tiempo necesario para realizar las prácticas en los laboratorios es demasiado, debido a que no se cuenta con las herramientas tecnológicas adecuadas, en la mayoría de las veces estas prácticas no se concluyen con el éxito esperado, lo cual imposibilita el desarrollo técnico de los estudiantes de ingeniería.

Es por esto que cabe mencionar que el docente no tiene la facilidad de desarrollar una clase teórica-práctica, debido a la falta de elementos de análisis electrónicos, el espacio destinado en donde se imparte la asignatura y a su vez cabe mencionar el escaso conocimiento de los estudiantes, todo lo mencionado genera en conjunto un retraso en el cumplimiento del cronograma establecido para la cátedra y lo cual genera escasos conocimientos para los estudiantes. La falta de equipos genera que el docente no realice un estudio global que exija una mayor dificultad, lo cual genera que el docente limite su enseñanza a conceptos básicos y esto conlleva a realizar clases más teóricas que en un ambiente práctico.

1.1.1 Formulación del problema

El problema radica en la falta de recursos tecnológicos para realizar las prácticas que son necesarias, en la carrera de ingeniería en Teleinformática de la Universidad de Guayaquil, al no contar con los suficientes equipos en el laboratorio de networking, el proceso de enseñanza-aprendizaje se limita debido a que los estudiantes necesitan aplicar los conocimientos adquiridos en el aula de clase.

1.2. Justificación

El laboratorio es de principal apoyo para los estudiantes puesto que permite que los estudiantes se introduzcan a fondo en la realidad del mundo profesional, mediante la práctica el estudiante experimenta casos en los cuales mediante el análisis puede originar soluciones a los problemas planteados.

El proceso de implementación en software de un sistema de modulaciones digitales permite a los estudiantes, puedan enfocarse directamente en el análisis y comportamiento de las señales, y a su vez adquiera nuevos conocimientos con la puesta en práctica y dicho elemento se convierta en el material de refuerzo del aspecto teórico. Realizando una comparación de los cálculos matemáticos y los resultados que obtiene.

A través del presente proyecto se pretende lograr que los estudiantes, complementen el conocimiento teórico, presente en las materias de comunicación.

El tiempo de implementación se reduce puesto que el estudiante, no tiene necesidad de manipular o adquirir elementos físicos, evitando el riesgo de que estos puedan dañarse, al contrario, la optimización de herramientas y el tiempo lo cual permitirá que el tiempo sobrante sirva para consultar dudas entre estudiantes y docentes sobre el procedimiento a realizar

La importancia de este proyecto es poder otorgar a los estudiantes un sistema en el cual se puedan realizar prácticas de modulaciones digitales, permitiendo el análisis de los resultados obtenidos luego de la modulación digital sin ningún inconveniente técnico, lo cual permitirá el desarrollo del estudiante en el mundo educativo y profesional.

Como factor principal del presente trabajo de investigación a través del Desarrollo e Innovación del Fondo Competitivo de Investigación – FCI –2018, se tiene como tiene como tópico el **“Desarrollo de un espacio tecnológico en el área de comunicaciones mediante sistemas de procesamiento de datos que permitan la generación de sistemas de comunicaciones”**.

1.3. Delimitaciones

La presente investigación se realizará en el laboratorio de networking en donde se instalará un archivo ejecutable y se podrá realizar modulaciones digitales y contribuir con el perfeccionamiento de los conocimientos teóricos en la asignatura de comunicaciones. La

implementación de este proyecto es factible, puesto que el nivel de conocimiento en lo que respecta al manejo de la herramienta App Designer del software MATLAB es alto debido a que es de fácil manejo y permite comprender los fundamentos teóricos de esta asignatura.

Se evaluarán los diferentes tipos de modulaciones digitales que son estudiadas en el octavo semestre de la carrera de teleinformática

Se diseñará en software los diferentes sistemas de modulaciones digitales que los estudiantes requieren en su proceso de aprendizaje.

Mediante la utilización de este software se tendrá mayor accesibilidad a prácticas estructuradas y perfeccionamiento de los estudiantes en la asignatura de comunicaciones.

1.4. Hipótesis

1.4.1 Hipótesis General

La unificación de un sistema de modulaciones digitales complementará el conocimiento teórico, mediante la utilización de software, lo cual permitirá impulsar el desarrollo académico de los estudiantes de la carrera de Ingeniería en Teleinformática.

1.4.2 Variable Independiente

La implementación del software con las diferentes modulaciones digitales requeridas por los estudiantes.

1.4.3 Variable Dependiente

El completo acierto en los procesos, que permitan ejecutar la implementación en software de los sistemas de modulaciones digitales concurrentes en la práctica correspondiente a la carrera de Ingeniería en Teleinformática.

1.4.4 Variables Empericas de la Variable Independiente

- Nivel de conocimiento de sistemas de modulaciones requerido.
- Disminuciones en el grado de dificultad de los conocimientos teóricos adquiridos.
- Menor tiempo de los estudiantes en el cumplimiento de prácticas.
- Menos recursos materiales necesarios para cumplir con una práctica de laboratorio en lo que respecta a sistemas de modulaciones digitales.

1.4.5 Tipo de estudio

La metodología para desarrollar dentro del presente proyecto de implementación e

investigación la cual supone un desarrollo de compromiso tecnológico se centra en una metodología bibliográfica y experimental.

Bibliográfico

Recopilar investigaciones, revistas de alto impacto, documentación, y demás fuentes que supongan información fidedigna, veraz y muestren una relación con el tema.

Exploratorio

Reunir fuentes de información que contribuyan al presente trabajo de titulación y su correcta ejecución haciendo mención a las variables que se presentan en el mismo.

Descriptivo

Determinar la metodología a utilizar, lo cual posibilita la realización de los objetivos propuestos en el presente trabajo de titulación.

Investigativo

Las diferentes actividades de indagación, que sirvan de soporte para el análisis y organización, lo cual permita decidir los requisitos indispensables para el presente trabajo.

1.5. Operacionalización

Tabla 1. Operacionalización

Objetivo General: Implementar en software un sistema de modulaciones digitales ASK, FSK, BPSK, QPSK, QAM.

Objetivos Específicos	Variables.	Dimensión.	Indicador.
1. Estudiar el comportamiento de un sistema de modulación digital.	Sistemas de Modulación Digitales.	<ul style="list-style-type: none"> • Bibliografía • Búsqueda e internet. 	<ul style="list-style-type: none"> • Número de artículos similares. • Número de citas bibliográficas
2. Determinar los elementos para el desarrollo de los sistemas de modulaciones digitales	Sistemas de Modulaciones Digitales	<ul style="list-style-type: none"> • App Designer de MATLAB 	<ul style="list-style-type: none"> • Tiempo utilizado para la implementación del sistema
3. Analizar las señales en un sistema de modulación digital.	Sistemas de Modulaciones Digitales	<ul style="list-style-type: none"> • App Designer de Matlab 	de modulaciones digitales.
4. Desarrollar los sistemas de modulaciones digitales en App Designer de Matlab.	App Designer	<ul style="list-style-type: none"> • App Designer de 	• Nivel de
5. Simular los tipos de			

modulaciones digitales en App Designer Matlab.	App Designer	Matlab	dificultad
6. Implementar un sistema de visualización de la señal modulada digitalmente en App Designer.	<ul style="list-style-type: none"> • App Designer 	<ul style="list-style-type: none"> • App Designer de Matlab 	<ul style="list-style-type: none"> • Tiempo utilizado para la implementación del sistema de modulaciones digitales • Tiempo utilizado para las modulaciones digitales.

Operacionalización de Variables, 2019. Información tomada directamente del autor. Elaborado por el autor.

Capítulo II

Marco Teórico

2.1 Antecedentes

A lo largo del tiempo la tecnología ha variado constantemente en las últimas décadas, no solo a nivel nacional, sino que el impacto global de los sistemas de comunicación proporciona herramientas muy útiles para seguir creciendo en esta rama, la informática, debido a que el conocimiento y entendimiento de sistemas de comunicación digital es fundamental en la formación de profesionales en el área de telecomunicación es importante conocer que esto se debe balancear el conocimiento de conceptos teóricos y su implementación práctica.

En este proyecto de implementación con el fin de obtener unos resultados eficaces se han realizado investigaciones precisas y se ha respaldado de recomendaciones de expertos, de proyectos y estudios a lo largo de este tiempo de elaboración.

Tal como propone:

(Santiana 2014) en su recopilación llamada Modulación Electrónica; Desarrollo de un Modem; Modulación de Datos en el cual su planteamiento principal de trabajo fue crear y desenvolver el proyecto de un modem utilizando la técnica de modulación D-QPSK sobre la plataforma de entrenamiento FPGA Stratix IV de Altera.

Según (GARCÍA, 2016) manifestó que:

Haciendo uso de la tecnología el estudiante aprende, de esta manera mejora los diferentes procesos que asume el aprendizaje, rendimiento y su organización curricular, se menciona también “el aspecto práctico eleva el interés, lo cual crea mayor curiosidad en su conocimiento lo cual supone investigación, evaluación y comparta los conocimientos que este adquiera”

El estudio establece e implementa una guía de prácticas para sistemas de comunicación digital con los instrumentos del laboratorio, cada una de estas, contiene los pasos a seguir para completar cada objetivo propuesto y obtener una solución exitosa.

Además, los temas introducidos en este documento son los más importantes en el procedimiento de conversión de señales analógica-digital y su modulación, iniciando desde un muestreo de una señal analógica, la digitalización, codificación hasta algunas modulaciones de señales binarias como ASK, FSK y PSK.

Un tercer trabajo como el de (Molina., 2016) aporta los procesos realizados en las prácticas, los cuales son desarrollados con equipos electrónicos TIMS (Telecommunication

Instructional Modelling) y comprobados a nivel teórico como práctico, visualizando cada uno de los resultados con instrumentos apropiados de medición, entre los que se incluyen osciloscopios y multímetros.

La evolución ulterior de las técnicas de modulación y detección aumentará la eficiencia espectral en los sistemas multiplexados por división de longitud de onda (WDM), y mejorará la robustez frente a degradaciones de la transmisión, y facilitar la compensación eléctrica de tales impedimentos.

Como se explica en este documento, la modulación no binaria con detección coherente produce el mejor rendimiento en todos los aspectos anteriores, pero a costa de una mayor complejidad de la implementación.

Se menciona que las diferentes unidades de educación, como universidades son las que se encargan de incluir componentes, que permitan a los estudiantes alcanzar un nivel, que les permita obtener una estabilidad y logren alcanzar los diferentes objetivos propuestos.

En una investigación del (SNNA, 2015) se mostró que:

Las carreras de ingeniería muestran un alto índice de aceptación en las diferentes universidades públicas, principalmente se hacen mención las que engloban áreas técnicas, debido que mediante esta se da solución a problemas cotidianos, mediante lo cual es necesario que se combinen conocimientos de carácter teórico-practico, otorgándole habilidades y destrezas para la manipulación de equipos.

En el presente trabajo investigativo se tiene como referencia diferentes trabajos obtenidos de varios autores; entre ellos la publicación del autor Torres (2016) “Los laboratorios como ambientes de aprendizaje, se muestran como escenarios pedagógicos en donde se desarrollan actividades que permiten fortalecer la gestión del conocimiento por parte del estudiante”, además agrega “Estos ambientes de aprendizaje están formados por un conjunto de elementos, materiales y actores (docentes y estudiantes) que participan en el proceso de aprendizaje, basado en objetivos y propósitos claramente definidos desde el punto de vista cognitivo, procedimental y valores”.

En el ámbito de la educación se proyecta que es necesario construir relaciones entre la teoría y la práctica, de esta manera hacer converger el conocimiento didáctico y la acción docente, pero apenas se ha investigado sistemáticamente sobre ello. En esta tesis se llevan a cabo estas relaciones mediante un estudio de caso único en Educación Primaria, siguiendo una metodología etnográfica, para analizar las implicaciones de estas en la enseñanza y el desarrollo profesional docente.

Según (BERRONES FRANCO, 2013), a través el tema “Análisis y estudio para el

mejoramiento del laboratorio de telecomunicaciones”, menciona que el proceso de enseñanza práctico y el estudio para la optimización del laboratorio, para las diferentes materias que se muestran en la curricular de la carrera.

El Consejo de educación Superior (2017) menciona que en el Ecuador se cuenta con un total de 55 Universidades y Escuelas Politécnicas entre públicas, autofinancias y cofinanciadas con un total de 587.799 de estudiantes matriculados y entre las Instituciones de Educación Superior de mayor demanda en el 2017, como se muestra en la figura N° 1.

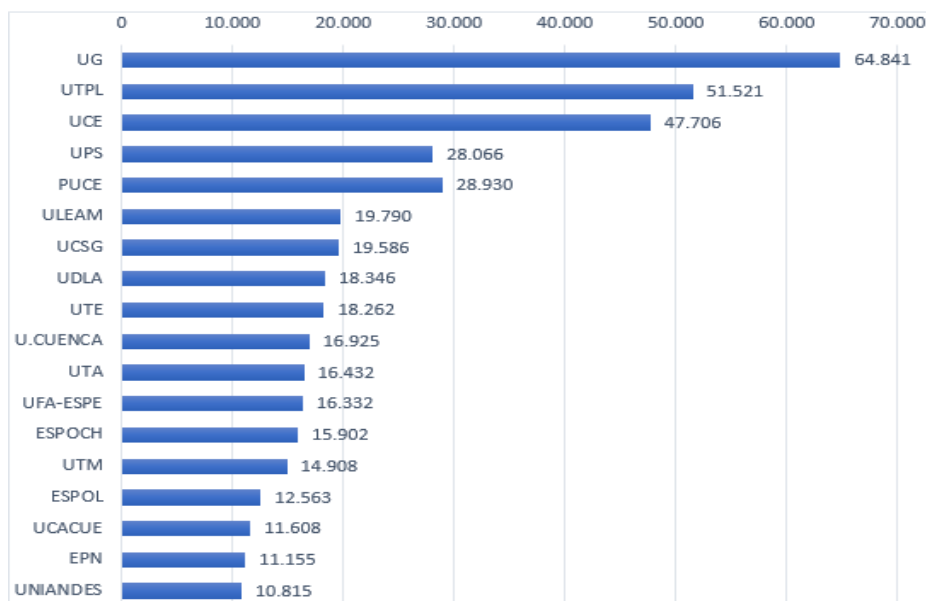


Figura 1. Auto reporte IES públicas, autofinanciadas y cofinanciadas, 2017. Información adaptada del SENESCYT y, SNIIESE. Elaborada por el autor.

2.2 Marco Conceptual

Según (GARCÍA, 2016) menciona : “El estudiante aprende usando la tecnología y mejora los procesos de aprendizaje, evaluación y organización curricular”, además agrega “la práctica despierta el interés en el estudiante, potenciando la curiosidad por el conocimiento, de tal forma que investigue, pruebe, seleccione, evalúe y comparta con otros el conocimiento”, “la excelencia académica debe otorgar a los estudiantes y docentes las herramientas pedagógicas útiles para la conexión entre los conocimientos teóricos adquiridos y la práctica experimental validada”.

Según el SNIIESE en el campo de conocimiento donde se encuentra la rama de ingeniería en el 2017 existe un total de 39.462 graduados lo cual ocupa el cuarto lugar con respecto a los otros campos de conocimiento que se encuentran dentro de las Universidades y Escuelas Politécnicas del Ecuador (Consejo de Educación Superior, 2017), como se puede observar en la tabla N° 2.

Tabla 2. *Estudiantes matriculados por campo de conocimientos*

Graduados por campo de conocimiento	
Administración de Empresas y Derecho	153619
Agricultura, Silvicultura, Pesca y Veterinaria	14461
Artes y Humanidades	12214
Ciencias Naturales, Matemáticas y Estadística	9914
Ciencias Sociales, Periodismo e Información	35849
Educación	94677
Ingeniería, Industria y Construcción	39462
Salud y Bienestar	48637
Servicios	9284
Tecnologías de la Información y Comunicación	17259

Información adaptada del SNIIESE 2017. Elaborado por el autor.

Según Gómez (2013) señala “La importancia de los laboratorios en la enseñanza de la ingeniería” indica que las prácticas de laboratorio ayudan a que los egresados cuenten con algún tipo de experiencia que los prepare para su formación en el campo laboral, ya que los empleadores valoran la experiencia, cualquier iniciativa en la actualización de nuevos programas de enseñanza en las ingenierías debe continuar reforzando la necesidad de realizar trabajos presenciales dentro de los laboratorios”.

2.2.1 Software

Según (Sommerville, 2015) “El software no solo son programas sino, todos los documentos asociados a la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta”.

2.2.2 Software Educativo

Se debe tener en cuenta que el software en una institución cuenta con diferentes funciones como son administrar los recursos de la información y suministrar herramientas de habilitación de los estudiantes (Aranda, 2014)

El concepto genérico de software educacional o educativo se entiende como cualquier programa computacional cuyas características estructurales y funcionales ofrezcan ayuda al proceso de enseñar, aprender y administrar. Un concepto más específico de Software educacional o educativo lo considera el material de aprendizaje diseñado para ser utilizado con un computador en los procesos de enseñanza y aprendizaje

2.2.3 MATLAB

En un ambiente fácil de utilizar, en el cual los problemas y las soluciones son indicados como se escriben matemáticamente, sin la programación habitual. El nombre MATLAB proviene de MATRIX LABORATORY (Laboratorio de Matrices). MATLAB fue escrito desde sus inicios para suministrar un acceso de manera sencilla al software matricial desarrollado por los proyectos LINPACK y EISPACK, los cuales constituyen lo más avanzado en programas de cálculo matricial.

MATLAB es un sistema interactivo en donde el elemento básico de datos es una matriz que no necesita medición. Esto permite solucionar muchos problemas numéricos en una fracción del tiempo que llevaría hacerlo en lenguajes como C, BASIC o FORTRAN. MATLAB ha ido evolucionando en los últimos años gracias a la colaboración de muchos usuarios.

En los entornos universitarios se ha convertido en una herramienta de enseñanza estándar para cursos de introducción en álgebra lineal aplicada, así también como cursos avanzados en otras áreas.

En la industria, MATLAB se utiliza para investigación y para solucionar problemas prácticos de ingeniería y matemáticas, con un gran hincapié en aplicaciones de control y procesamiento de señales. MATLAB así también aporta una serie de soluciones específicas denominadas TOOLBOXES. Estas son muy importantes para la mayoría de los usuarios de MATLAB y son un grupo de funciones que amplían el entorno para de esta manera 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.

Posiblemente una de las características más importante de MATLAB es su capacidad de crecimiento. Lo cual permite al usuario convertirse en un autor contribuyente, creando sus propias aplicaciones.

En resumen, las prestaciones más importantes de MATLAB son:

- Escritura del programa en lenguaje matemático.
- Implementación de las matrices como elemento básico del lenguaje, lo que permite una gran disminución del código, ya que no necesita implementar el cálculo matricial

- Implementación de aritmética compleja.
- Un amplio contenido de órdenes específicas, agrupadas en TOOLBOXES.
- Posibilidad de ampliar y adaptar el lenguaje, mediante ficheros de script y funciones como se visualiza en la figura N° 2.

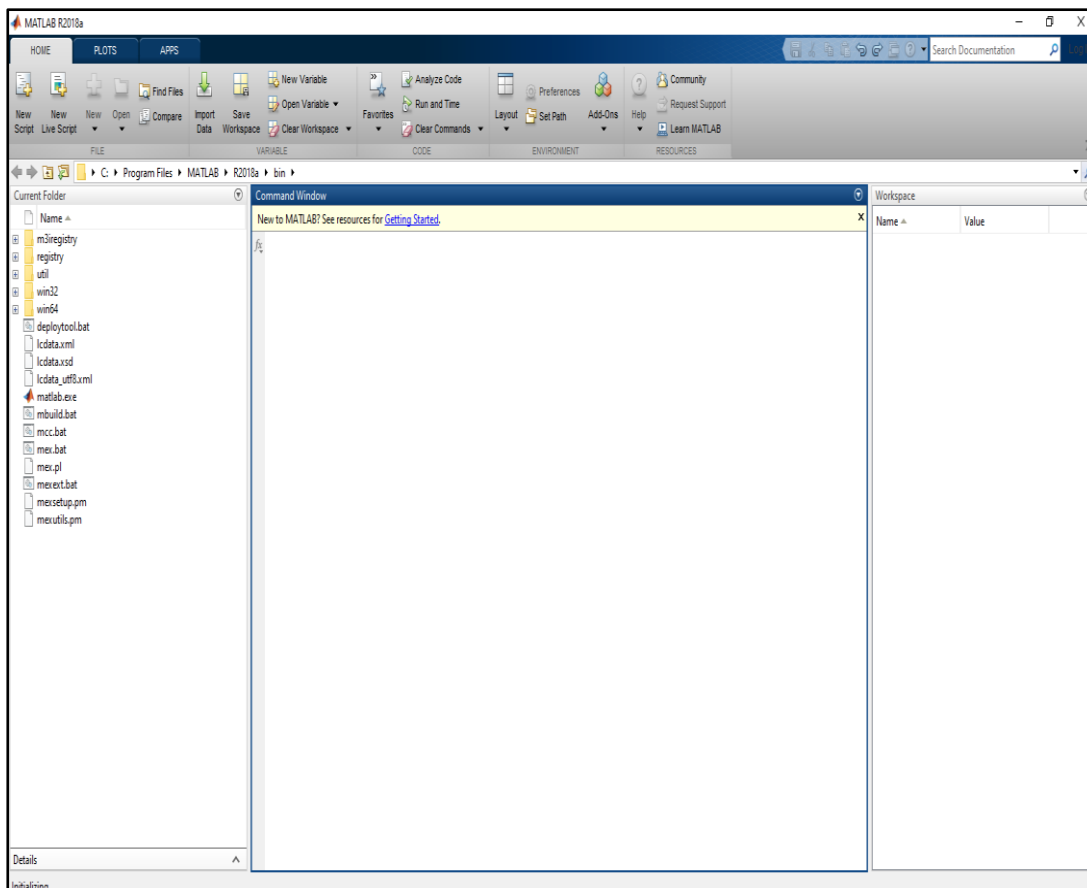


Figura 2. Vista de Comandos Software Matlab. Información adaptada directamente del autor. Elaborado por el autor.

Matlab ha pasado de ser algo creado simplemente para dar apoyo en cursos relacionados con teoría de matrices a transformarse en una poderosa herramienta tanto en el ambiente educativo como en el industrial. A nivel pedagógico se ha transformado en la primordial herramienta de los cursos relacionados con el álgebra matricial, tanto a nivel básico como a nivel superior. A nivel industrial, tienen un gran número de aplicaciones en muchos problemas prácticos de ingeniería y matemáticas.

Características Software Matlab

- Cálculos intensivos desde un punto de vista numérico.
- Gráficos y visualización avanzada.
- Lenguaje de alto nivel basado en vectores, arrays y matrices.

- Colección muy útil de funciones de aplicación.

Las eficaces capacidades de cálculo técnico que tiene MATLAB se ponen al alcance de los estudiantes, aunque limita el tamaño de las matrices a diferentes elementos, la edición para estudiante conserva toda la potencia de la versión profesional de MATLAB 4.0, es una plataforma diseñada para que los estudiantes puedan ejecutarlo en sus propios computadores personales bajo Windows.

2.2.3.1 App Designer

App Designer es una herramienta de software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio (Math Works,2018), como se visualiza en la figura N° 3.

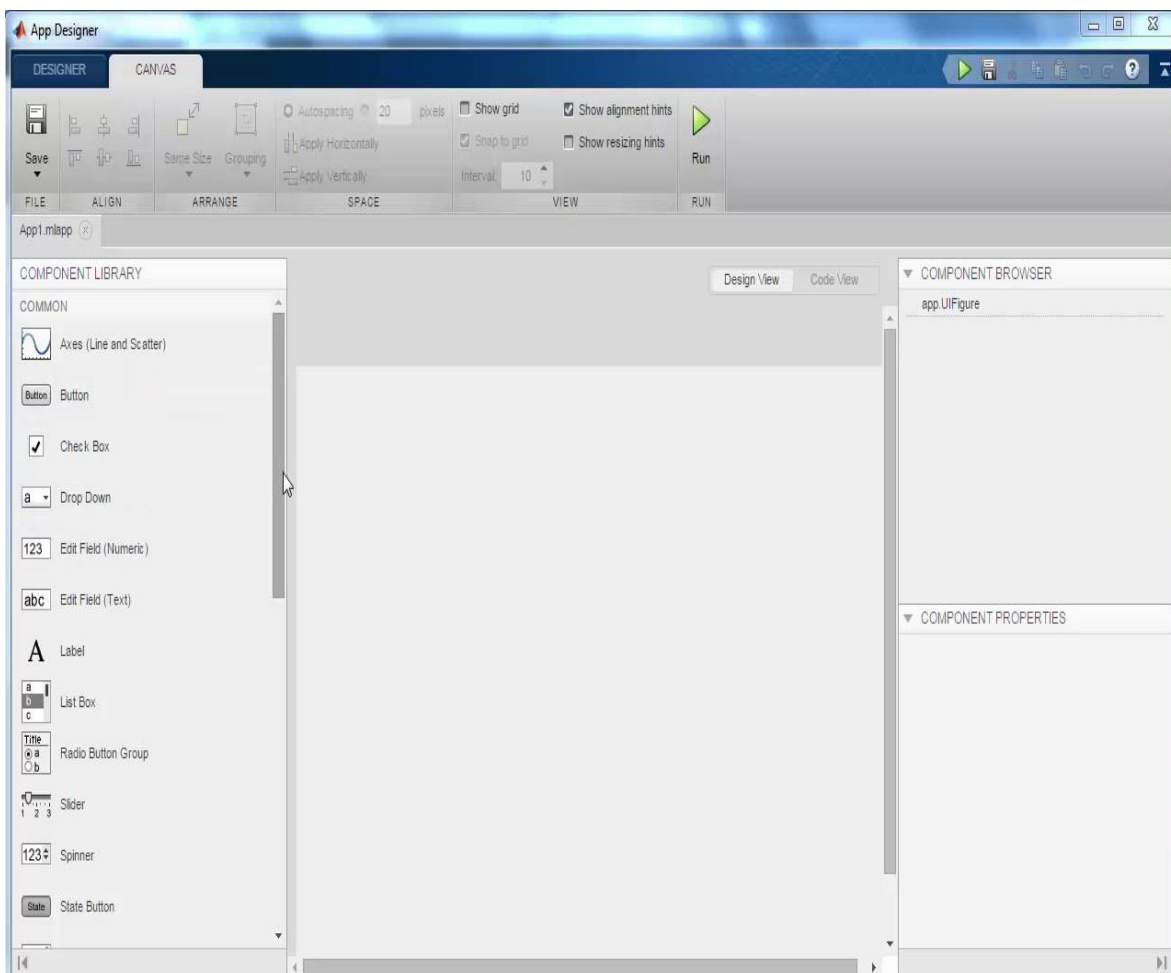


Figura 3. Vista de pantalla de la App Designer. Información tomada y adaptada directamente del autor. Elaborado por el autor.

Dentro del entorno de App Designer se cuenta con diferentes componentes, los cuales se asignan en el desarrollo de una aplicación, los detalles de cada uno de los componentes se visualizan en la tabla N°3.

2.2.3.2 Componentes de App Designer

Tabla 3. Componentes de Herramienta App Designer

Componente	Función
Uiaxes	Inserta ejes de IU para gráficos en App Designer
Uibutton	Incluye un botón pulsador o componente de botón de estado
Uibuttongroup	Inserta un grupo de botones para administrar los botones de radio y los botones de alternar
Uidropdown	Inserta componente desplegable
Uieditfield	Inserta un componente de campo de edición de texto o numérico
Uilabel	Inserta componente de etiqueta
Uislider	Inserta componente de control deslizante
Uispinner	Inserta componente spinner
Uitextarea	Inserta componente de área de texto
Uitogglebutton	Inserta componente de botón de alternancia

Tabla 4. Componentes App Designer

Componente	Función
Uifigure	Inserta una figura para diseñar aplicaciones
Uipanel	Inserta un objeto contenedor de panel
Uitabgroup	Inserta contenedor para paneles con pestañas
Uitab	Inserta un panel con pestañas

Información tomada adaptada de Mathworks, 2018. Elaborado por el autor.

Tabla 5. Componentes de instrumentación

Componente	Función
Uigauge	Inserta un componente gauge
Uiknob	Inserta un componente de mando
Uilamp	Inserta un componente de lámpara
Uiswitch	Inserta un interruptor deslizante, un interruptor basculante o un componente de interruptor de palanca

Información adaptada de Mathworks, 2018. Elaborado por el autor.

Tabla 6. *Propiedades de App Designer*

Componente	Función
UIAxes Properties	(App Designer) Aspecto y comportamiento de los ejes de la interfaz de usuario
Button Properties	(App Designer) Controlar la apariencia y el comportamiento del botón
ButtonGroup Properties	(App Designer) Controlan la apariencia y el comportamiento del grupo de botones
DropDown Properties	(App Designer) Control de apariencia de lista desplegable y comportamiento
EditField Properties	(App Designer) Controla la apariencia y el comportamiento del campo de edición
Label Properties	(App Designer) Crear componente de etiqueta
NumericEditField Properties	(App Designer) Controla la apariencia y el comportamiento del campo de edición numérica,
ProgressDialog Properties	(App Designer) Control de apariencia y comportamiento del cuadro de diálogo de progreso
RadioButton Properties	(App Designer) Controla la apariencia del botón de radio
Slider Properties	(App Designer) Control del aspecto y el comportamiento del control deslizante
Spinner Properties	(App Designer) Controla la apariencia y el comportamiento de los hilanderos
StateButton Properties	(App Designer) Controlan la apariencia y el comportamiento del botón de estado
TextArea Properties	(App Designer) Controlan la apariencia y el comportamiento del área de texto
ToggleButton Properties	(App Designer) Control del aspecto del botón de alternar

Información adaptada de Mathworks, 2018. Elaborado por el autor.

Tabla 7. *Componentes del contenedor*

Componente	Función
UI Figure Properties	(App Designer) Controlan la apariencia y el comportamiento de la figura de la interfaz de usuario
Panel Properties	(App Designer) Apariencia del panel de control
TabGroup Properties	(App Designer) Apariencia y comportamiento del grupo de pestañas de control
Tab Properties	(App Designer) Apariencia de la pestaña de control

Información adaptada de Mathworks, 2018. Elaborado por el autor.

Tabla 8. *Componentes de instrumentación*

Componente	Función
Discrete Knob Properties	(App Designer) controla la apariencia y el comportamiento de los mandos discretos
Gauge Properties	(App Designer) Controlar la apariencia y el comportamiento del medidor
Knob Properties	(App Designer) Control del aspecto y comportamiento del mando
Lamp Properties	(App Designer) Apariencia de la lámpara de control
LinearGauge Properties	(App Designer) Controlan la apariencia y el comportamiento de los medidores lineales
SemicircularGauge Properties	(App Designer) Control del aspecto del medidor semicircular
Switch Properties	(App Designer) Control de apariencia y comportamiento del switch
ToggleSwitch Properties	(App Designer) Control de apariencia y comportamiento del conmutador

Información adaptada de Mathworks, 2018. Elaborado por el autor

2.2.4 Sistema de Comunicación

El sistema de comunicación es el encargado de transmitir señales que contienen información de un lugar a otro mediante un canal de comunicación que separa al transmisor

del receptor.

Se puede utilizar dos medios los cuales pueden ser:

Medio Guiado o alámbrico: Guían las ondas a través de un camino físico, los cuales pueden ser cable coaxial, la fibra óptica y el par trenzado.

Medio No guiado o inalámbrico: Los medios no guiados proporcionan un soporte para que las ondas se transmitan, es decir las envían a través del aire.

Existen tres elementos básicos en el sistema de comunicación los cuales son: transmisor, medio o canal de transmisión y el receptor. Cada uno de estos elementos cumple una función característica dentro del sistema.

Transmisor: Mediante el transmisor se producen los datos los cuales permiten codificar un mensaje y lo lleva a través del canal en forma de onda.

Medio o canal de transmisión: Establece un enlace que permita comunicar el emisor con el receptor, llevando la señal portadora que es donde viaja la información mediante la comunicación.

Receptor: Permite recuperar la señal que llega desde el sistema transmisor.

2.2.5 Protocolo TCP/IP

El protocolo TCP/IP es un conjunto de normas para formatos de mensaje y procedimientos por medio de las cuales los programas de aplicación y las máquinas intercambian información, este protocolo puede interpretarse en términos de niveles o capas, se encuentra diseñado para ser un componente de la red y tiene sus tareas asignadas. (IBM,2017).

El formato de capas del protocolo TCP/IP según IBM (2017) es:

Estos protocolos TCP/IP reciben los datos de la aplicación, y estos datos los dividen en partes más pequeñas llamadas paquetes, agregan una dirección de destino y, a continuación, pasan los paquetes a la siguiente capa de protocolo, la capa de red de Internet. La capa de red de internet pone el paquete en un datagrama de **IP (Internet Protocol)**, pone la cabecera y la cola de datagrama, decide dónde enviar el datagrama (directamente a un destino o a una pasarela) y pasa el datagrama a la capa de interfaz de red. La capa de interfaz de red acepta los datagramas **IP** y los transmite como tramas a través de un hardware de red específico,

por ejemplo, redes Ethernet o de Red en anillo (IBM, 2017), como se visualiza en la figura N°4.

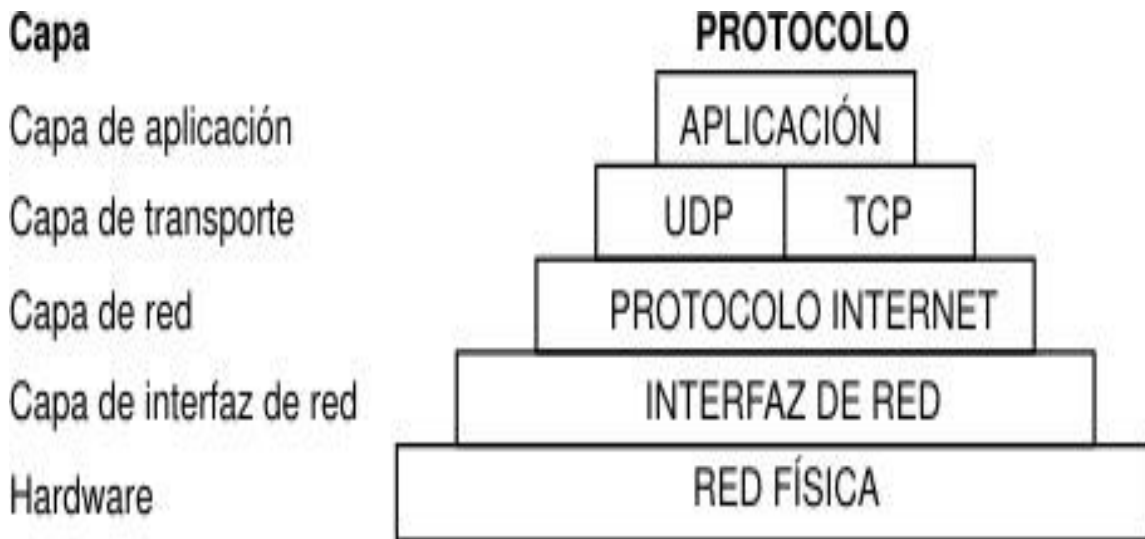


Figura 4. Protocolo TCP/IP. Información Recuperada de IBM. Elaborada por el autor.

En la figura N° 5 se visualiza el flujo información entre las diferentes capas del protocolo TCP/IP desde remitente hacia el sistema destino principal.

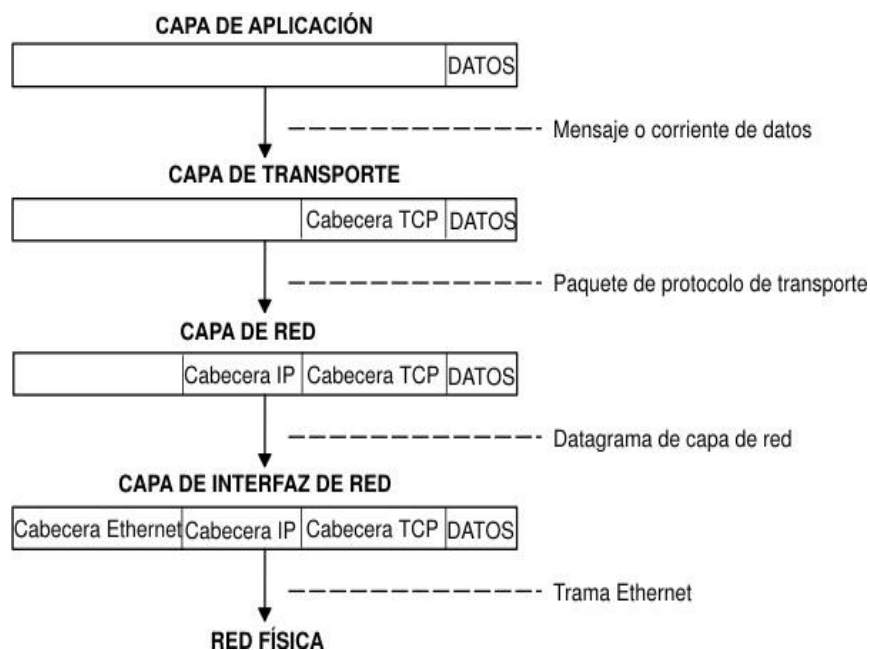


Figura 5. Flujo información de todas las capas del protocolo TCP/IP, 2017. Información tomada de IBM. Elaborado por el autor.

En la figura N° 6 se visualiza el proceso de transmisión y recepción simultaneo entre capas.

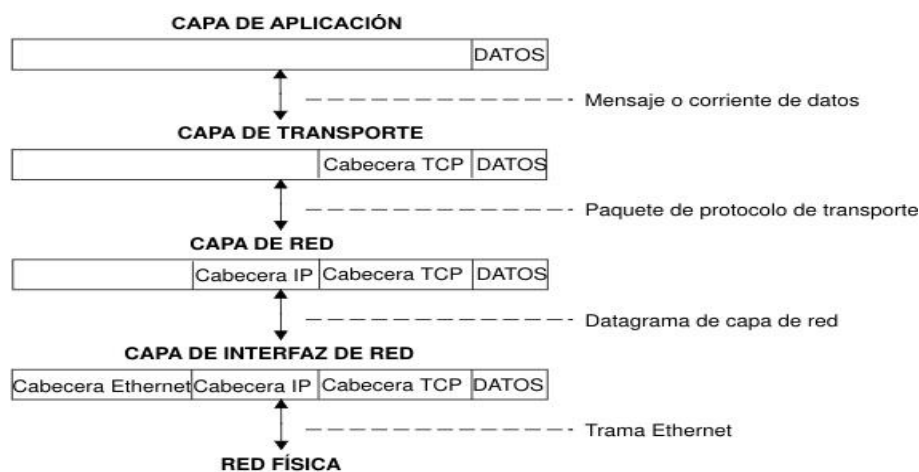


Figura 6. Transmisión y recepción de información en forma simultánea, 2017. Información tomada de IBM. Elaborada por el autor.

2.3 Modulaciones Digitales

El proceso de modulación digital engloba el conjunto de procesar diferentes grupos de bits, presentes en una señal portadora, con una señal de información que se representa a través de grupos de bits (1 y 0), a su vez llevan los datos a lo largo de un canal en la cual se destacan las modulaciones ASK, BPSK, FSK, QPSK, QAM como se muestra en la figura N°7.

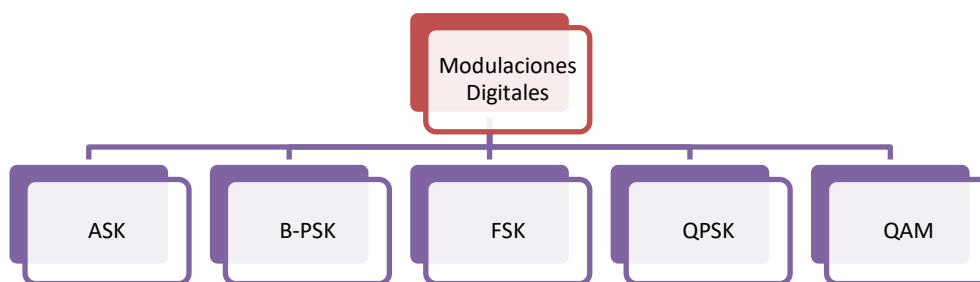


Figura 7. Esquema de modulaciones digitales, Información tomada directamente del autor. Elaborado por el autor

2.3.1 Modulación por Desplazamiento de Amplitud (ASK).

La modulación ASK, es un proceso en el cual se modifica la amplitud de la señal moduladora digital, y a su vez se muestran los valores binarios con dos amplitudes distintas, que son tomadas de referencia como el dígito binario que se representan haciendo uso de una señal portadora con una amplitud fija, y en la otra se centra la presencia de la señal portadora.

La señal ASK, cumple la función de modificar o alterar la amplitud de la señal portadora, mediante los pulsos que se muestran en la codificación, como se visualiza en la figura N° 8.

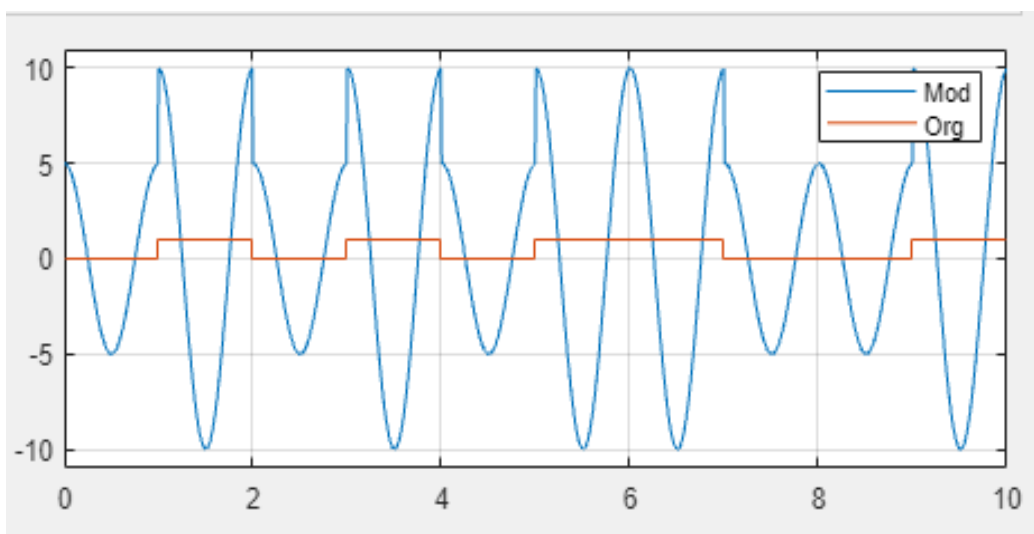


Figura 8. Ejemplo de modulación ASK. Información tomada directamente del autor. Elaborado por el autor.

Expresión matemática de la modulación:

$$V_{ASK}(t) = [1 + V_m(t)] \left[\frac{A_C}{2} \text{Sen}(\omega_c t) \right]$$

Donde:

$V_{ASK}(t)$ = voltaje de la onda de amplitud modulada

$\frac{A_C}{2}$ = amplitud de la portadora no modulada (volts)

$V_m(t)$ = señal binaria moduladora (volts)

$\omega_c t$ = frecuencia de la portadora en radianes (radianes por segundo)

2.3.2 Modulación por Desplazamiento de frecuencia (FSK).

El proceso de modulación por desplazamiento de frecuencia cumple los parámetros para la modulación analógica FM, con respecto a la modificación de su frecuencia. Para la modulación FSK, se tiene que los diferentes bits se muestran con diferentes frecuencias las cuales se aproximan a la de la señal portadora como se muestra en la figura N°9.

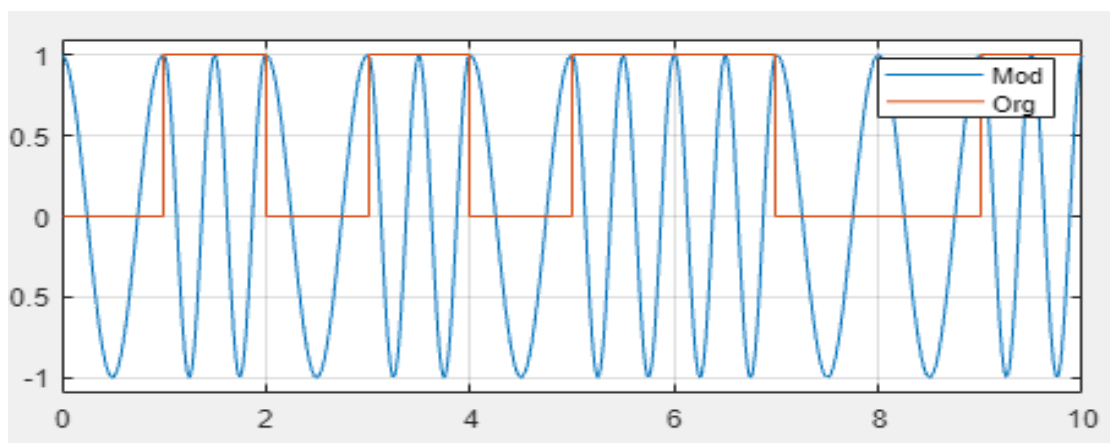


Figura 9. Esquema de modulación FSK. Información tomada directamente del autor. Elaborado por el autor.

La modulación FSK, se entiende también como el procedimiento de realizar cambios de frecuencia de la señal portadora, para realizar una transformación digital de información, en la cual se utilizan diferentes valores de frecuencia para cada uno de los símbolos. El índice de modulación presenta un valor representativo en la señal modulada mediante los cual se obtiene dos tipos de modulaciones FSK.

2.3.3 Modulación Digital de Cuadratura en Fase (QPSK)

La transmisión por desplazamiento de fase en cuadratura (QPSK) o, en cuadratura PSK, como también se la conoce, es otra forma de modulación digital de modulación angular de amplitud constante.

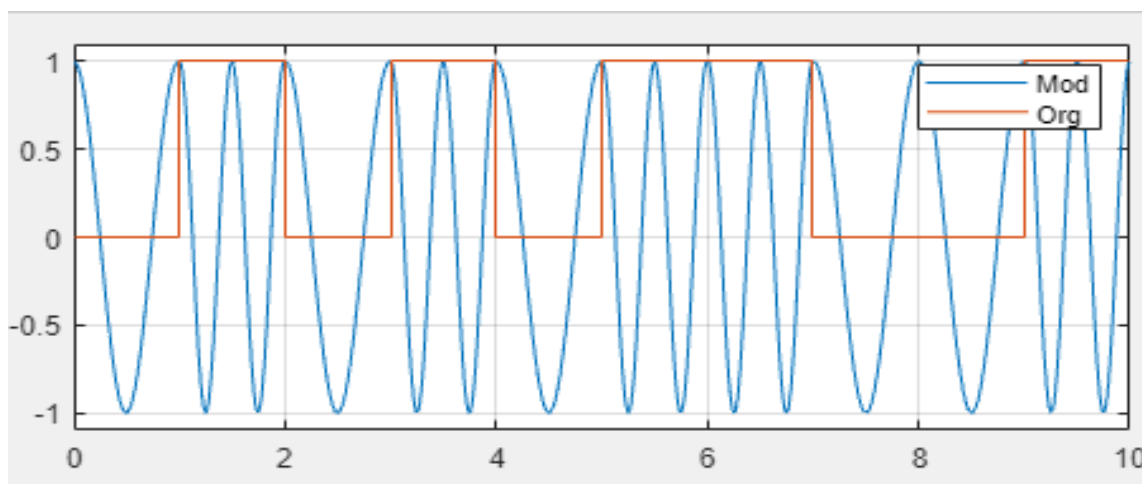


Figura 10. Modulación QPSK. Fuente. Información tomada directamente del autor". Elaborado por el autor.

La modulación QPSK toma bits de entrada, los cuales son dos a la vez, y crea un símbolo que representa una de las cuatro fases. Pero, aun así, el rendimiento de la modulación QPSK se puede mejorar mediante el uso de una técnica conocida como codificación Gray el cual es un sistema de numeración binario en el que dos números contiguos difieren solamente en

uno de sus dígitos. Este código fue diseñado principalmente para prevenir señales ilegales (señales falsas o viciadas en la representación) de los switches electromecánicos, y en la actualidad es utilizado para facilitar la corrección de errores en los sistemas de comunicaciones.

2.3.4 Modulación BPSK

El proceso de modulación BPSK, se realiza produciendo una modificación en la fase cada vez que la señal digital sufre una variación, en la cual se presentan cambios entre 0 y 180 grados respectivamente como se muestra en la figura N°14.

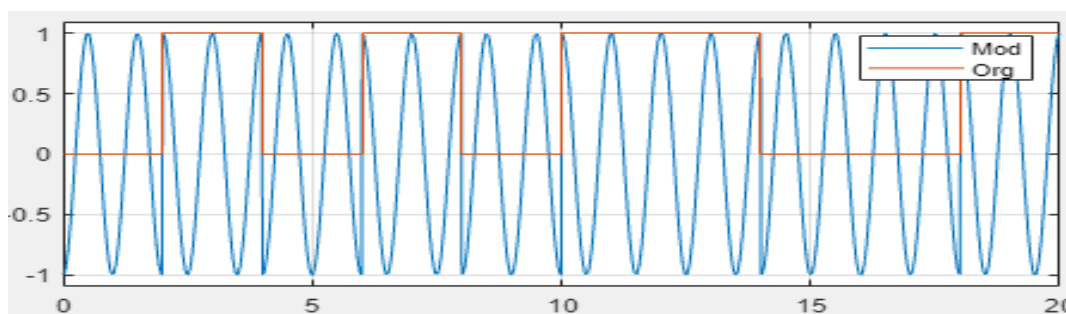


Figura 11. Modulación BPSK. Información adaptada directamente del autor. Elaborado por el autor.

2.3.5 Modulación QAM (8QAM, 16QAM, 32QAM)

La modulación de amplitud en cuadratura es una modulación lineal que consiste en modular en doble banda lateral dos portadoras de la misma frecuencia defasadas en 90 grados. Cada portadora es modulada por una de las dos señales a transmitir, finalmente las dos modulaciones se suman y la señal resultante es transmitida. El resultado de esta modulación se visualiza en la figura n° 12.

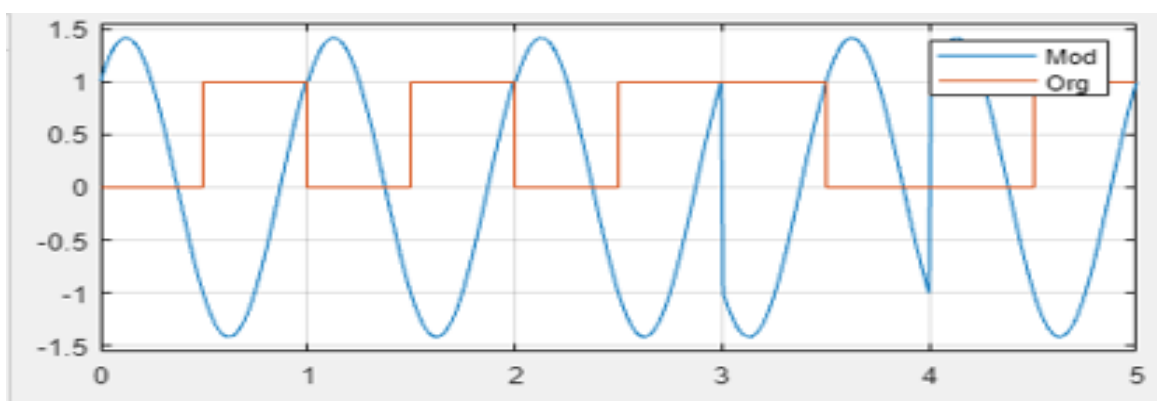


Figura 12. Modulación QAM. Información tomada directamente del autor. Elaborado por el autor

Esto quiere decir que la señal portadora será modificada en amplitud y fase, atendiendo a la señal moduladora, este proceso de mezcla da lugar a la señal modulada. Así se pueden obtener distintas combinaciones de amplitud y fase, dando lugar a los diferentes tipos de modulación QAM que existen:

- 8-QAM
- 16-QAM
- 32-QAM

El proceso de modulación 8 QAM, se basa en la técnica de modulación M-aria en donde se asume $M=8$, la señal que se obtiene de este tipo de modulador no es de amplitud fija, se puede decir que el receptor de 8 QAM, es similar al de la modulación BPSK, siendo su diferencia la información recuperada en la fase de detección y el producto obtenido, en la transformación mediante conversores analógico a digital.

El sistema modulador 16 QAM, se visualiza como un sistema en el cual se tienen 16 posibles combinaciones respecto a sus grupos de bits, su fase como la amplitud de la señal portadora no se mantienen fijos, con los cuales se realiza una división entre los cuatro canales, la salida de información que se obtiene es paralela.

2.3.6. Modulación por Multiplexión por división de frecuencias ortogonales (OFDM)

El proceso de modulación OFDM, es un mecanismo de modulación el cual permite utilizar la interferencia entre símbolos a consecuencia de la propagación multitrayecto presente en el medio de transmisión inalámbrico, se utiliza en las redes wi-fi así como en redes wimax, sobre redes LTE, UWB, DVB, también emplean en los sistemas de transmisión por cable, como ADSL, o PLC.

2.4 Marco legal

Según la (Constitución del Ecuador) el del apartado Título VII “Régimen del Buen Vivir, Capítulo primero, indica en el Art. 350 “sistema de educación superior tiene como finalidad la formación académica y profesional con visión científica y humanista; la investigación científica y tecnológica; la innovación, promoción, desarrollo y difusión de los saberes y las culturas; la construcción de soluciones para los problemas del país, en relación con los objetivos del régimen de desarrollo.”. Basado en este artículo la Universidad de Guayaquil al ser parte del grupo de instituciones de educación superior tiene como objetivo, con los destacar la producción de tecnologías y proyectos que impulsen el avance del país.

Según (Ley Orgánica de Educación Superior) indica “el presupuesto asignado para las diferentes instituciones públicas de educación superior, a su vez los recursos se deberán fundamentar en un marco de calidad y demás criterios establecidos por la ley”, de esta

manera la Universidad de Guayaquil se mantiene con la misión de originar diferentes proyectos que revolucionen la industria con los diferentes pilares de gestión que el proceso requiera.

Según la Resolución de (CEAACES, 2018) indica que el Art 8 del Reglamento ibidem establece que “la evaluación del entorno de aprendizaje mide las condiciones académicas, investigativas, de gestión y organización necesarias para el desarrollo de las carreras en las instituciones de educación superior”, para esto la Universidad de Guayaquil establece diferentes indicadores, que permiten evaluar el avance y desarrollo de la institución.

Segun la (Ley Orgánica de Educación Superior, 2018), indica:

La educación superior ayudará al mejoramiento del conocimiento, a su vez implusará las diferentes áreas de la investigación científica, permitiendo la concesión de nuevas tecnológicas mediante la cual se crearán estudiantes y profesionales con valores en las diversas áreas del conocimiento, con principios eicos, que contribuyan al desarrollo de las diferentes instituciones de la república, dentro del orden democrático y su participación en la sociedad.

Capítulo III

Metodología

3.1 Descripción del Proceso Metodológico

El presente trabajo tiene como finalidad el desarrollo y la implementación de un sistema de modulaciones digitales usando la herramienta App Designer del software MATLAB. Este mecanismo permitirá realizar prácticas de laboratorio con mayor agilidad y sin la necesidad de equipos electrónicos.

Por medio de la presente investigación se concebiría la implementación de un sistema de modulaciones digitales (ASK, FSK, BPSK, QPSK, Y QAM), haciendo uso de la herramienta de interfaz gráfica App Designer del software MATLAB.

Finalizado el proceso de ejecución e implementación del sistema de modulaciones digitales, haciendo uso de la herramienta de interfaz gráfica de MATLAB, se visualizarán los diferentes tipos de modulaciones digitales, lo que ayudará al estudiante a determinar y sacar conclusiones con respecto a las distintas prácticas de laboratorios

3.2 Diseño de la investigación.

El diseño de investigación es de corte transversal puesto que verificará que tan efectiva ha sido el proceso de enseñanza y de esta manera conocer cuál será la utilización que se le dará al software propuesto y los beneficios obtenidos de su creación como herramienta destinada para el proceso de aprendizaje de comunicaciones digitales.

En el presente trabajo de investigación se aplicaron tres métodos de investigación los cuales permitieron incluir todo el campo de investigación requerido, los métodos utilizados fueron método experimental, deductivo, bibliografía.

Dentro de la presente investigación el método bibliográfico, se centrará en la compilación de información de un sistema.

3.3 Enfoque de la Investigación.

La investigación será de tipo descriptiva puesto que especifica los diferentes tipos de modulaciones digitales, bibliográfica basada en las teorías relacionadas a lo que respecta a sistemas de modulaciones digitales y de tipo experimental porque es necesario el uso del laboratorio en el cual se harán las pruebas en el computador para verificar la correcta ejecución del software de modulación digital.

3.3.1 Método Bibliográfico.

El método de investigación bibliográfico se centra en la verificación de estudios relacionados, papers, tesis de grado y libros de sistemas de modulaciones digitales para conocer las diferentes modulaciones digitales que se pueden incluirse (ASK, BPSK, FSK, QPSK y QAM) las mismas que se lograrán desarrollarse en el software que se propone en la presente investigación.

3.3.2 Método Experimental.

El presente trabajo investigativo, pretende incentivar la inclusión de nuevas tecnologías que permitan el desarrollo dentro las practicas, que permitan mejorar el proceso de enseñanza aprendizaje.

Dentro del presente proyecto de investigación se utilizará la herramienta de interfaz grafica App Designer del software MATLAB, el cual se desarrollará un sistema de modulaciones digitales, el cual busca demostrar la operatividad, que asumen los diferentes tipos de modulaciones digitales.

3.3.3 Método Deductivo.

Mediante el método deductivo se opta por definir las diferentes variables que engloba el presente proyecto de investigación, tales como satisfacción, y tiempo utilización, en el cual se utilizó la escala de Likert para determinar el grado de aceptación de la propuesta planteada. Según (Matas, 2018) “Las llamadas escalas Likert” son instrumentos psicométricos donde el encuestado debe indicar su acuerdo o desacuerdo sobre una afirmación, ítem o reactivo, lo que se realiza a través de una escala ordenada y unidimensional.

A través de este método se determinará las diversas inquietudes y opiniones de los estudiantes sobre el manejo del software que permitirá la enseñanza de la metodología teórica-practica mediante la cual se logrará denotar la importancia sobre la utilización de laboratorios equipados, y complementados con herramientas de software y hardware. Se hace uso de la encuesta y su percepción medible a través de la escala de Likert, que encierran las diferentes variables que permitan visualizar la satisfacción, que se obtiene por medio de las prácticas realizadas en el sistema de modulaciones digitales, en conjunto de los estudiantes del penúltimo semestre de la carrera de Ingeniería en Teleinformática.

3.4 Estructuración de la Investigación.

La presente investigación se centra en la aplicabilidad del software para la realización de prácticas de laboratorio de sistemas de modulaciones digitales de los estudiantes de octavo semestre de la carrera de Ingeniería de Teleinformática de la Universidad de Guayaquil, en donde se analizaron los tipos de sistemas de modulaciones digitales a incluirse para obtener resultados favorables en el aprendizaje de los estudiantes.

3.4.1 Población y Muestra.

La población objetivo son los estudiantes de la Carrera de Ingeniería en Teleinformática. Según (HERNÁNDEZ, 2013) “la población como estudio estadístico de un conjunto de elementos con características que se asemejen, de los cuales cada elemento se denomina individuo. Al contar con un gran número de individuos de la población, se toma una parte de la población la cual representará la muestra.” Se considera que la muestra debe ser insesgada que incluya todas las posibles respuestas y eficiente basada en tener una cantidad suficiente de observaciones.

3.4.2 Selección de la Muestra.

En la muestra se seleccionó la universidad de Guayaquil debido a que esta cuenta con un gran número de población estudiantil, por ejemplo, la Facultad de Ingeniería Industrial presenta un total de 3.235 estudiantes repartidos en 5 carreras.

Para este trabajo de investigación se va a seleccionar a los estudiantes de la carrera de Ingeniería en Teleinformática de la universidad de Guayaquil, se tomará en relación los estudiantes del octavo semestre de esta carrera para la muestra los mismo que procederán a utilizar este software.

Actualmente en la carrera de Ingeniería en Teleinformática se encuentran matriculados 622 estudiantes es decir la población $N=622$, se tiene en cuenta la muestra $n=23$ número de alumnos del octavo semestre de la carrera en quienes se enfocará esta investigación.

Se ha tomado esta muestra por las características que tienen en común lo cual facilitará la obtención de datos para esta investigación como son: el nivel de conocimientos, el desenvolvimiento y el tiempo de ejecución del software, de aquí se obtendrá conclusiones al respecto.

Para la recopilación de información se deduce que no se recurre a la misma persona encuestada por lo que cada individuo va a representar cierto

porcentaje y cantidad determinada de personas encuestadas, se representa esta acción como la fracción de muestreo n/N y el factor N/n , obteniendo estos resultados:

Fracción de Muestreo.

$$f = \frac{n}{N} \quad (1)$$

$$f = \frac{23}{622}$$

$$f = 0,036 \times 100\% = 3,6\%$$

El resultado de la muestra representará el 3,6% de la población de estudiantes de Ingeniería en Teleinformática de la Universidad de Guayaquil

Factor de Elevación

$$f = \frac{N}{n} \quad (2)$$

$$f = \frac{622}{23}$$

$$f = 27$$

Con el resultado obtenido cada individuo representa a 27 personas, esto nos ayuda a no repetir la encuesta a las mismas personas.

3.5 Técnicas e instrumentos

Los diferentes elementos que facilitan la obtención de información llevan a cabo diferentes actividades que ayudan al investigador a recolectar información que facilite dar respuesta a las diversas preguntas que fueron establecidas dentro de la presente investigación.

Se puede recalcar técnicas de compilación de datos, como la observación, la entrevista y la encuesta.

Tabla 9. *Técnicas e instrumentos aplicados en la metodología de la investigación*

Técnicas	Instrumentos
Observación	Lista de cotejo Guía de observación Escala de estimación
Encuesta	Cuestionario Test Prueba de conocimiento
Entrevista	Guía de la entrevista

Información adaptada de técnicas e instrumentos de investigación. Elaborador por el autor.

3.5.1 Observación

El proceso de observación es una técnica que consiste en llevar a cabo la visualización, de manera sistemática, cualquier hecho, fenómeno o situación que se genere en la naturaleza o en la sociedad, en relación de objetivos de investigación preestablecidos (Arias F. , 2012).

3.5.1.1. Resultados de la observación

Mediante la observación llevada a cabo en el laboratorio de Networking de la carrera de Ingeniería en Teleinformática, se dio a dilucidar diversos inconvenientes que afectan en el proceso de enseñanza y aprendizaje.

- Se tiene una infraestructura física y lógica deficiente para el proceso de enseñanza práctica.
- Escasos conocimientos referente a los tópicos tratados por parte de los estudiantes, en el aspecto experimental lo cual supone un retraso del programa de catedra impartida por el docente.
- Falta de elementos pertinentes al desarrollo de actividades prácticas.

3.5.2 Resultado de la práctica

Mediante el análisis de las señales digitales en el software se visualizo cómo se comportan las diferentes señales de información enviadas a través del sistema de transmisión, y mediante el proceso de modulación se obtienen resultados fiables y relacionados en la teoría como se muestra en la tabla N°10.

Al finalizar la implementación en software de los sistemas de modulaciones digitales, se realizó una post encuesta, que permitió ratificar la información sobre la adquisición de conocimientos por parte de los estudiantes.

Tabla 10. *Resultados obtenidos proceso de transmisión y recepción*

Parametros	Medio Guiado (UTP-CAT5)	Medio No Guiado (WI-FI)
Distancia	0-5mts	0-5mts
Latencia	0.5ms	0.7ms
Información Recuperada	90%	89%

Información adaptada de técnicas e instrumentos de investigación. Elaborador por el autor.

3.5.3 Encuesta

Según (Casas Anguita, Repullo Labrador, & Donado Campos, 2003) . “La encuesta es una técnica que utiliza un conjunto de procedimientos estandarizado de investigación mediante los cuales se recoge una muestra para analizar casos representativos de un universo o población más amplia, a través de la cual se pretende describir, predecir y/o explicar una serie de características”. la encuesta se realizó a los estudiantes de octavo semestre en el laboratorio de networking, en las horas designadas para la materia de simulación de sistemas, la tabulación de la encuesta se visualiza en el Anexo 2.

3.5.4 Entrevista

Según (Amador,2009) define a la entrevista, como una comunicación interpersonal que se establece entre el investigador y el sujeto de estudio con el fin de obtener las respuestas verbales que se plantean interrogantes sobre el problema propuesto.

3.5.4.1. Resultados de la Entrevista

El desarrollo de la entrevista se llevó a cabo con el Ing. Neiser Stalin Ortiz docente de la materia de Simulación de Sistemas, de la carrera de Ingeniería en Teleinformática de la Universidad de Guayaquil en la cual indicó lo siguiente:

- El laboratorio de Networking, de la carrera de Ingeniería en Teleinformática, no se encuentra en óptimas condiciones para el correcto desarrollo de la cátedra.
 - La práctica mediante circuitos electrónicos asume un papel importante para un correcto entendimiento de la materia, pero el estudiante posee pocos conocimientos
 - A través del proyecto FCI cuyo nombre es **“Desarrollo de un espacio tecnológico en el área de telecomunicaciones mediante sistemas de procesamiento de datos que permitan la generación de sistemas de comunicaciones”**, se tiene como principal objetivo generar un espacio en el cual se realicen prácticas combinando hardware y software, lo cual optimice el proceso de enseñanza-aprendizaje.
- .

Capítulo IV

Desarrollo de la Propuesta

Dentro del presente capítulo se mostrarán el desarrollo del proyecto de investigación, en el cual se detallarán las diferentes herramientas y componentes que se utilizaron para la elaboración e implementación en software de los sistemas de modulaciones digitales realizado en el laboratorio de Networking de la carrera de Ingeniería en Teleinformática de la Universidad de Guayaquil, proyecto el cual tiene como finalidad reducir el tiempo para el entendimiento del funcionamiento de los sistemas de modulaciones digitales, lo cual ayuda a los estudiantes sean capaces de dar soluciones en cualquier momento de su vida estudiantil y profesional.

El tiempo destinado para la utilización del software del sistema de modulaciones digitales fue de 3 horas de la materia de simulación de sistemas de la carrera de Ingeniería en Teleinformática, mediante lo cual se demostró la fiabilidad y confiabilidad del uso del software.

El software del sistema de modulaciones digitales se divide en seis paneles como se puede observar en la figura N° 13 que son:

1. Panel Transmisor: en donde se elige el tipo de modulación digital y la comunicación.
2. Panel de Configuraciones: en donde se establece los valores para el proceso de modulación de la voz.
3. Panel de Conexión: donde se ingresa la ip, puerto de comunicación y el tiempo.
4. Panel de Sensores: donde se eligen los diferentes sensores como: switch, ingreso de voz, texto en tiempo real.
5. Panel Receptor: en donde se recibe la información que llega desde el transmisor
6. Panel de Displays: en donde se puede ver la señal de voz demodulada, los leds en sus dos estados (encendido y apagado)

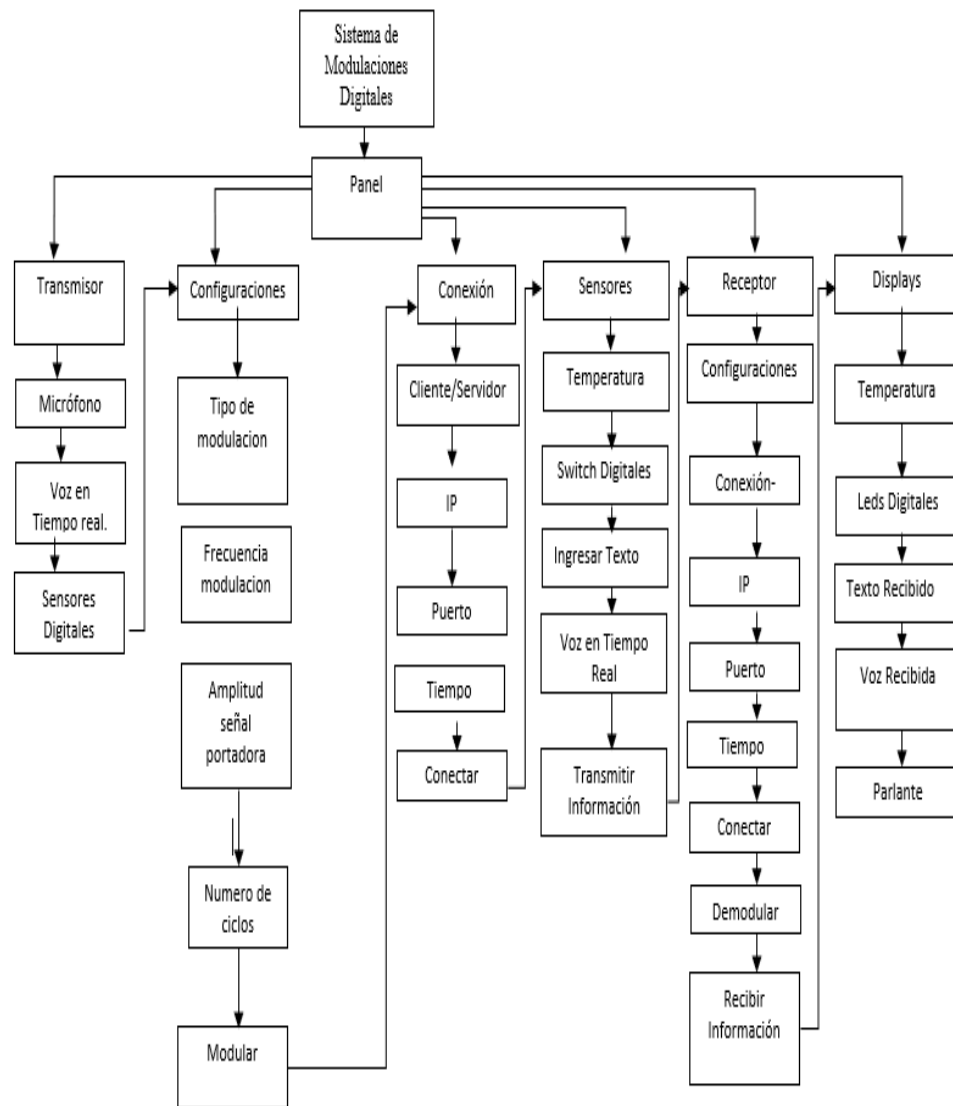


Figura 13. Esquema de Diseño del Sistema de Modulaciones Digitales. Información tomada directamente del autor. Elaborado por el autor.

4.1 Elaboración de la Aplicación con la herramienta App Designer del Software MATLAB

El proceso de implementación en software de las modulaciones digitales se realizará en la herramienta de interfaz gráfica App Designer con la cuenta el programa MATLAB.

Se tendrá como base el software MATLAB, que es un sistema que permite compilar código a manera de script el cual maneja una extensión (archivo .m), o a su vez haciendo uso de su ventana de comando.

El desarrollo de la aplicación se puede observar mediante un diagrama de diseño, como se muestra en la figura N° 14

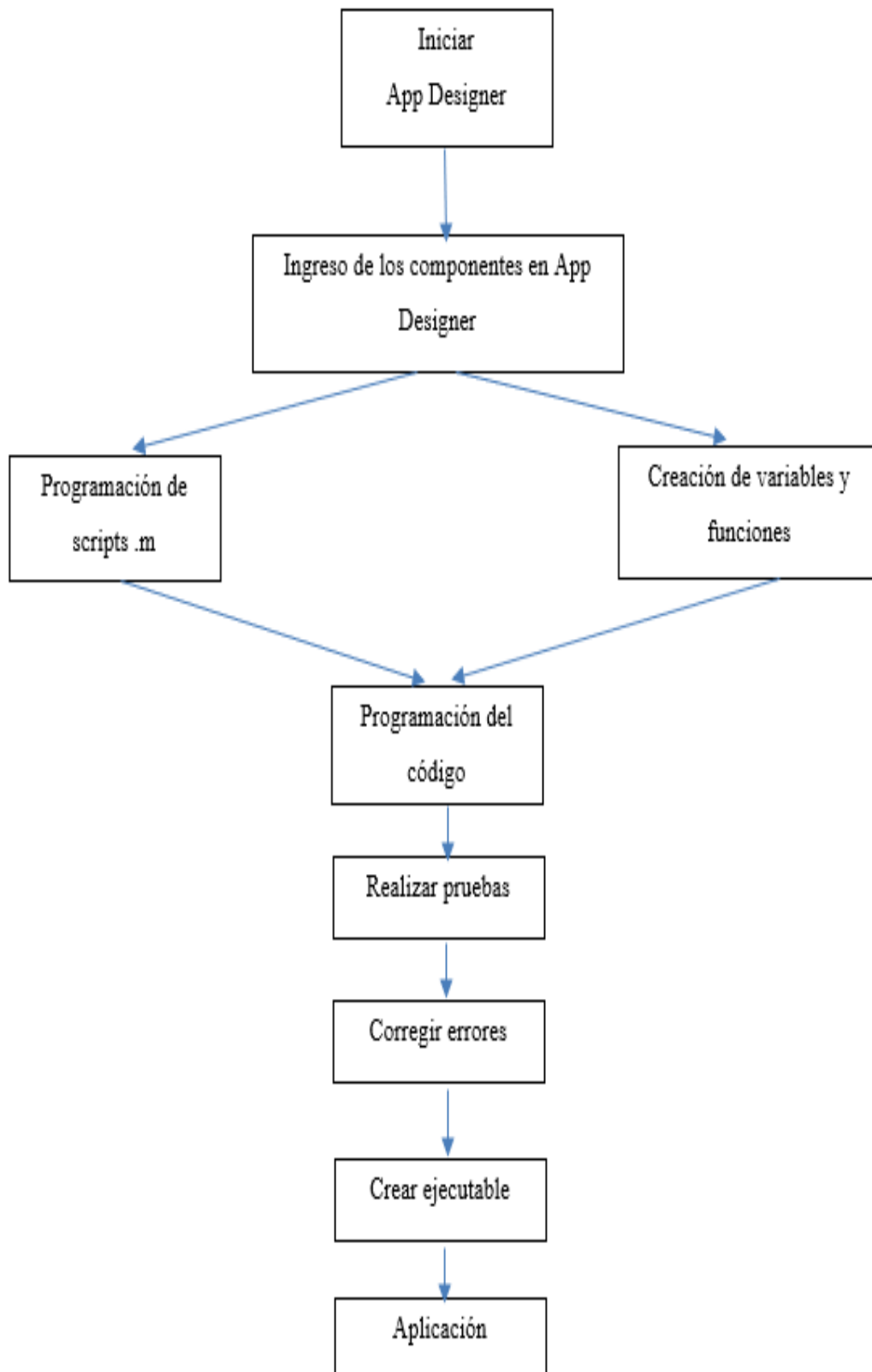


Figura 14. Diagrama de diseño de la aplicación del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

4.1.1 Panel Transmisor

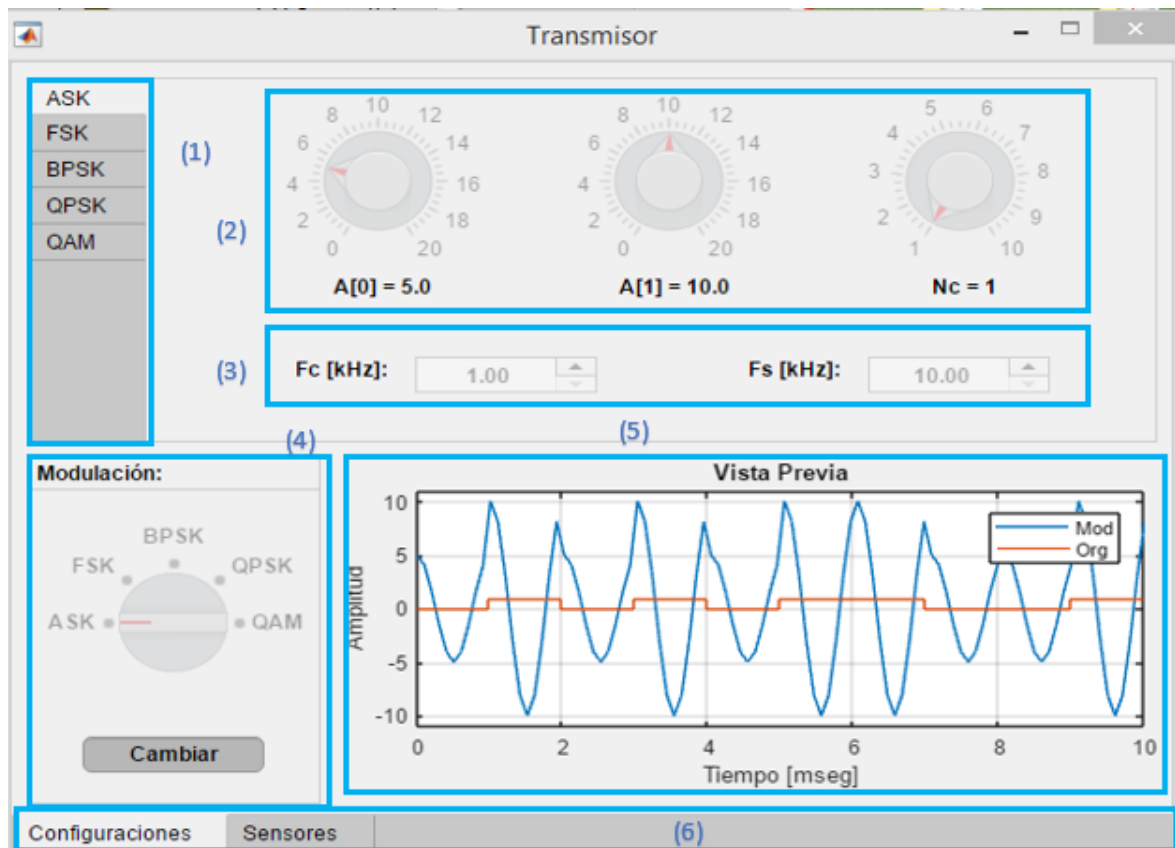


Figura 15. Descripción del panel transmisor. Información tomada directamente del autor. Elaborado por el autor.

Tabla 11. Controles Panel Transmisor

Parámetros	Componentes
Tipos Modulaciones Digitales	Toggle Group Button
A0/A1/Nc	Knob
Fc/Fs	Drop Down
Vista Previa	Axes
Modulación	Discrete Knob
Seleccionar	Button
Configuraciones	Panel
Sensores	Tab

Información adaptada de técnicas e instrumentos de investigación. Elaborador por el autor.

Dentro del panel transmisor, como se puede observar en la figura N° 15 se utilizó un componente toggle group button (Tipos de modulaciones digitales) dentro de este componente (1) tipos de modulaciones digitales (ASK, FSK, BPSK, QPSK, QAM). Se

utilizó tres componentes Knob (A0, A1, Nc) (2), en este componente se eligen manualmente la amplitud de las señales y a su vez el número de ciclos por segundo se utilizaron dos componentes drop down (Fc y Fs) (3), los cuales definen la frecuencia de la señal portadora y la frecuencia de muestreo, se utilizó comandos como read que permite leer la información ingresada, se agregó el componente discrete Knob (Modulación) (4), el cual permite escoger el tipo de modulación digital, que se mostrará en el sistema. (5) se hace uso del comando plot, el cual permite visualizar la gráfica del tipo de modulación que se seleccione. (6), se incluye los botones configuraciones y sensores que permiten enviar la información desde el transmisor hacia el receptor como se muestra en el manual de usuario en el Anexo1.

4.1.2 Panel Transmisor-Sensores

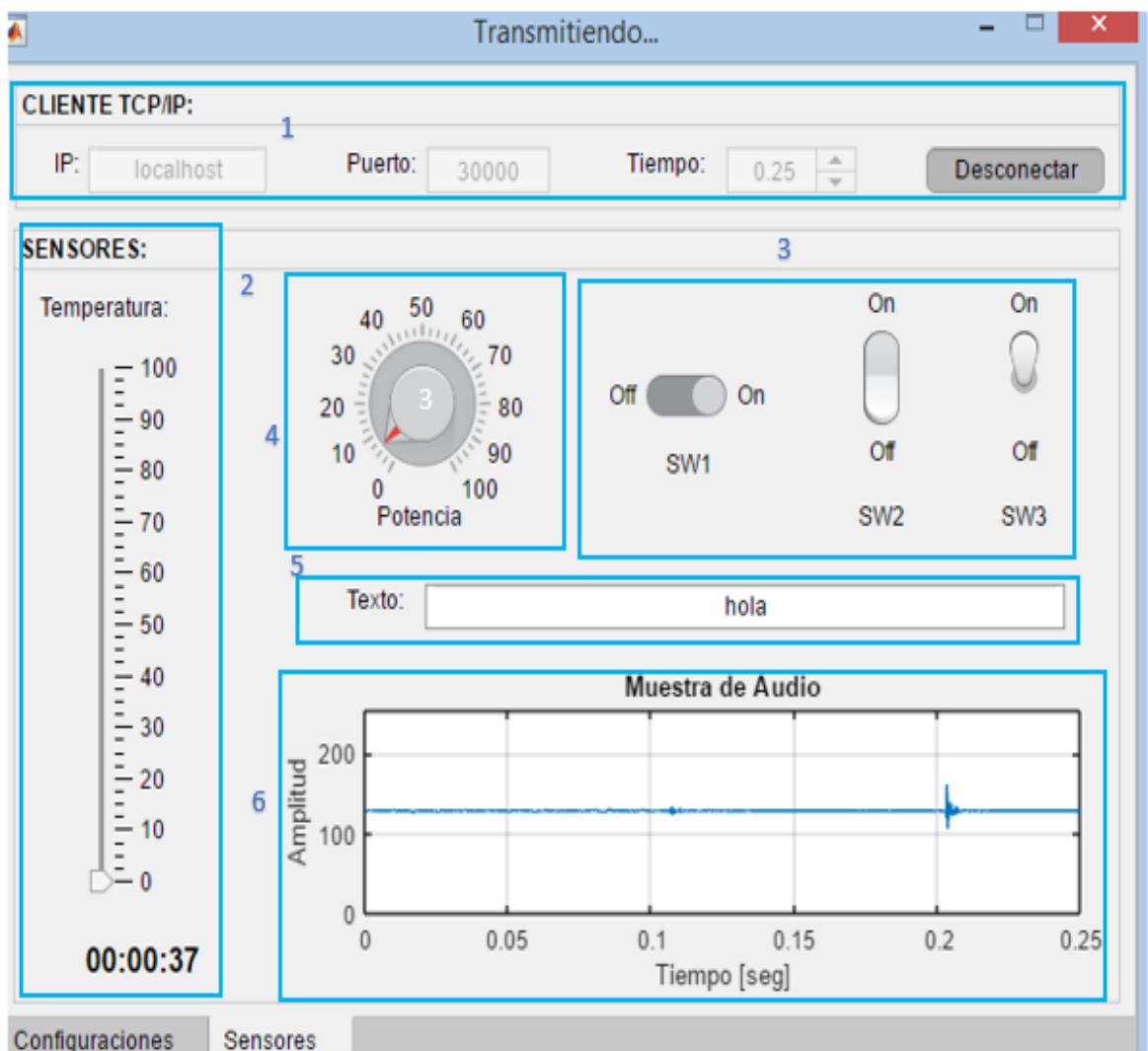


Figura 16. Descripción del panel transmisor-sensores. Información tomada directamente del autor Elaborado por el autor.

Tabla 12. *Controles Panel Transmisor-Sensores*

Parámetros	Componentes
Potencia	Toggle Group Button
SW1/SW2/SW3	Knob
Texto	Drop Down
Temperatura	axes
Muestra de Audio	Discrete Knob
Tiempo	Spinner
Conectar	Button

Información adaptada de técnicas e instrumentos de investigación. Elaborador por el autor.

En la tabla N°12, se describen los componentes utilizados para el desarrollo del panel de controles en el transmisor como se puede ver en la figura N° 19 , dentro de la presente ventana se utilizan apuntadores a las funciones previamente establecidas a través de los diferentes archivos.m de los tipos de modulaciones y demodulaciones digitales como askmodulacion,fskmodulacion, (1) se colocan la dirección ip y el puerto de comunicación para realizar el proceso de transmisión y recepción y a su vez el tiempo en milisegundos , se utiliza comandos como Timer los cuales se pasa por parámetro el audio grabado, cantidad de bits y amplitud de señal portadora.(2)Temp, SW1,SW2,SW3(3), (4)potencia, utilizan comandos como position el cual permite determinar su posición dentro de la aplicación se utiliza comando como enable, stop para indicar el inicio de la transmisión. (5) se utiliza un edit fiel (text),en este panel se realiza el envío de texto de prueba que luego es recuperado a través del sistema receptor, (6) se utiliza un axes el cual presenta la muestra de audio enviada desde el transmisor de igual manera se logra realizar el proceso de transmisión y recepción haciendo uso de los diferentes controles el desarrollo del código se puede visualizar en el Anexo 2.

4.1.3 Panel Receptor-Configuraciones

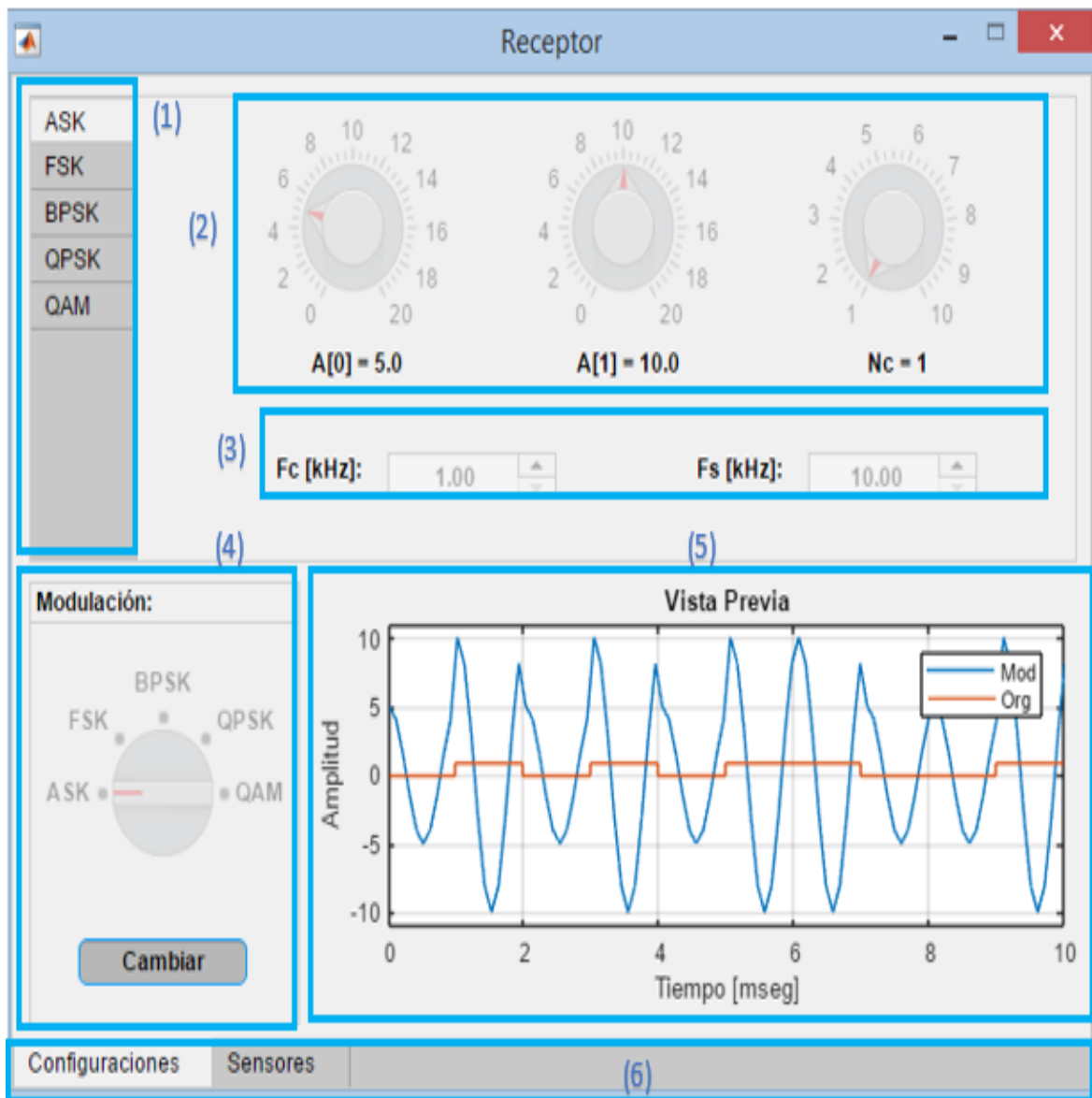


Figura 17. Descripción del panel Receptor-Configuraciones. Información toma directamente del autor
Elaborado por el autor.

En el presente panel mediante el cual se realiza la transmisión se utilizan diferentes componentes que se describen en la tabla N°13. (1) se utiliza un componente group button en el cual se selecciona el tipo de modulación. (2) tienen tres componentes Knob en los cuales se selecciona A_0 , A_1 , N_c , (3) se tiene 2 componentes drop down button, los cuales denotan la frecuencia de corte y la frecuencia de muestreo, (4) se usa un componente discrete Knob, mediante el cual se selecciona o se cambia el tipo de modulación digital, (5) se utiliza un componente axes, el cual se muestra una vista previa del tipo de modulación digital seleccionada, (6) se utiliza un componente tab group, en el cual se despliegan dos pestañas una de configuraciones y otra de sensores.

Tabla 13. *Controles Panel Transmisor-Sensores*

Parámetros	Componentes
Ip/Puertos	Edit Field (Numeric):
Tiempo	Drop Down Button
Conectar	Button
Temperatura	Linear Gauge:
Led1/Led2/Led3	Lamp
Texto de Prueba	Edit Field (Text)

Información adaptada de técnicas e instrumentos de investigación. Elaborador por el autor.

4.1.4 Panel Receptor- Sensores

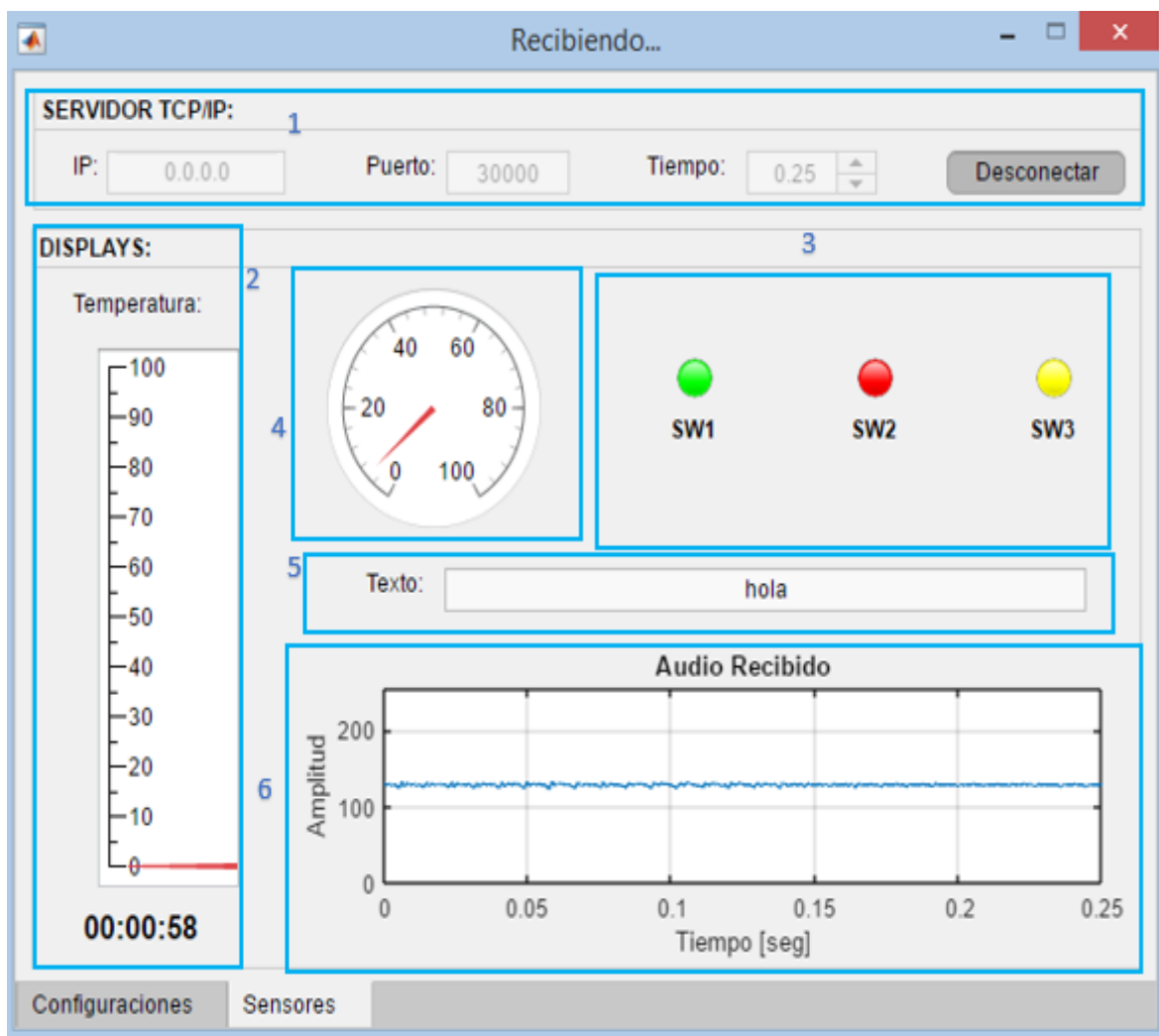


Figura 18. Descripción del panel receptor -sensores. Información toma directamente del autor Elaborado por el autor.

En este panel cuyo nombre es sensores como se muestra en la figura N°18, En el componente servidor TCP/IP (1), se utilizó comandos como tcpip, strcmpi, los cuales permiten realizar la comunicación, (2) se insertó un componente display, en donde se muestra las variaciones del sensor de temperatura recibida desde el sistema transmisor, (3) se tienen tres componentes lamp(SW1,SW2,SW3), los cuales reciben las señales digitales de encendido y apagado desde el sistema transmisor, (4) se utilizó un componente Gauge, el cual recibe y muestra las variaciones del sensor de potencia por parte del transmisor.(5) se cuenta con un componente edit field text, dentro de este componente se muestra el texto recibido desde el sistema transmisor, (6) se utilizó un componente axes en el cual se muestra la voz recibida en tiempo real por parte del sistema transmisor.

Tabla 14. *Controles Panel Receptor-Sensores*

Parámetros	Componentes
Ip/Puertos	Edit Field (Numeric)
Tiempo	Drop Down Button:
Conectar	Button:
Temperatura	Linear Gauge:
Led1/Led2/Led3	Lamp:
Texto de Prueba	Edit Field (Text)

Información adaptada de técnicas e instrumentos de investigación. Elaborador por el autor

4.2 Implementación de la aplicación de modulaciones digitales

Para el presente proyecto de implementación del sistema de modulaciones digitales, se utilizaron dos laptops como se muestra en la figura n°19, en donde una será usada como transmisor y la otra será el receptor, a su vez también se utilizaron dos tipos de comunicación a través de un medio guiado (cable de par trenzado cat5) y también haciendo uso de un medio no guiado (Wi-Fi, basado en el estándar IEEE 802.1n).

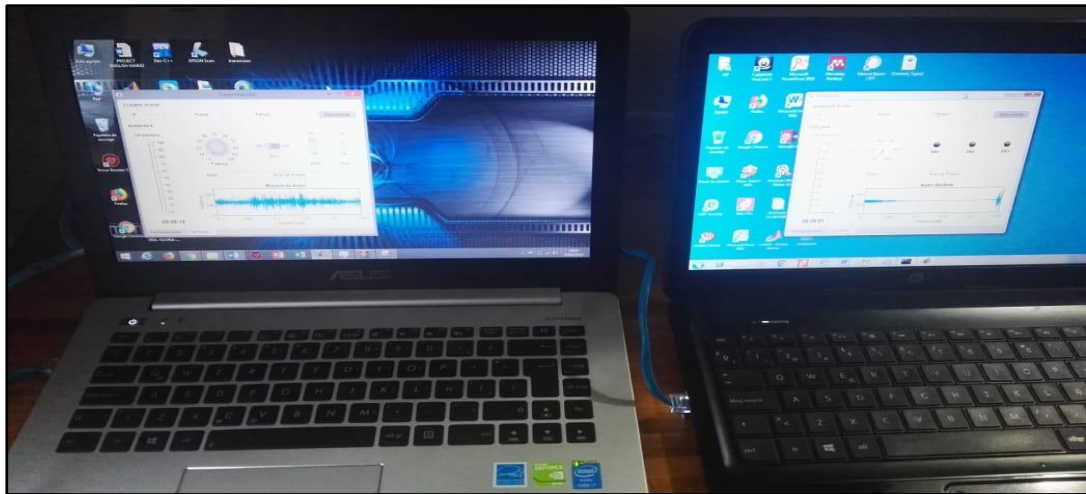


Figura 19. Transmisión y recepción ASK utilizando un canal de comunicación alámbrico. Información tomada directamente del autor. Elaborado por el autor.

4.2.1 Evaluación 1. Transmisión y Recepción ASK a través de medio guiado (Cable par trenzado UTP cat5)

Se conectan dos computadores dentro del mismo segmento de red y se realiza su comunicación utilizando un cable ethernet. En la figura n° 20 se visualiza la implementación del software en el laboratorio de networking con los estudiantes del octavo semestre.



Figura 20. Implementación del sistema de modulaciones digitales. Información tomada y adaptada directamente del autor. Elaborado por el autor.

Se ingresa al primer panel (Transmisor) como se observa en la figura N° 21, donde se selecciona el tipo de modulación digital en la cual se ingresa los valores de la amplitud de la señal portadora, la frecuencia de muestro y frecuencia de corte y el número de ciclos para la señal que va a ser modulada.

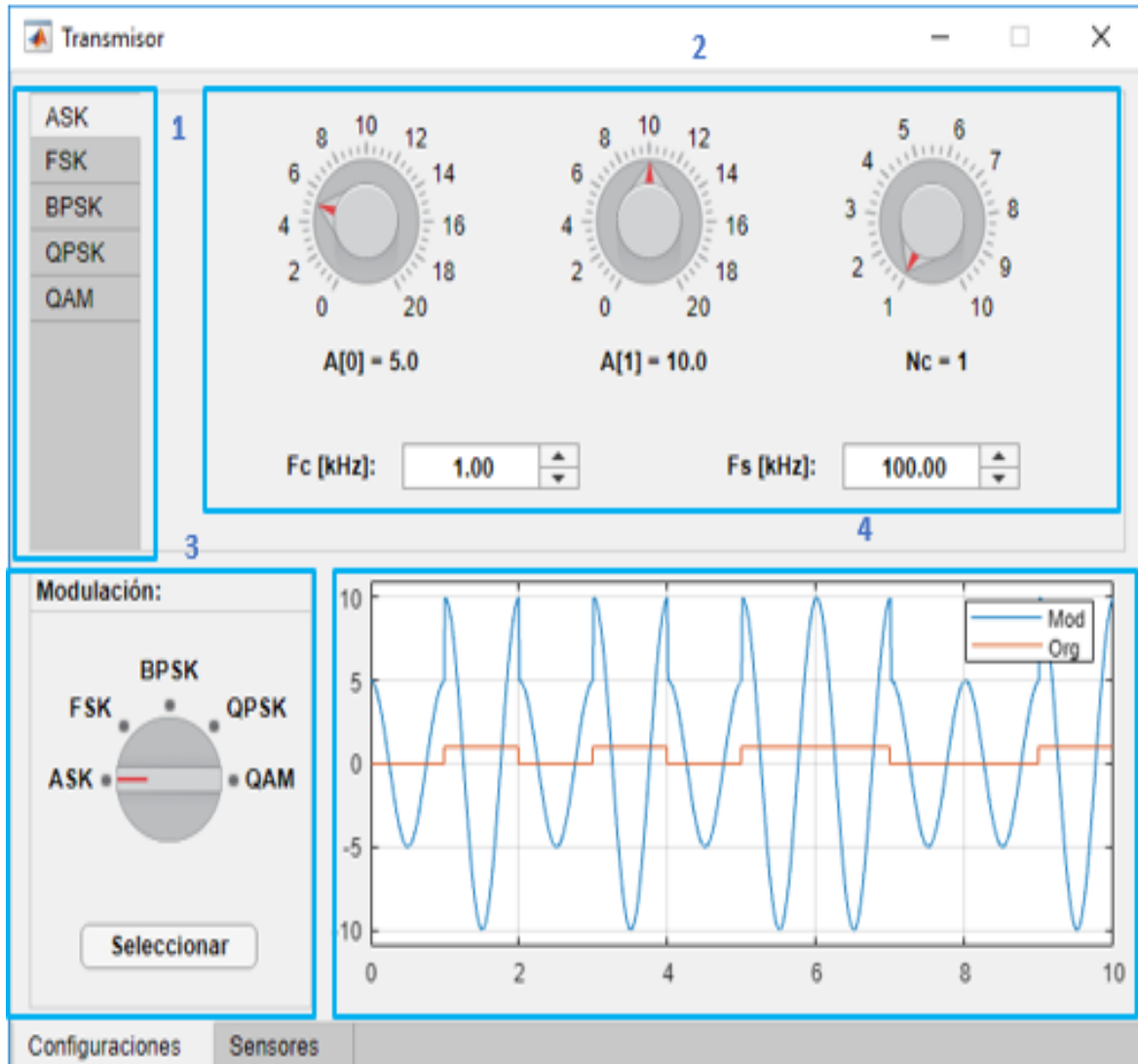


Figura 21. Panel 1: Transmisor/Configuraciones. Información tomada directamente del autor. Elaborado por el autor.

En el segundo panel (Transmisor- Sensores), luego de haber seleccionado los parámetros iniciales para la modulación digital, se da clic en la pestaña de sensores en la cual se procede a ingresar los datos a transmitir en tiempo real, así como la transmisión de la señal de voz.

Para realizar la modulación ask se obtiene la señal recibida en tiempo real Txsignal, la cual se modula en consideración al número de bits, la frecuencia de corte que se establezca en los parámetros iniciales como son frecuencia de muestro, frecuencia de corte, numero de ciclos, numero de bit, y amplitud de la señal portadora.

Luego que se selecciona el tipo de modulación y los parámetros iniciales se debe pasar a la pestaña de sensores como se puede visualizar en la figura N°22, donde se observan los controles, así como el ingreso de la voz en tiempo real.

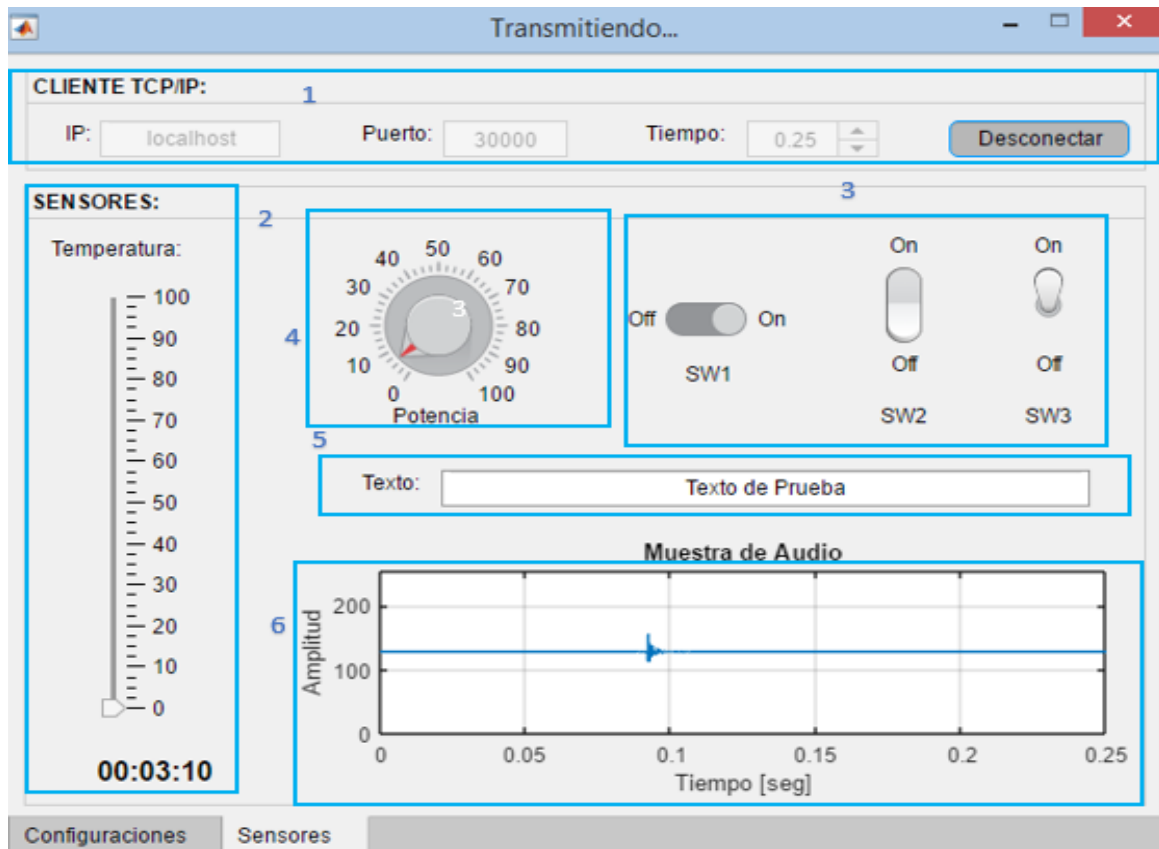


Figura 22. Panel 2: Transmisor-Sensores. Información tomada directamente del autor. Elaborado por el autor.

En la figura N°23 se muestra el panel receptor, en el cual se deben incluir las configuraciones iniciales establecidas anteriormente en el panel de transmisión.

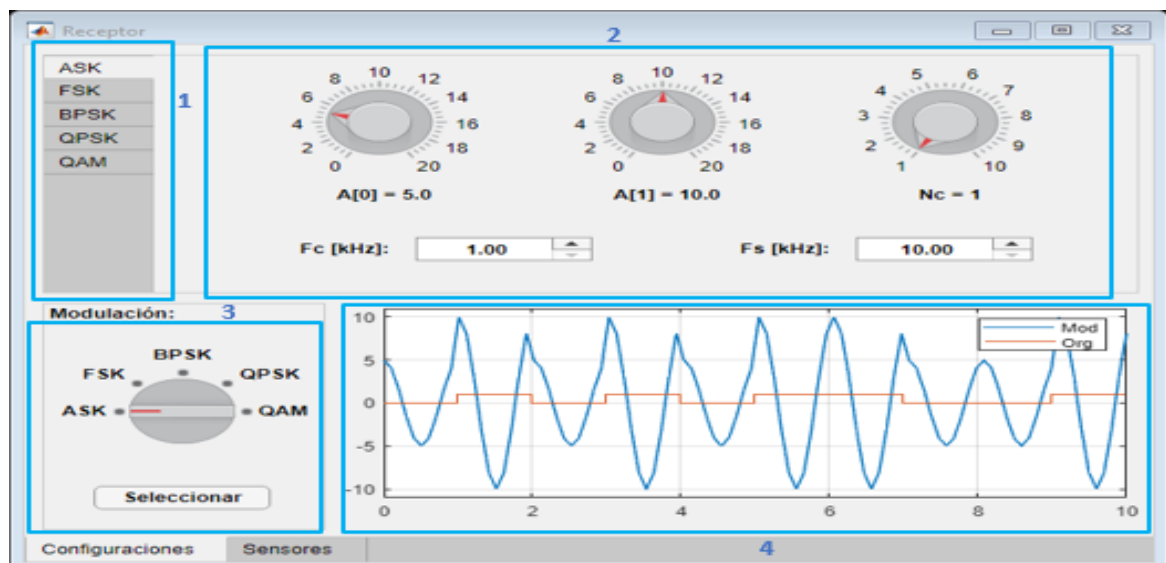


Figura 23. Panel 3: Receptor. Información tomada directamente del autor. Elaborado por el autor

En el cuarto panel receptor (Sensores), como se muestra en la figura N°24, se puede visualizar la recepción de la información que llega a través de la red desde el panel transmisor.

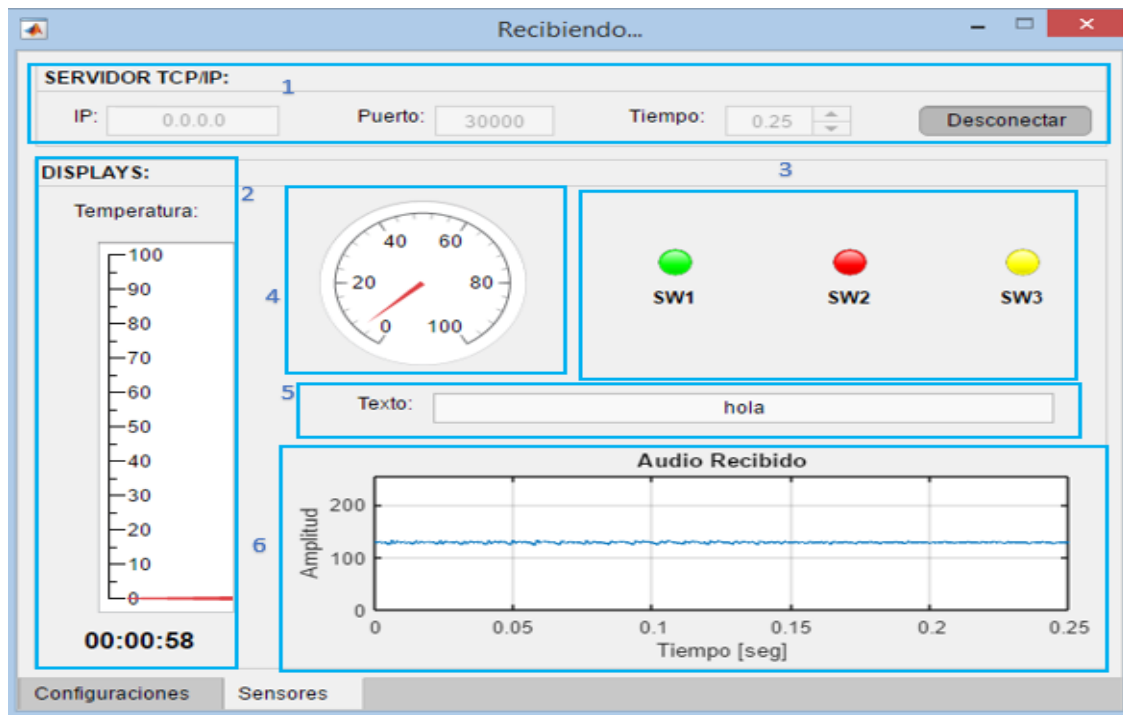


Figura 24. Panel 4: Receptor/sensores Información tomada directamente del autor. Elaborado por el autor.

4.2.2 Evaluación 2. Transmisión y Recepción ASK a través de medio no guiado (red Wi-Fi con el estándar IEEE 802.11n)

Se conectan dos computadores dentro del mismo segmento de red y se comunican utilizando la conexión Wi-Fi, como se visualiza en la figura n° 25.

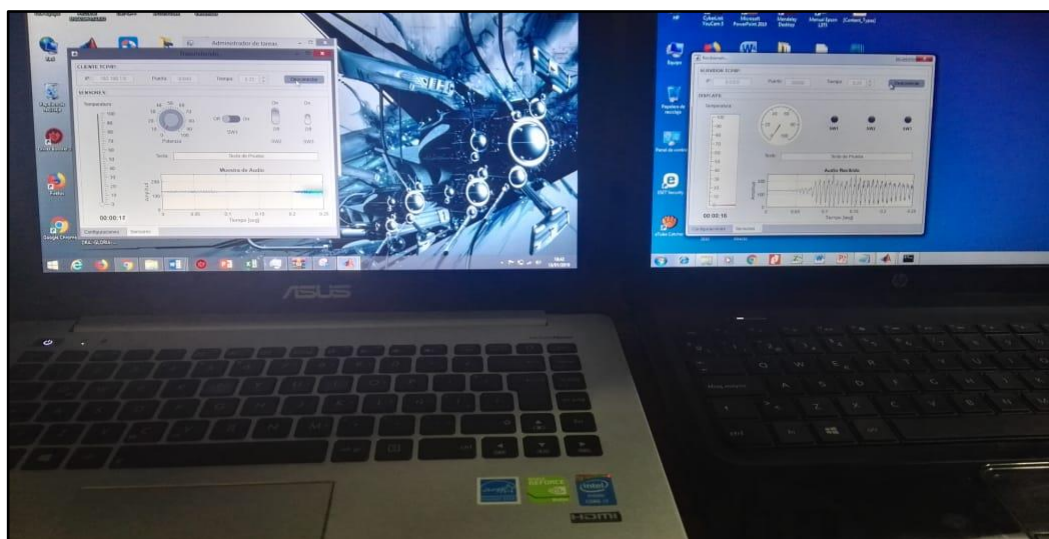


Figura 25. Transmisión y recepción Digital utilizando un canal de comunicación inalámbrica. Información tomada directamente del autor. Elaborado por el autor.

Se ingresa al primer panel (Transmisor) como se observa en la figura N° 25, donde se selecciona el tipo de modulación digital en la cual se ingresa los valores de la amplitud de la señal portadora, la frecuencia de muestro y frecuencia de corte y el número de ciclos para la señal que va a ser modulada.

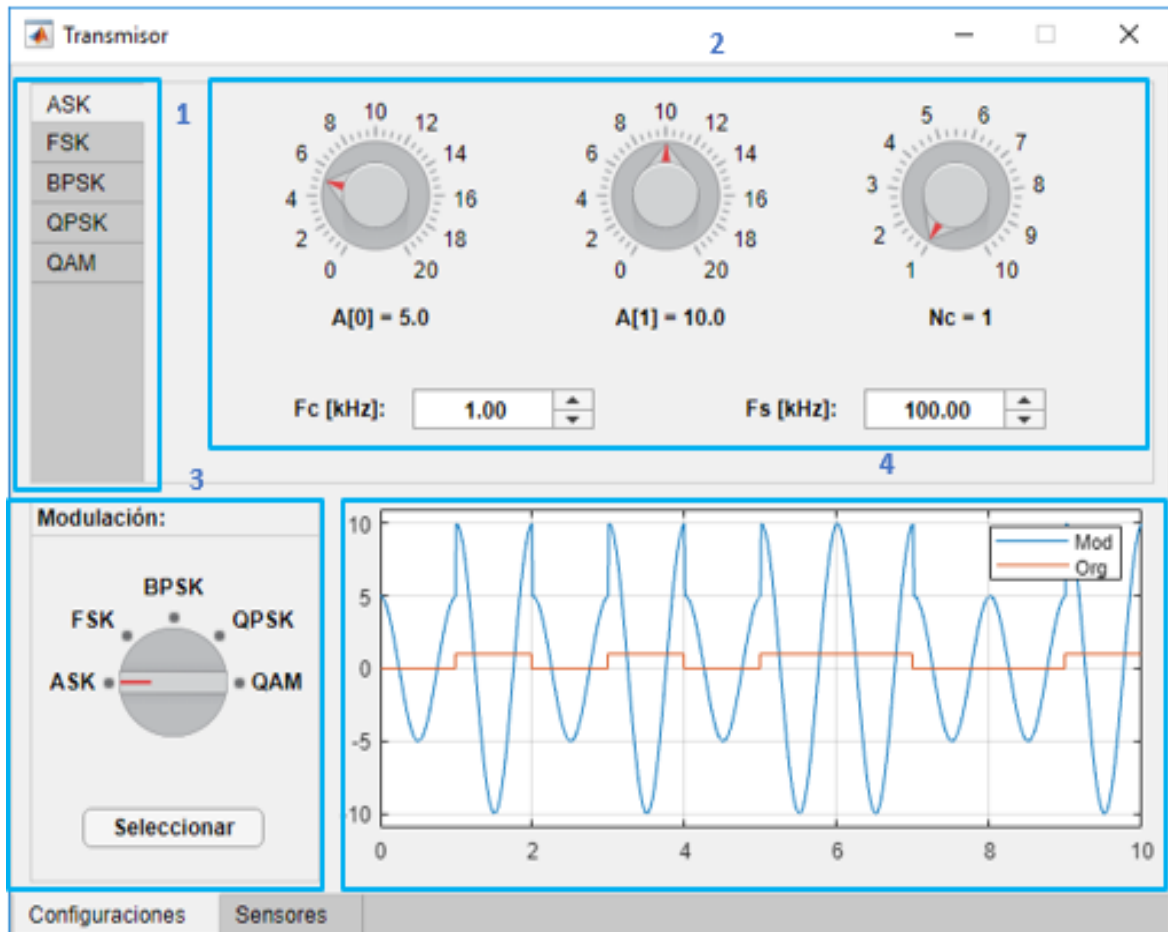


Figura 26. Panel 1: Transmisor/Configuraciones. Información tomada directamente del autor. Elaborado por el autor.

En el segundo panel (Transmisor /sensores), luego de haber seleccionado los parámetros iniciales para la modulación digital, se da clic en la pestaña de sensores en la cual se procede a ingresar los datos a transmitir en tiempo real, así como la transmisión de la señal de voz.

Para realizar la modulación ask se obtiene la señal recibida en tiempo real Txsignal, la cual se modula en consideración al número de bits, la frecuencia de corte que se establezca en los parámetros iniciales como son frecuencia de muestro, frecuencia de corte, numero de ciclos, numero de bit, y amplitud de la señal portadora.

Luego que se selecciona el tipo de modulación y los parámetros iniciales debemos pasar a la pestaña de sensores como se puede visualizar en la figura N°26, donde se observar controles, así como el ingreso de la voz en tiempo real.

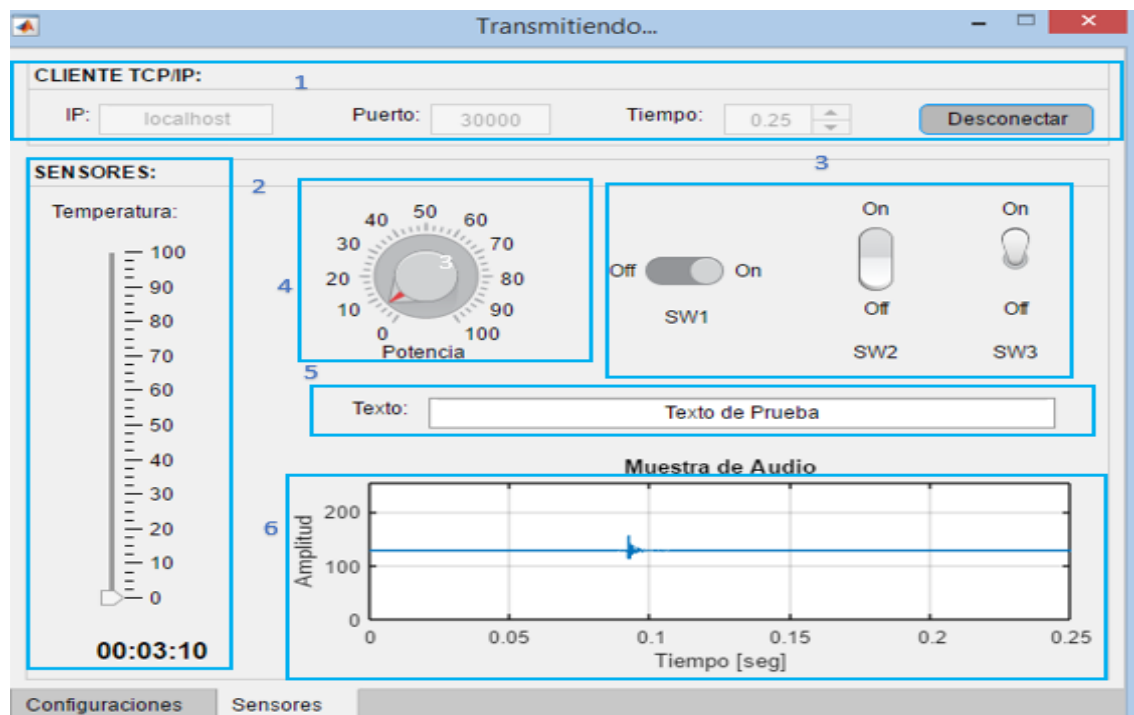


Figura 27. Panel 2: Sensores. Información tomada directamente del autor. Elaborado por el autor.

En la figura N°27 se muestra el panel receptor, en el cual se deben incluir las configuraciones iniciales establecidas anteriormente en el panel de transmisión.

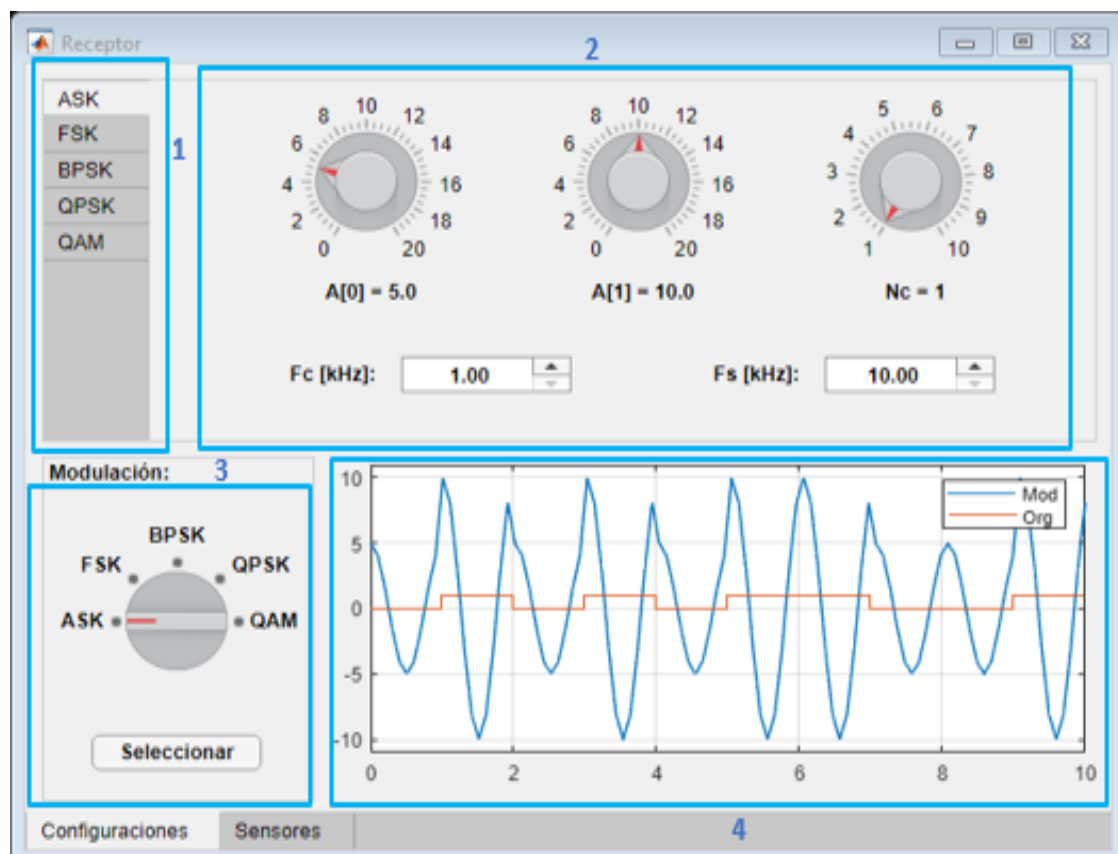


Figura 28. Panel 3/ Receptor. Información tomada directamente del autor. Elaborado por el autor.



Figura 29. Panel 4: Receptor/sensores Información tomada directamente del autor. Elaborado por el autor.

En el cuarto panel receptor (Sensores), como se muestra en la figura N°29, se puede visualizar la recepción de la información que llega a través de la red desde el panel transmisor.

Se analizó el tiempo de la metodología teórica y práctica, lo cual se determina la eficiencia, en la cual la metodología practica optimiza el tiempo llevado a cabo por los estudiantes, lo cual conlleva un mayor conocimiento respecto a los temas tratados en clases, lo cual permite que la materia se lleve de una manera más globalizada.

4.3 Resultado de la práctica

Mediante el análisis de las señales digitales en el software se visualizó como se comportan las diferentes señales de información enviadas a través del sistema de transmisión, y mediante el proceso de modulación se obtienen resultados fiables y relacionados en la teoría como se muestra en la tabla N°15.

Al finalizar la implementación en software de los sistemas de modulaciones digitales, se realizó una post encuesta, que permitió ratificar la información sobre la adquisición de conocimientos por parte de los estudiantes.

Tabla 15. Resultados obtenidos proceso de transmisión y recepción

Parametros	Medio Guiado (UTP-CAT5)	Medio No Guiado (WI-FI)
Distancia	0-5mts	0-5mts
Latencia	0.5ms	0.7ms
Información Recuperada	90%	89%

Información adaptada de técnicas e instrumentos de investigación. Elaborador por el autor.

A través de las pruebas realizadas se obtiene los siguientes resultados como se muestra en la tabla N° 15, en la cual se tomo como referencias parámetros de distancia, latencia, y porcentaje de información recuperada, en la ejecución del sistema se hicieron dos tipos de pruebas, haciendo uso de un medio guiado (cable UTP) y un medio no guiado para lo cual se utiliza la red wi-fi. Se obtiene que de acuerdo a la distancia planteada de 0-5mts utilizando el medio guiado la latencia que se genera es de 0.5milisegundos, a su vez haciendo uso del medio no guiado se produce una latencia del 0.7 milisegundos, con respecto al porcentaje de información recuperada se obtuvo el 90% de la información enviada a través del transmisor hacia el receptor mediante la utilización del medio guiado, y utilizando la red wi-fi(medio no guiado) se recupero un 89% de la información transmitida hacia el receptor.

4.4 Desarrollo de la Encuesta

Se procederá a realizar un post encuesta sobre la implementación en software de los sistemas de modulaciones digitales, en la cual se usó del método de Likert, en el cual se toma en cuenta cuatro niveles, medibles desde los de mayor aceptación, hasta los menos aceptables.

Las encuestas se encuentran dirigidas a los estudiantes de octavo semestre 2018-2019 CII de la carrera de Ingeniería en Teleinformática, luego se procede con la tabulación de los resultados de las encuestas como se visualiza en el ANEXO 2, lo cual permite tener una conclusión sobre la realización de prácticas, lo cual permite denotar la veracidad del proyecto de implementación.

4.5 Análisis de encuesta

Haciendo uso del modelo de Likert se determinó la importancia de contar con la metodología teórica-practica, permitiendo a los estudiantes comprender de manera eficaz la

cátedra.

Se realizó una post-encuesta con la cual se corroboró la importancia de la aplicación de la metodología teórica-práctica, en donde se observó que la gran mayoría de estudiantes considera que se obtiene un mayor entendimiento entre la conjunción de clases teóricas y clases prácticas, lo cual hace imprescindible contar con laboratorios adecuados.

Mediante los datos obtenidos se logró determinar que las prácticas en el proceso de enseñanza-aprendizaje utilizando software, asume un papel importante como se observa en la tabla n°16, en donde se agrupan los porcentajes de las opiniones obtenidas sobre las preguntas 1,2,3,4 que se centran en la optimización de tiempo y reducción de recursos en el momento de realizar una implementación que permita obtener una mejor comprensión y adquisición de los conocimientos.

Tabla 16. *Porcentaje de opinión sobre la metodología teórica-práctica.*

Nº	Totalmente de acuerdo	De acuerdo
1	74%	22%
2	48%	48%
3	65%	35%
4	39%	52%

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor.

Se toma como referencia los resultados obtenidos en la tabla N°16, con los cuales se concluye que existe un alto grado de aceptación por parte de los estudiantes sobre la propuesta del sistema de modulaciones digitales y también se obtiene que los estudiantes encuestados están de acuerdo en que el desarrollo de la metodología práctica debe ser utilizada en la carrera de Ingeniería en Teleinformática.

Tabla 17. *Porcentaje de opinión sobre conocimientos del software Matlab y App Designer*

Nº	Totalmente de acuerdo	De acuerdo
5	61%	39%
6	39%	52%
7	39%	57%
8	65%	35%
9	83%	17%
10	70%	30%

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor.

Como se muestra en la tabla N° 17, los resultados obtenidos en la encuesta realizada

denotan la necesidad de tener conocimientos sobre uso del software Matlab y App Designer, con lo cual se hace mención a las preguntas 5, 6, 7, 8, 9, 10.

4.6 Análisis de la hipótesis

Después de implementar en software los sistemas de modulaciones digitales se verifico que el proyecto permite dinamizar conocimientos adquiridos sobre la modulación digital.

A través de la encuesta realizada se pudo comprobar que el uso de una metodología teórica-práctica ayuda a los estudiantes a tener mejores conocimientos sobre los tipos de modulaciones digitales, se pudo confirmar la necesidad de que los estudiantes como docentes se encuentren capacitados en el uso del software.

Mediante una tabla de verdad se realizará la comprobación del cumplimiento de la hipótesis planteada en el capítulo I.

H1: La unificación de un sistema de modulaciones digitales complementará el conocimiento teórico, mediante la utilización de software, lo cual permitirá impulsar el desarrollo académico de los estudiantes de la carrera de Ingeniería en Teleinformática.

Luego de implementar el sistema de modulaciones digitales se pudo corroborar que el uso de la aplicación permite visualizar y realizar un análisis, de manera eficaz las técnicas de modulaciones digitales (ASK,BPSK,FSK,QPSK,QAM), por la cual favorece la comprensión de los conocimientos que se reciben en relación a estos tópicos de la materia, haciendo uso de las encuestas se pudo verificar la aceptación de la metodología teórica-práctica, la cual asume un rol fundamental en el proceso de enseñanza y aprendizaje, la cual es de una gran importancia dentro de las materias de carácter técnico, y de esta manera aportar a una formación académica de calidad para el estudiante.

C: Mediante la hipótesis planteada, se respondió de manera favorable que permitieron otorgar veracidad al desarrollo del presente trabajo de investigación.

Se denota la siguiente condicional:

$$H1 \rightarrow C$$

Tabla 18. *Tabla de verdad*

H1	C	H1→C
1	1	1

Investigación tomada directamente del autor. Elaborado por el autor

Como se muestra en la tabla N°18 el resultado es verdadero, con lo cual se demuestra el cumplimiento de lo propuesto al inicio del presente trabajo de investigación, con lo cual se finaliza este proyecto con resultados positivos.

4.7 Conclusiones

Mediante el desarrollo del presente trabajo de investigación, la implementación del sistema propuesto, la información y resultados obtenidos de las encuestas se concluye:

El desarrollo de prácticas de laboratorio en las carreras de ingeniería es de suma importancia, debido a que permiten a los estudiantes obtener una mejor formación académica, que les permite un correcto desempeño profesional.

MATLAB, cuenta con diversas herramientas que permiten optimizar el tiempo y obtener resultados precisos y seguros, con la cual se logra un mejor desempeño de actividades prácticas.

Los espacios de la carrera de Ingeniería en Teleinformática no se encuentran en óptimas condiciones que permitan realizar prácticas, con la cual se obtienen diversos inconvenientes al momento de asumir el desarrollo de proyectos de las materias de carácter técnico.

Se deduce respecto a las pruebas del sistema realizado que se debe ingresar los mismos datos en el panel transmisor y el panel receptor, antes de realizar la modulación digital, en el momento de iniciar con la transmisión en tiempo real de la señal mensaje(voz), como se muestra en la figura N°22.

El proceso de transmisión y recepción de la información se realizó a través de dos paneles siendo el transmisor y el receptor, verificando la entrada y salida de información a través de sensores digitales y axes para verificar la modulación de la voz.

Mediante la encuesta se demostró que el 74% se encuentra totalmente de acuerdo y un 22%, se encuentra de acuerdo en que las prácticas son un principal apoyo para el proceso de enseñanza aprendizaje de los estudiantes.

Al respecto de las características de diseño y ubicación de los elementos de la aplicación, se ha demostrado el alcance de los objetivos planteados, con un 39% están totalmente de acuerdo y un 57% de acuerdo, en que el manejo de la aplicación es de fácil uso, con lo cual se determinó que un estudiante puede obtener en cualquier momento una visión del nivel en que se encuentre, y los procedimientos que debe seguir en la utilización del software.

Se comprobó que el 48% de los estudiantes encuestados está totalmente de acuerdo, y de igual manera un 48% está de acuerdo; que asumir un mayor uso del software permite reducir costos y tiempo.

Con la utilización del sistema se denotó que el 65% de los estudiantes están totalmente de acuerdo y el 35% están de acuerdo que se obtiene una mejor comprensión y adquisición de los conocimientos sobre los sistemas de modulaciones digitales.

A través de la implementación del software se determinó que el 70% está totalmente de

acuerdo y el 30% de acuerdo en que las clases de carácter teórico y práctico son de principal apoyo para que los estudiantes puedan adquirir mejor los conocimientos impartidos en la catedra.

Los resultados que se obtuvieron a través de las encuestas realizadas, permitieron medir el grado de aceptación, por parte de los estudiantes, mediante las diferentes preguntas planteadas; con lo cual se obtuvo una respuesta positiva, con lo cual se realizó el análisis del presente trabajo de investigación.

4.8 Recomendaciones

A través del presente trabajo de investigación, el resultado que se obtuvo en base a las encuestas, y la implementación del software se recomienda:

La optimización de espacios técnicos, laboratorios y aulas que permitan el desarrollo de proyectos e implementaciones prácticas.

Otorgar capacitaciones a los estudiantes sobre el manejo, creación de software que permitan brindar nuevas soluciones a problemas de aspecto técnico y equipamiento de la carrera de Ingeniería en Teleinformática.

Los estudiantes deben tener conocimientos sólidos sobre los sistemas de modulaciones digitales, con lo cual se puede reducir el tiempo en la ejecución de prácticas.

Se recomienda evaluar a los estudiantes, en el aspecto teórico y práctico, con lo cual se puede determinar el grado de conocimiento de los estudiantes e impulsar de mejor manera su perfil profesional.

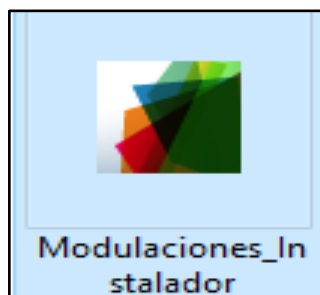
ANEXOS

Anexo 1 Manual de usuario

Manual de usuario de la aplicación del sistema de modulaciones analógicas

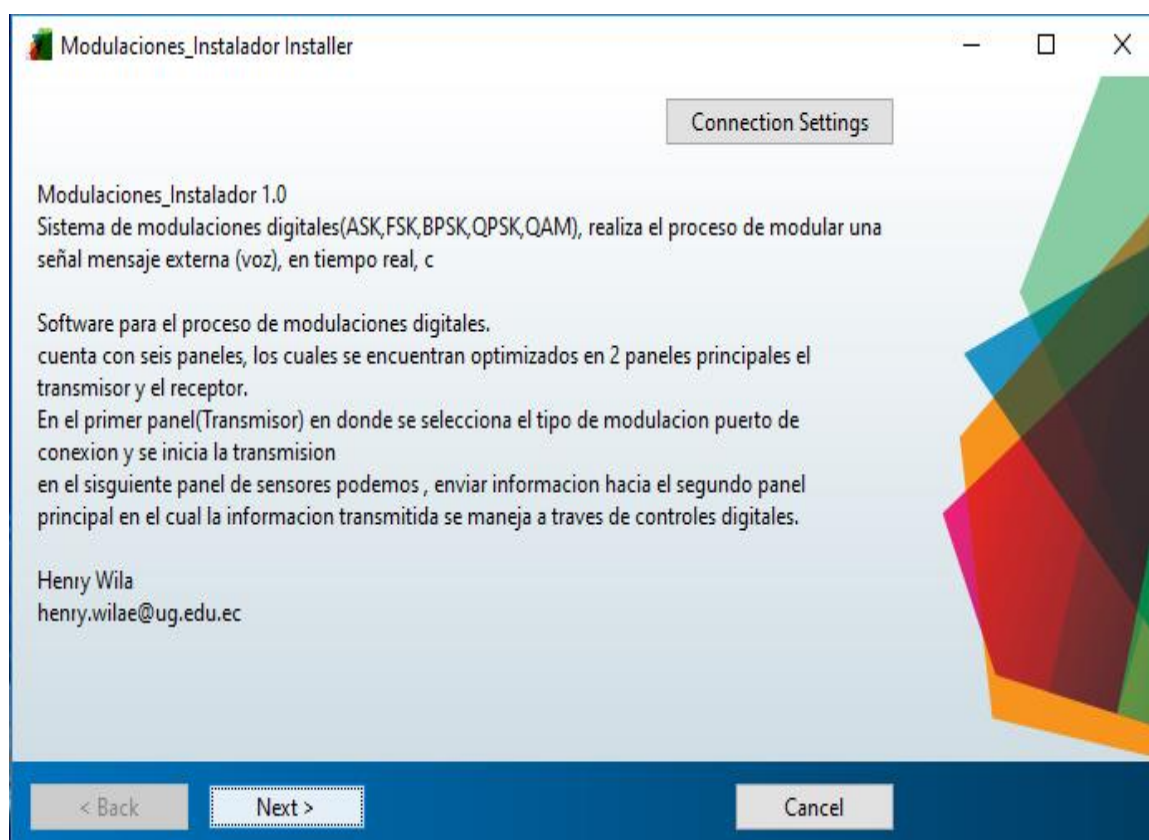
Instalación de la aplicación

Se recomienda 3.5 GB de espacio libre en el disco duro y una memoria RAM de 8 GB en el ordenador donde se va a instalar la aplicación.



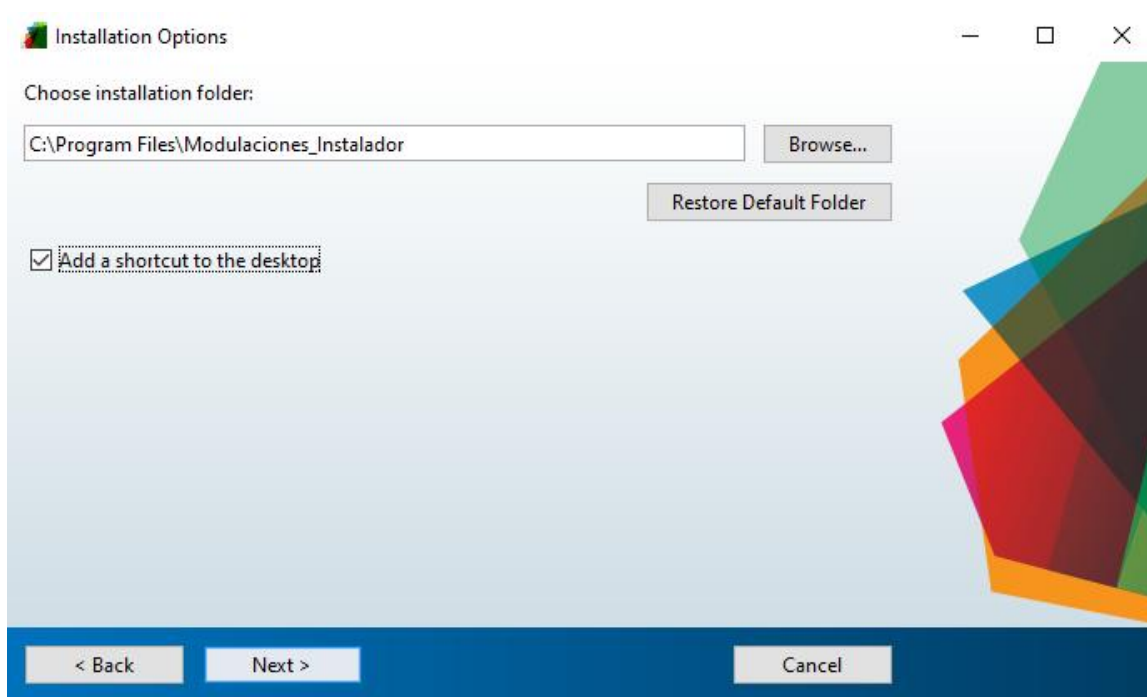
Se debe dar doble clic en el archivo setup de nombre Modulaciones_Instalador para proceder con la instalación de este.

Luego que el archivo empieza a instalarse aparece una ventana con un mensaje que diga si se permite que la aplicación realice cambios en el sistema. Procedemos a dar clic en si.



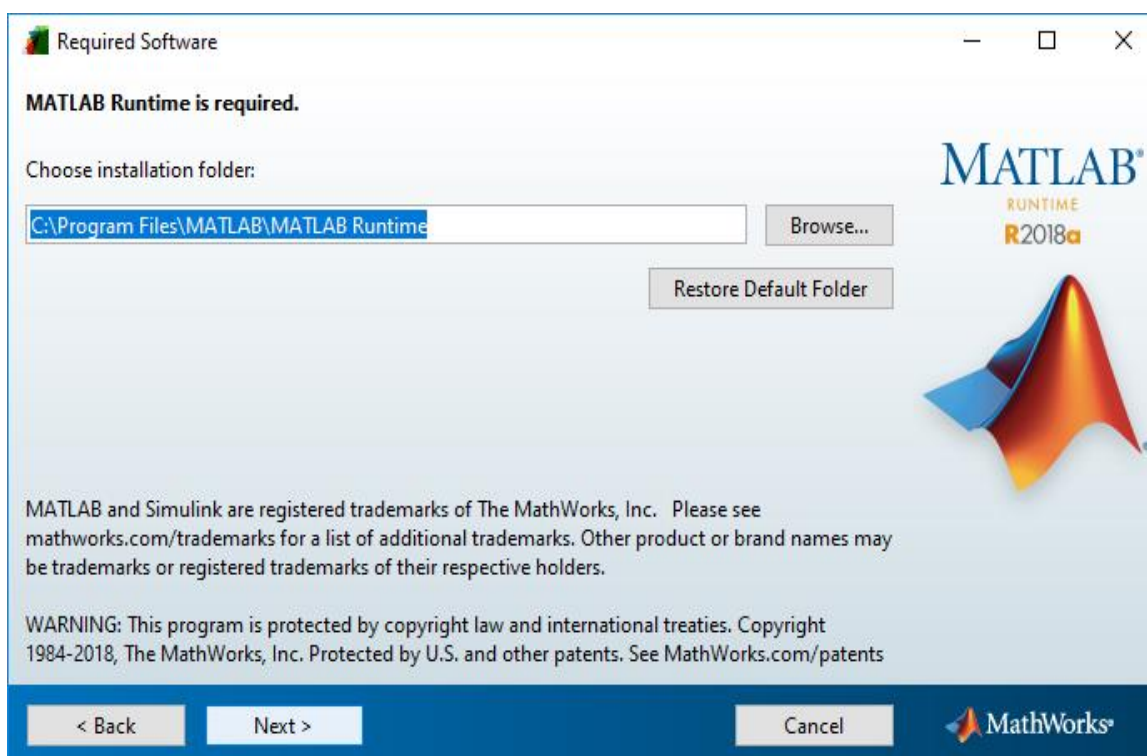
*Ventana de instalación del sistema de modulaciones digitales. Información tomada directamente del autor.
Elaborado por el autor.*

En la siguiente ventana damos clic en el botón Next.



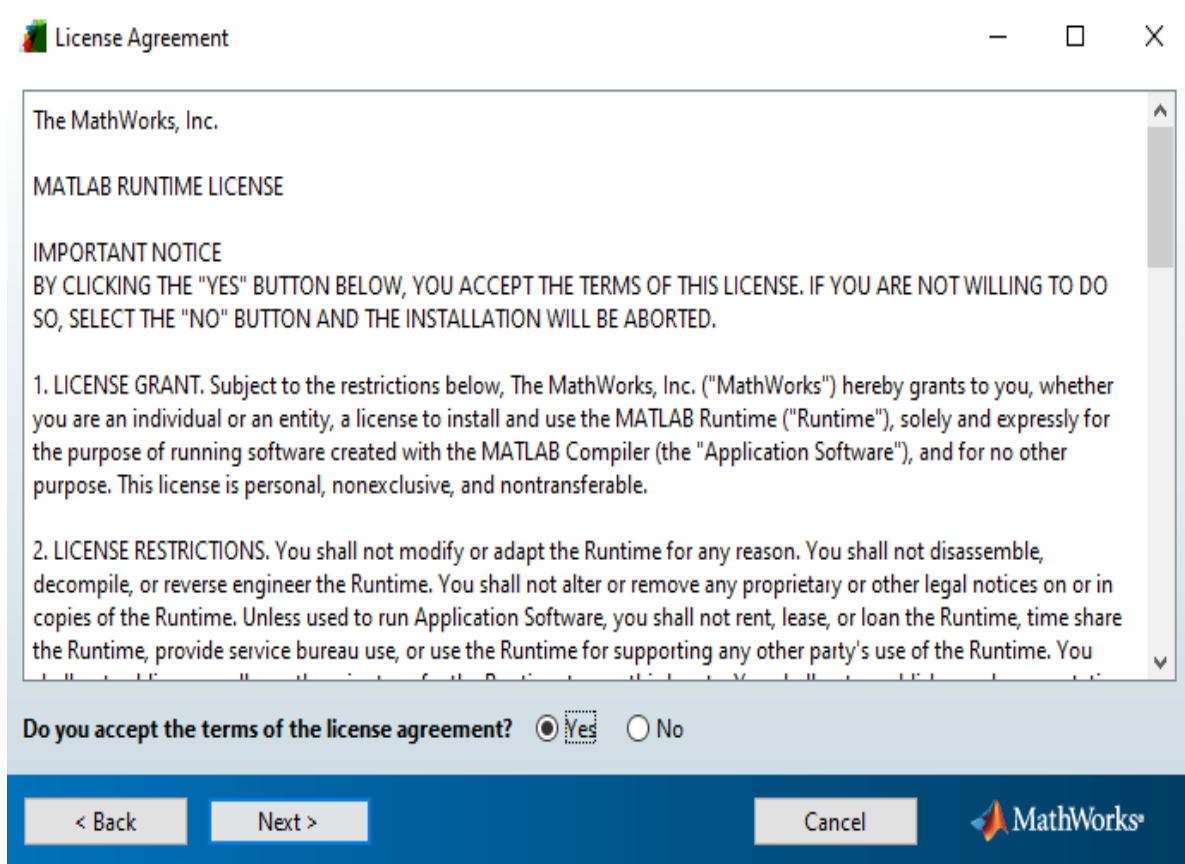
Selección de carpeta para instalación del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

En esta ventana se procede a seleccionar el casillero (1), para crear un icono de acceso directo en el escritorio, luego damos clic en el botón Next (2)

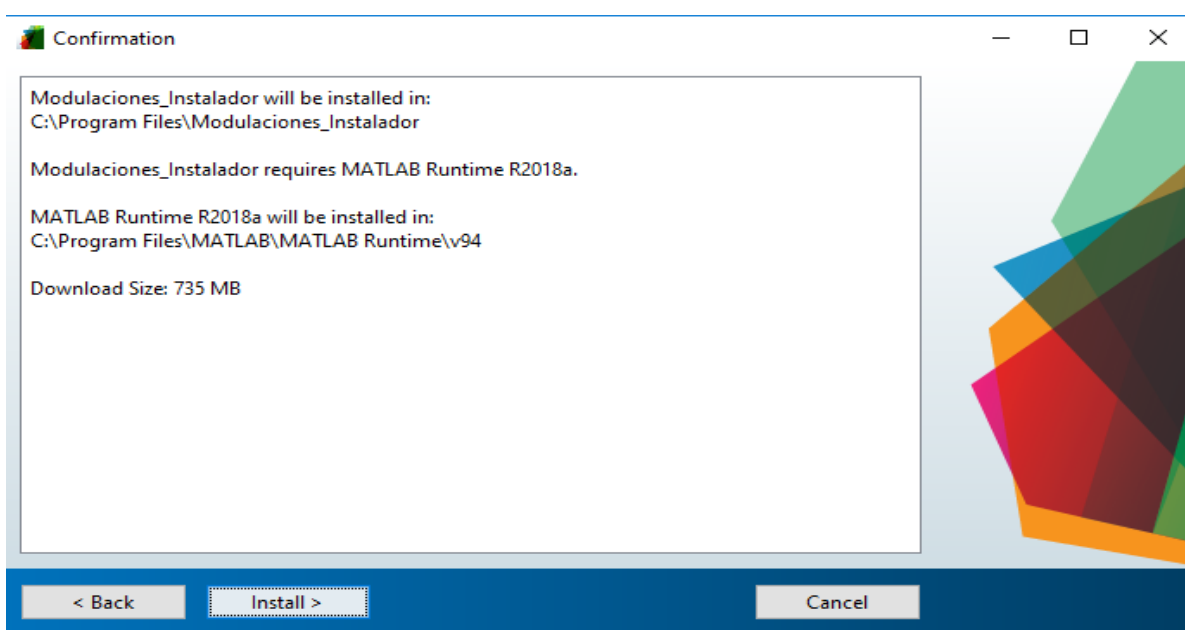


Selección de carpeta para instalación del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

En esta ventana seleccionamos la ubicación para la instalación (1) y luego se da clic en el botón Next (2).



Acuerdo de licencia del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.



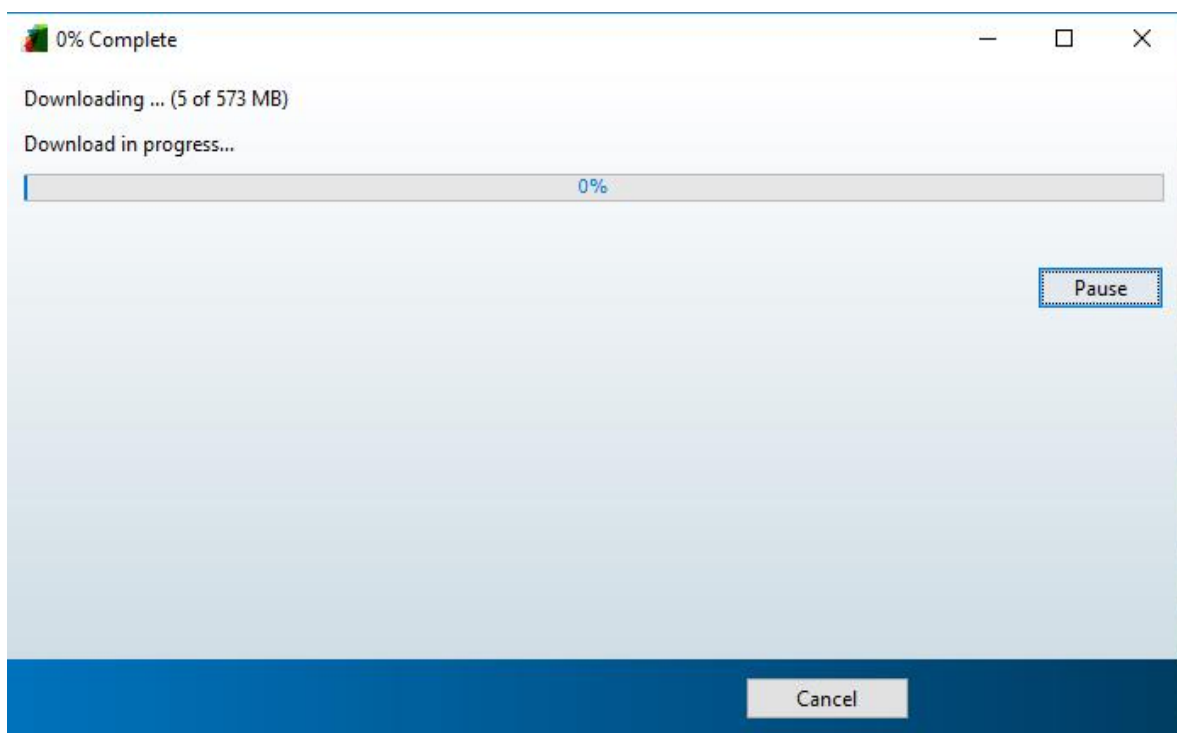
Confirmación de instalación del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

En la siguiente ventana se debe leer y aceptar los acuerdos de la licencia Yes (1), luego

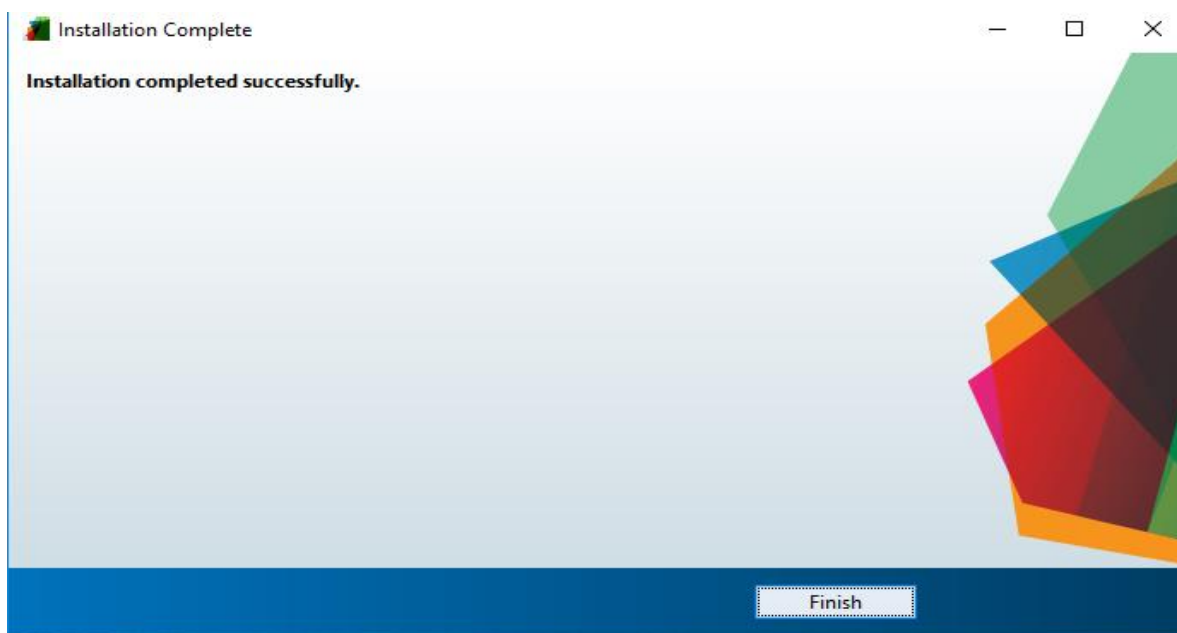
se da clic en el botón Next (2).

Se procede a dar clic en el botón Install.

La instalación de la aplicación puede tomar varios minutos.



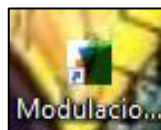
Proceso de instalación del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.



Ventana de finalización de instalación del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

Se debe dar clic en Finish, para finalizar la instalación de la aplicación.

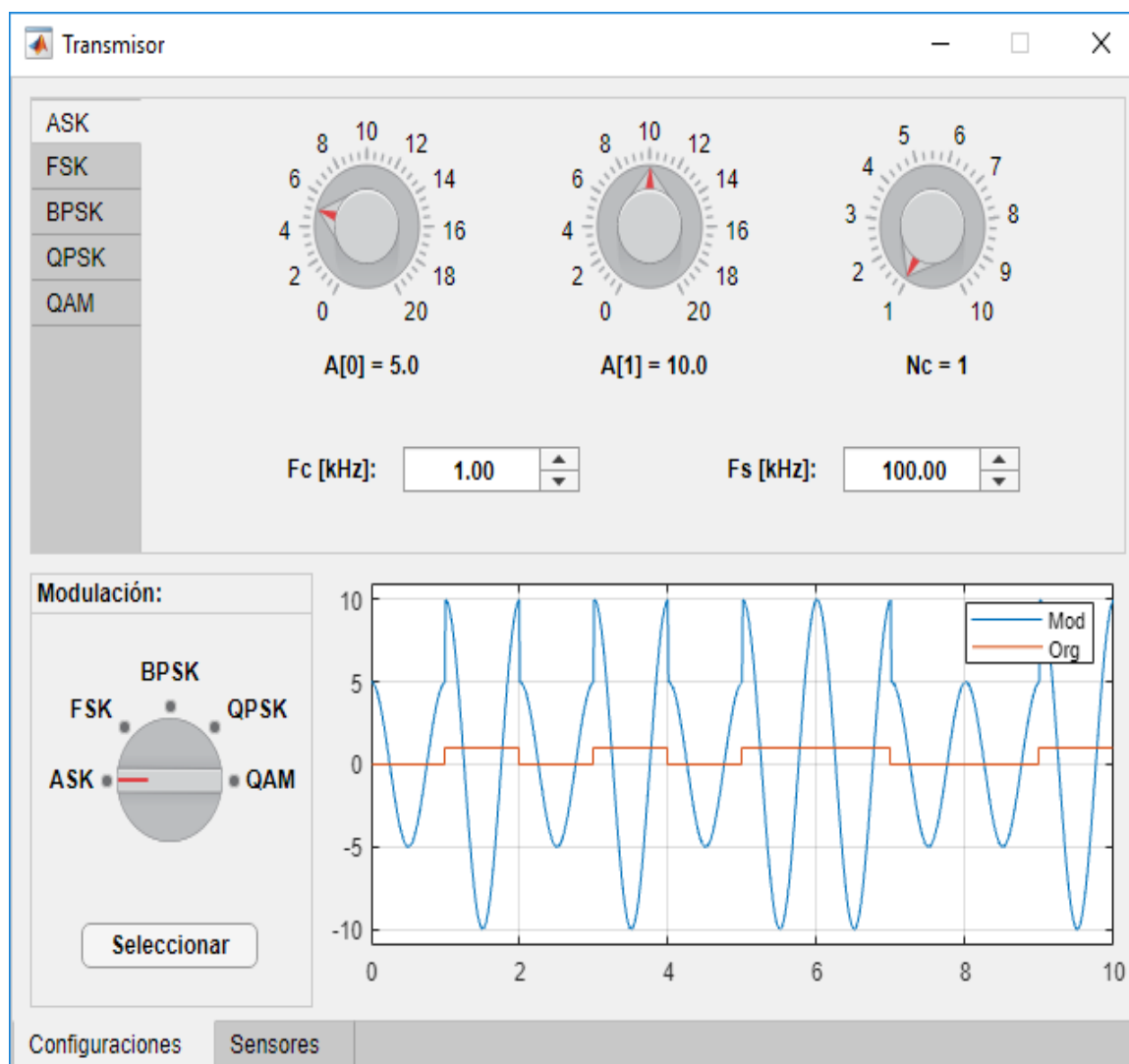
Una vez finalizada la instalación se creará un icono en el escritorio, procedemos a dar doble clic para iniciar la aplicación.



Icono del ejecutable del sistema de modulaciones digitales Información tomada directamente del autor. Elaborado por el autor.

Funcionamiento de la aplicación del sistema de modulaciones digitales.

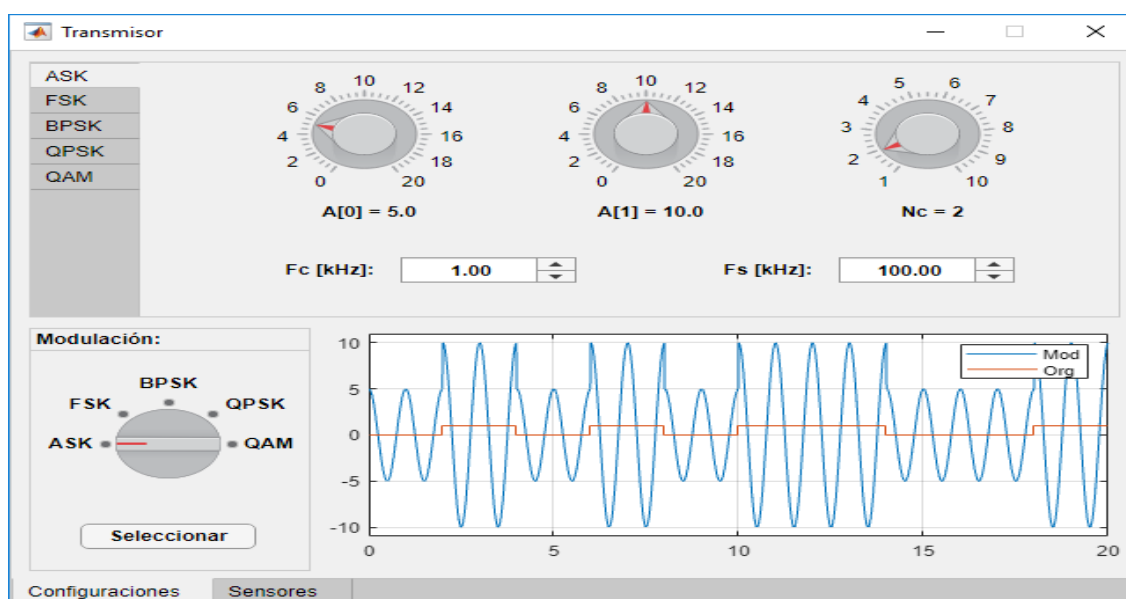
Modulaciones/Panel Transmisor



Panel Transmisor del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

En este panel se captura la información que se va a transmitir, en este panel también se selecciona los valores mediante los controles.

Modulacion ASK.

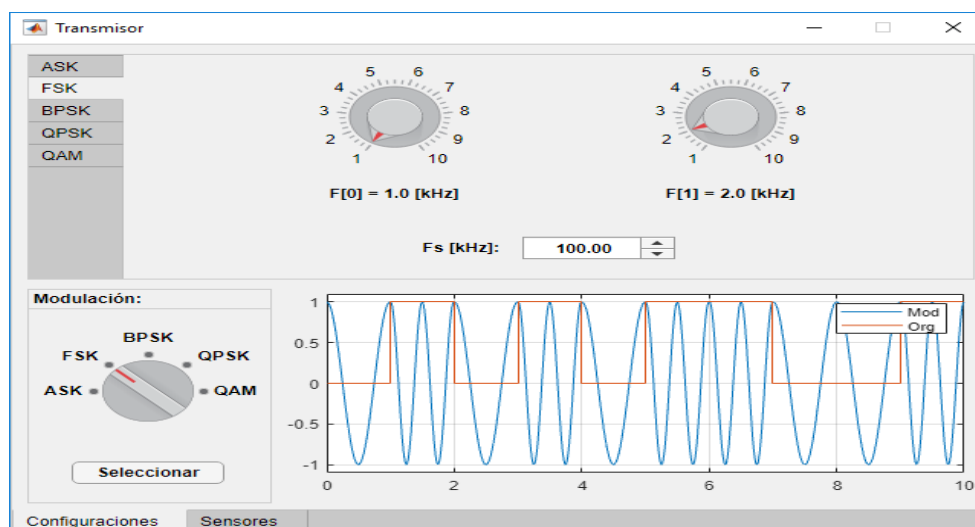


Modulacion ASK del sistema de modulaciones digitales. Información tomada directamente del autor.
Elaborado por el autor.

Donde:

- **A [0]:** Amplitud cuando se transmite un bit igual a cero (0).
- **A [1]:** Amplitud cuando se transmite un bit igual a uno (1).
- **Nc:** Numero de ciclos por cada bit a transmitir.
- **Fc:** Frecuencia de la portadora en kilo Hertz.
- **Fs:** Frecuencia de muestreo en kilo Hertz.

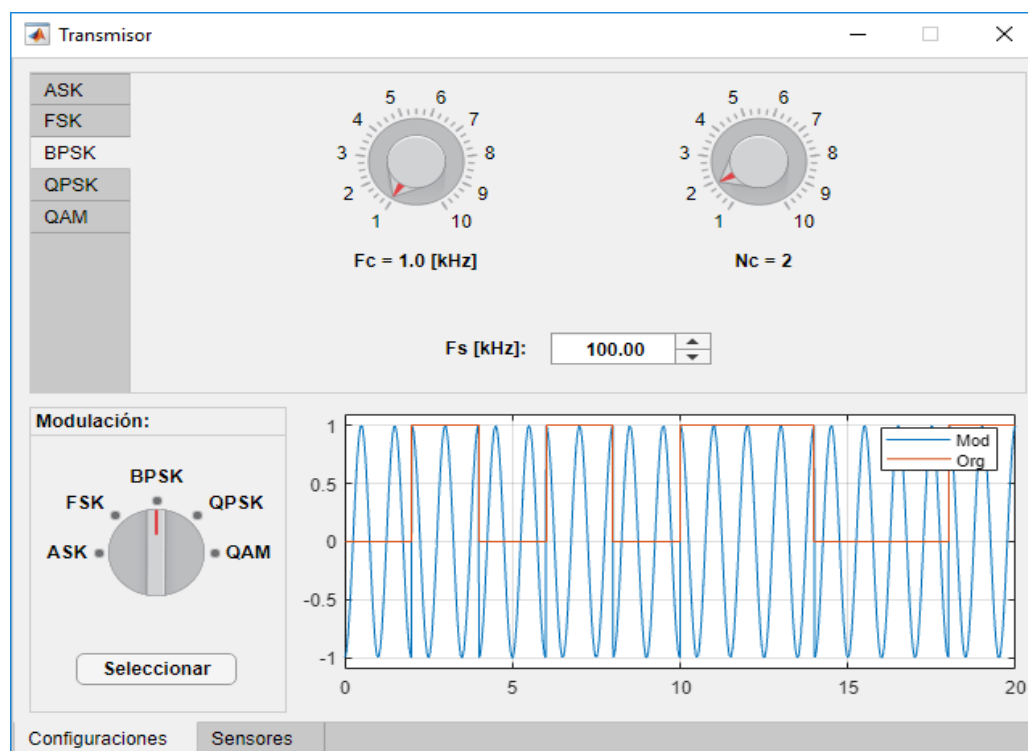
Modulacion FSK/ Panel Transmisor



Modulación FSK del sistema de modulaciones digitales. Información tomada directamente del autor.
Elaborado por el autor.

Donde:

- **F [0]:** Frecuencia cuando se transmite un bit igual a cero (0) en kilo Hertz.
- **F [1]:** Frecuencia cuando se transmite un bit igual a uno (1) en kilo Hertz.
- **Fs:** Frecuencia de muestreo en kilo Hertz

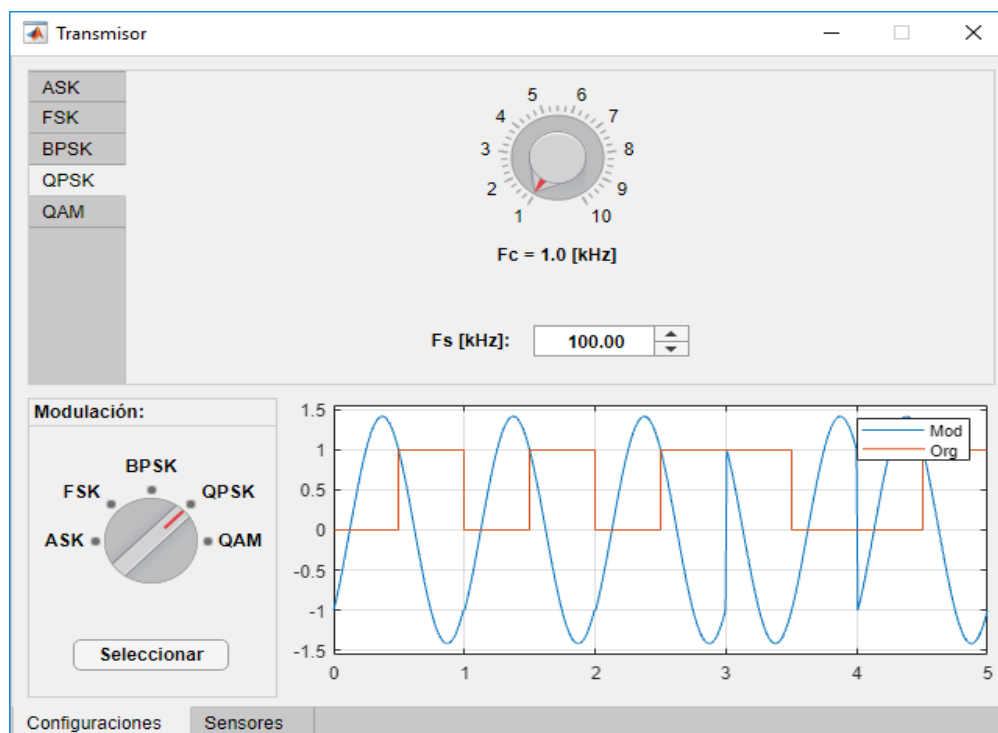
Modulador BPSK/Transmisor

*Modulación BPSK del sistema de modulaciones digitales. Información tomada directamente del autor.
Elaborado por el autor.*

Donde:

- **Nc** Numero de ciclos por cada bit a transmitir.
- **Fc** Frecuencia de la portadora en kilo Hertz.
- **Fs** Frecuencia de muestreo en kilo Hertz.

Modulación QPSK

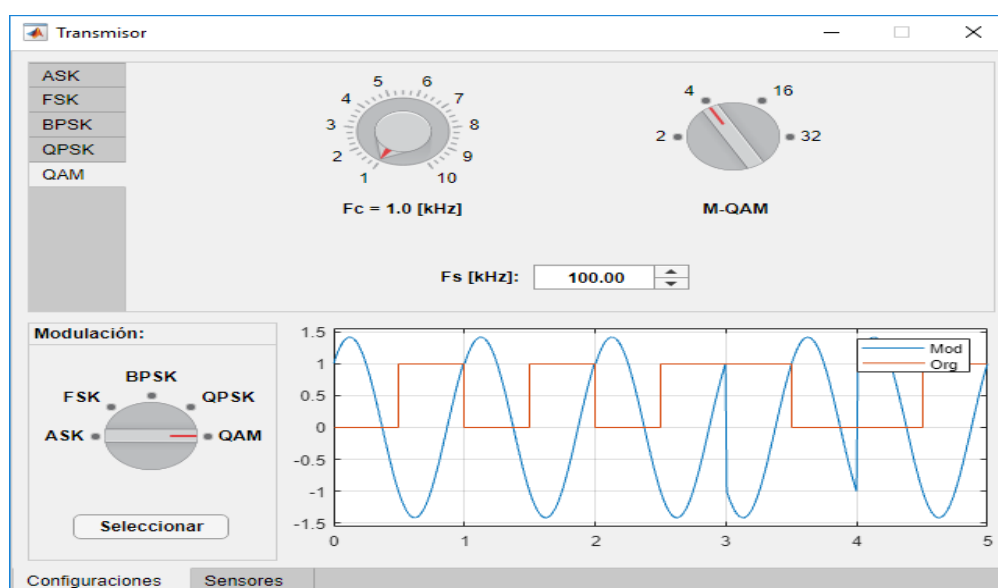


*Modulación QPSK del sistema de modulaciones digitales. Información tomada directamente del autor.
Elaborado por el autor.*

Donde:

- F_c Frecuencia de la portadora en kilo Hertz.
- F_s Frecuencia de muestreo en kilo Hertz.

Modulación QAM



*Modulación QAM del sistema de modulaciones digitales. Información tomada directamente del autor.
Elaborado por el autor.*

Donde:

- M Tipo de modulación QAM.
- Fc Frecuencia de la portadora en kilo Hertz.
- Fs Frecuencia de muestreo en kilo Hertz.

Configuración Cliente TCP/IP

CLIENTE TCP/IP:

IP: Puerto: Tiempo:

Configuración del Cliente. Información tomada directamente del autor. Elaborado por el autor.

Donde:

- IP: Dirección IP del servidor (localhost solo si está en el mismo pc).
- Puerto Puerto usado por el servidor.
- Tiempo Tiempo de refresco para mandar los datos.

Configuración Servidor TCPIP

SERVIDOR TCP/IP:

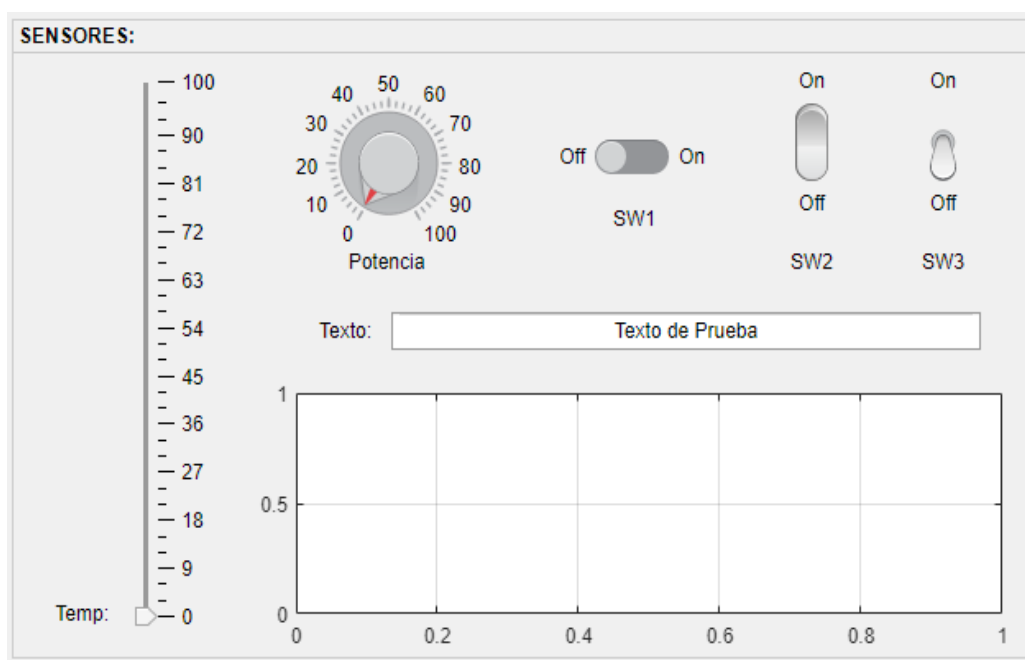
IP: Puerto: Tiempo:

Configuración del Servidor. Información tomada directamente del autor. Elaborado por el autor.

Donde:

- IP: Dirección IP del servidor (0.0.0.0 solo si es la dirección del pc).
- Puerto Puerto usado por el servidor.
- Tiempo Tiempo de refresco para mandar los datos.

Sensores

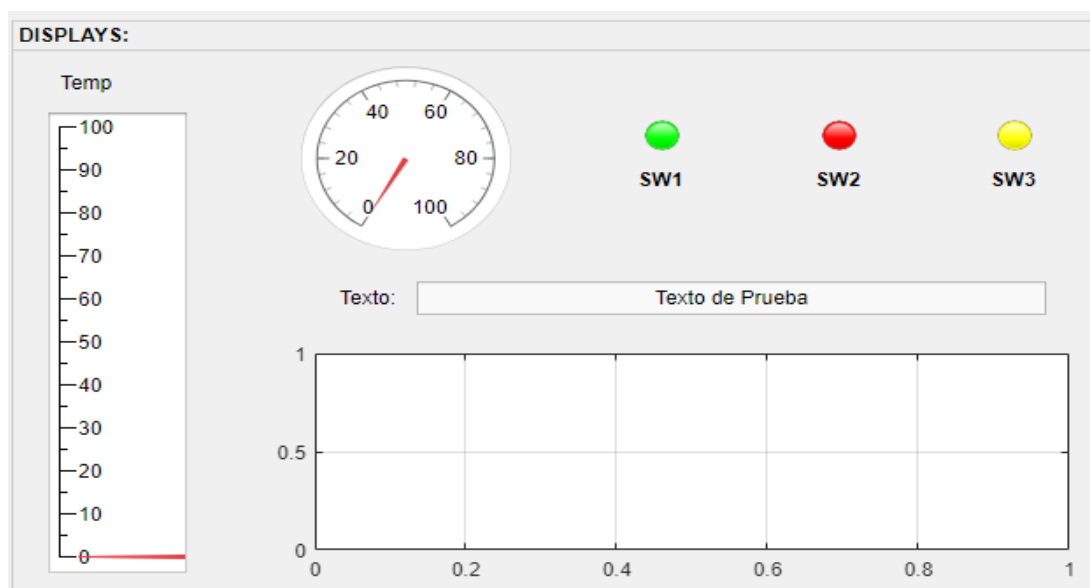


*Panel de Sensores del sistema de modulaciones digitales. Información tomada directamente del autor.
Elaborado por el autor.*

Donde:

- Temp: Sensor de Temperatura.
- Potencia: Potencia usada.
- SW1: Switch 1
- SW2: Switch 2
- SW3: Switch 3
- Texto: Texto a enviar.
- Eje: Audio ingresado.

Displays



Panel de Displays del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

Donde:

- Temp: Temperatura recibida.
- Potencia: Potencia recibida.
- SW1: Switch 1
- SW2: Switch 2
- SW3: Switch 3
- Texto: Texto recibido.
- Eje: Audio demodulado.

Selección de Modulación Para Usar

Antes de empezar la transmisión y/o recepción, se debe escoger la modulación a usar mediante el selector de modulación y presionando el botón de “Seleccionar”.



Selector de tipo de modulación del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

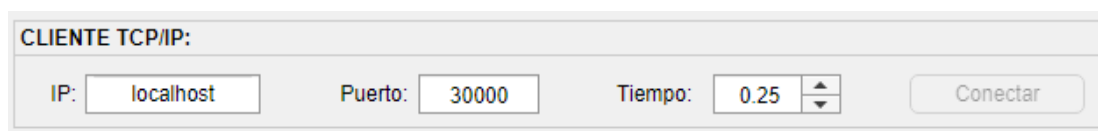
Una vez seleccionado el tipo de modulación, esta no podrá ser cambiada al igual que los parámetros que configuran la misma.



Panel de tipo de modulación del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

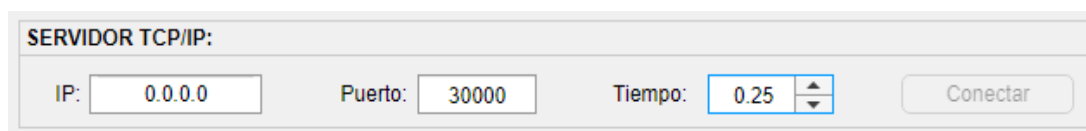
Conexión TCPIP

Para poder iniciar la comunicación TCPIP primero se debe activar el servidor desde el receptor. Para ello se debe ingresar la IP del equipo (usualmente 0.0.0.0), el puerto y el tiempo de refresco. Cabe aclarar que, si no se ha seleccionado un tipo de modulación, no se podrá habilitar la conexión.



Conexiones del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

Posteriormente, se procede a realizar la conexión desde el lado del cliente de la misma manera:



Conexiones del servidor del sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

Si se han configurado correctamente, la transmisión deberá empezar inmediatamente.

Desconexión

Para realizar la desconexión sin ningún inconveniente, se debe primero desconectar el

cliente y luego el servidor.

Solo cuando los programas se encuentren desconectados, se puede cambiar los parámetros de la modulación.

NOTAS

- Elevar la frecuencia de muestreo demasiado puede resultar en retardos en la transmisión/recepción.
- Si la frecuencia de muestreo es muy baja, se puede producir errores en la demodulación, por lo que se recomienda que la frecuencia mínima de muestreo sea por lo menos 10 veces superior a la frecuencia de la portadora.
- Si el transmisor y el receptor no se encuentran configurados, puede resultar desde una incorrecta demodulación o incluso que el programa muestre errores o se congele.
- Dependiendo del tipo de modulación, puede ocurrir mayores o menores cortes en la transmisión de los datos, ya que hay modulaciones que necesitan mayor cantidad de muestras (fsk) y otras que requieren menos (32-QAM).
- Siempre conectar primero el servidor y posteriormente el cliente. De igual manera, se debe desconectar primero el cliente y posteriormente el servidor para evitar errores.

Anexo 2

Preguntas post-encuesta

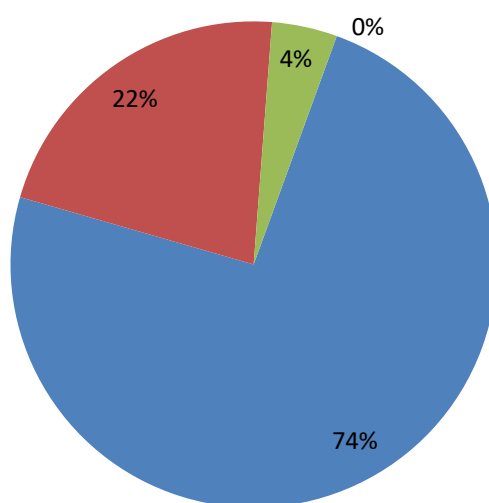
1. ¿Las prácticas dentro del laboratorio sirven de apoyo para el proceso de enseñanza de los estudiantes?

Practicas laboratorio como apoyo en proceso de enseñanza

Descripción	Frecuencia	%
Totalmente de acuerdo	17	74
De acuerdo	5	22
En desacuerdo	1	4
Totalmente en desacuerdo	0	0
Total	23	100

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor

■ Totalmente de acuerdo ■ De acuerdo ■ En desacuerdo ■ Totalmente en desacuerdo



Distribución porcentual de las opiniones de los estudiantes en relación con las prácticas laboratorio como apoyo en proceso de enseñanza. Información tomada directamente del autor. Elaborado por el autor.

El 74% de los encuestados indica que las prácticas dentro del laboratorio sirven de apoyo para el proceso de enseñanza de los estudiantes, mientras que el 22% está de acuerdo y solo el 4% no está de acuerdo con la contribución que pueden dar las prácticas de laboratorio en el proceso enseñanza aprendizaje.

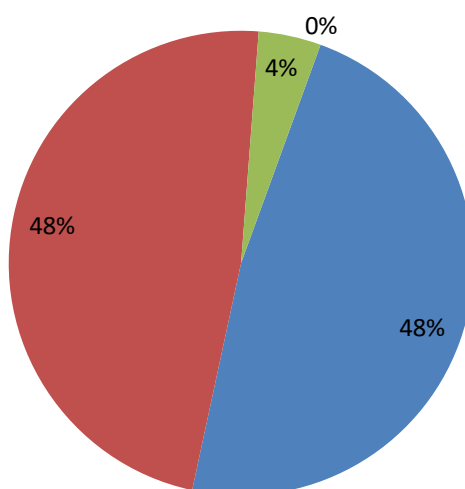
2. ¿Las prácticas dentro del laboratorio, permiten reducir tiempo, costos, tanto a los estudiantes como a docentes?

Beneficios dentro de las prácticas de laboratorio

Descripción	Frecuencia	%
Totalmente de acuerdo	11	48
De acuerdo	11	48
En desacuerdo	1	4
Totalmente en desacuerdo	0	0
Total	23	100

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor

■ Totalmente de acuerdo ■ De acuerdo ■ En desacuerdo ■ Totalmente en desacuerdo



Distribución porcentual de las opiniones de los estudiantes en relación con los beneficios dentro de las prácticas de laboratorio. Información tomada directamente del autor. Elaborado por el autor.

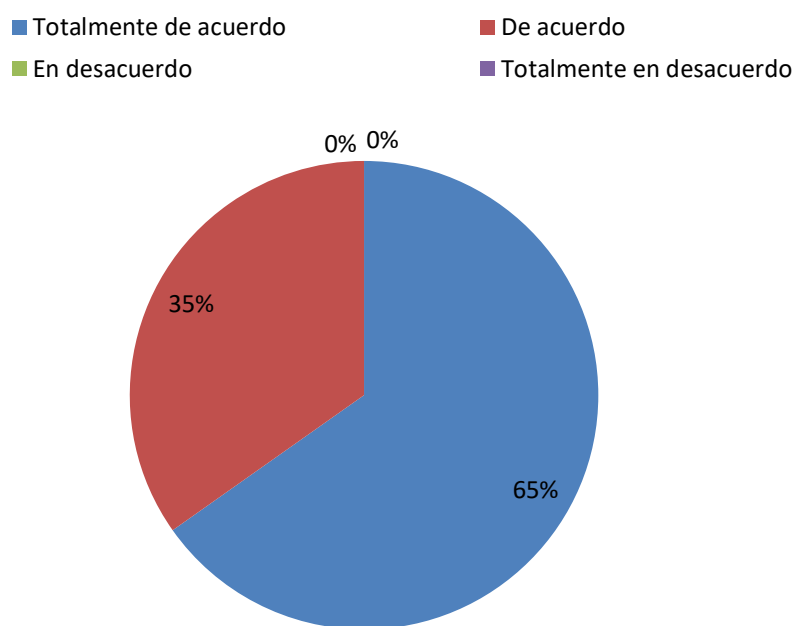
El 48% de los encuestados está totalmente de acuerdo que las prácticas dentro del laboratorio permiten reducir tiempo, costos, tanto a los estudiantes como a docentes, otro 48% está de acuerdo con esta afirmación y el 4% está en desacuerdo.

3. ¿Existe necesidad de que los estudiantes dominen el uso de software para la elaboración de prácticas?

Encuesta para el dominio de software

Descripción	Frecuencia	%
Totalmente de acuerdo	15	65
De acuerdo	8	35
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	23	100

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor



Distribución porcentual de las opiniones de los estudiantes con respecto del uso del software para las practicas. Información tomada directamente del autor. Elaborado por el autor.

El 65% de los encuestados está totalmente de acuerdo que existe necesidad de que los estudiantes dominen el uso de software para la elaboración de prácticas y el 35% está de acuerdo puesto que reducen el tiempo y los costos asociados a la utilización de diferentes equipos sin embargo permiten que el estudiante aplique los conocimientos teóricos aprendidos en el aula de clase.

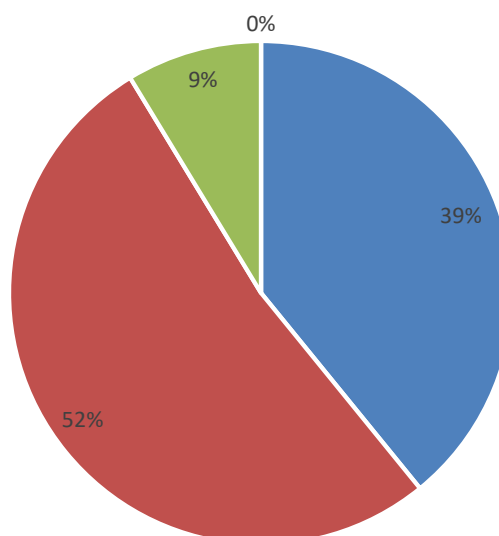
4. ¿La utilización de MATLAB es sencillo y más práctico que el uso de otro hardware?

Comparaciones entre MATLAB y otros hardware

Distribución	Frecuencia	%
Totalmente de acuerdo	9	39
De acuerdo	12	52
En desacuerdo	2	9
Totalmente en desacuerdo	0	0
Total	23	100

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor

■ Totalmente de acuerdo ■ De acuerdo ■ En desacuerdo ■ Totalmente en desacuerdo



Distribución porcentual de las opiniones de los estudiantes en relación comparaciones entre MATLAB y otros hardware. Información tomada directamente del autor. Elaborado por el autor

El 39% de los encuestados indica que la utilización de MATLAB es sencillo y más práctico que el uso de otro hardware, mientras que el 52% está de acuerdo. Existiendo de esta manera un alto nivel de aceptación por los estudiantes del uso de este sistema puesto que finalmente solo el 9% está en desacuerdo.

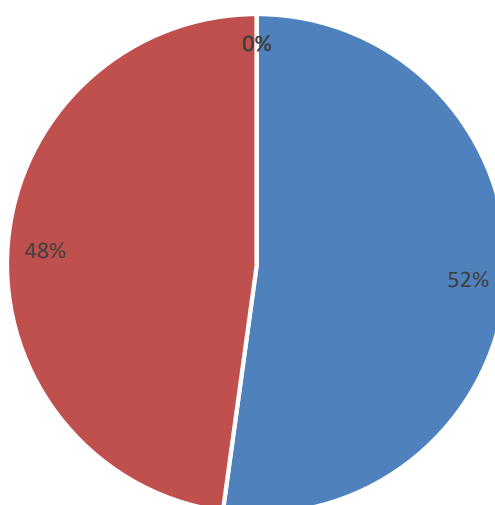
5. ¿El uso de la herramienta App Designer del software MATLAB, incluye componentes necesarios para el desarrollo de un sistema de modulaciones digitales?

Información tomada de la encuesta de la investigación, 2019. Elaborado por el autor

Descripción	Frecuencia	%
Totalmente de acuerdo	12	52
De acuerdo	11	48
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	23	100

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor

■ Totalmente de acuerdo ■ De acuerdo ■ En desacuerdo ■ Totalmente en desacuerdo



Distribución porcentual de las opiniones de los estudiantes con respecto a herramientas necesarias para el sistema de modulaciones digitales. Información tomada directamente del autor. Elaborado por el autor.

El 52% de los estudiantes cree que El uso de la herramienta App Designer del software MATLAB, incluye componentes necesarios para el desarrollo de un sistema de modulaciones digitales por completo, teniendo una respuesta positiva del 48% de los estudiantes que están de acuerdo de la utilidad y aplicabilidad de uso de este programa en las prácticas de modulaciones digitales.

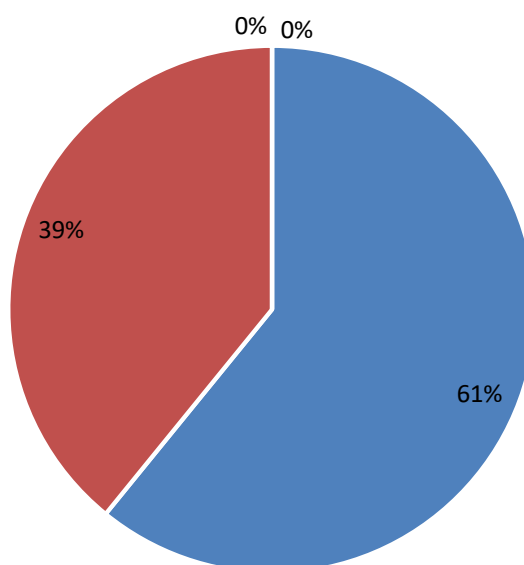
6. ¿Posee conocimientos necesarios para utilizar la aplicación de los sistemas de modulaciones digitales realizadas en la herramienta App Designer del Software Matlab?

Capacidad de aprendizaje para sistemas de modulación digital

	Frecuencia	%
Totalmente de acuerdo	14	61
De acuerdo	9	39
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	23	100

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor

■ Totalmente de acuerdo ■ De acuerdo
■ En desacuerdo ■ Totalmente en desacuerdo



Distribución porcentual de las opiniones de los estudiantes en relación con la capacidad de aprendizaje para sistemas de modulación digital. Información tomada directamente del autor. Elaborado por el autor.

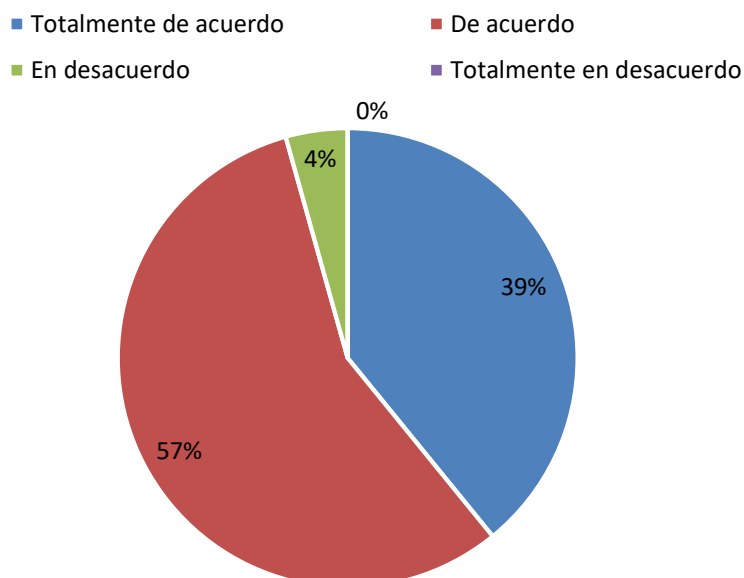
El 61% de los encuestados expresa que posee conocimientos necesarios para utilizar la aplicación de los sistemas de modulaciones digitales realizadas en la herramienta App Designer del Software Matlab totalmente, mientras que el restante 39% está de acuerdo, evidenciando que los estudiantes podrán utilizar adecuadamente este software para sus prácticas de modulaciones digitales.

7. ¿Cree que la aplicación desarrollada en la herramienta App Designer del software MATLAB, es de fácil uso y permite comprender los fundamentos teóricos?

Aprendizaje sencillo sobre fundamentos teóricos

Distribución	Frecuencia	%
Totalmente de acuerdo	9	39
De acuerdo	13	57
En desacuerdo	1	4
Totalmente en desacuerdo	0	0
Total	23	100

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor



Distribución porcentual de las opiniones de los estudiantes en relación al aprendizaje sencillo sobre fundamentos teóricos, Información tomada directamente del autor. Elaborado por el autor.

El 39% de los estudiantes está totalmente de acuerdo que la aplicación desarrollada en la herramienta App Designer del software MATLAB, es de fácil manejo y permite comprender los fundamentos teóricos, el 57% está de acuerdo cree de la misma forma que el manejo del software es fácil y contribuye con aprendizaje practico, el 4% está en desacuerdo puesto que indica que al usar este sistema se obvian ciertos aspectos.

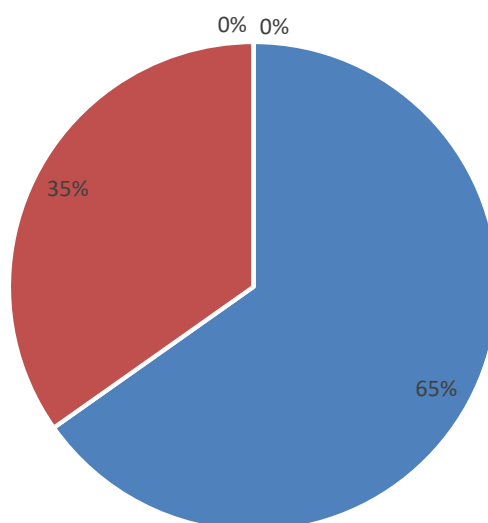
8. ¿Considera que la implementación de los sistemas de modulaciones digitales en la herramienta App Designer del software MATLAB, optimizo el tiempo para la adquisición de conocimiento de las clases?

Tiempo de captación y aprendizaje

Distribución	Frecuencia	%
Totalmente de acuerdo	15	65
De acuerdo	8	35
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	23	100

Información tomada de la encuesta de la investigación, 2019. Elaborado por el autor

■ Totalmente de acuerdo ■ De acuerdo



Distribución porcentual de las opiniones de los estudiantes en relación con el tiempo de captación y aprendizaje. Información tomada directamente del autor. Elaborado por el autor.

El 65% de los estudiantes y el 35% consideran que dentro de la implementación de los sistemas de modulaciones digitales en la herramienta App Designer del software MATLAB, esto redujo el tiempo para la adquisición de conocimiento de las clases, debido a que pudieron aplicar conocimientos teóricos adquiridos en las aulas de clase.

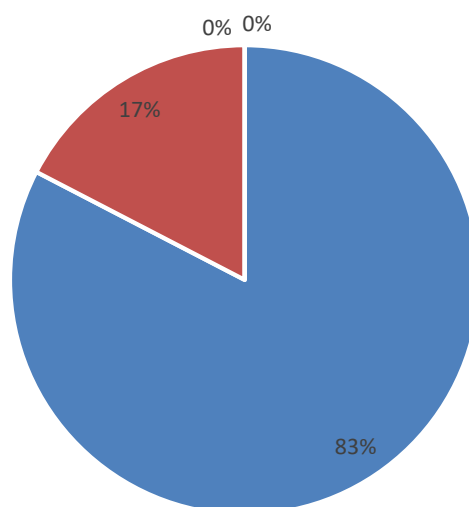
9. ¿Cree usted necesario la implementación de nuevos programas para realizar sistemas que utilicen señales digitales?

Programas para utilizarse en la digitalización

Descripción	Frecuencia	%
Totalmente de acuerdo	19	83
De acuerdo	4	17
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	23	100

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor

■ Totalmente de acuerdo ■ De acuerdo ■ En desacuerdo ■ Totalmente en desacuerdo



Distribución porcentual de las opiniones de los estudiantes en relación con los programas para utilizarse en la digitalización. Información tomada directamente del autor. Elaborado por el autor.

El 83% de los estudiantes está de acuerdo con esta iniciativa de usar sistemas diseñados en la herramienta App Designer del Software Matlab para realizar prácticas en laboratorio con los estudiantes de 8vo semestre de la carrera de Ingeniería en Teleinformática, mientras que el 17% está de acuerdo.

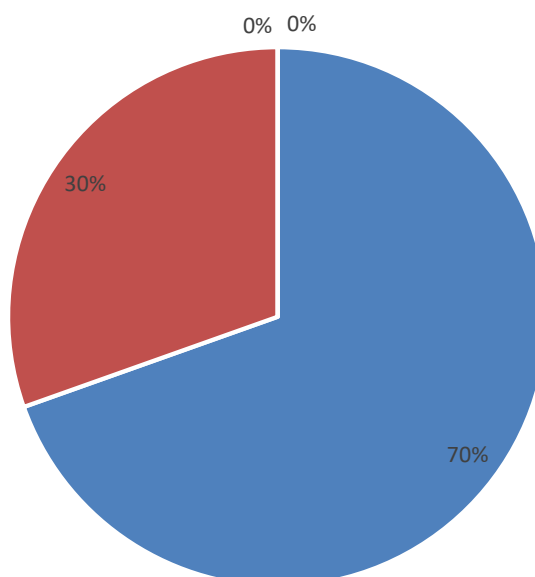
10. ¿El balance entre clases prácticas y teóricas son las correctas para que los estudiantes logren adquirir nuevos conocimientos dentro de la asignatura?

Adquisición de conocimientos en la materia

Distribución	Frecuencia	%
Totalmente de acuerdo	16	70
De acuerdo	7	30
En desacuerdo	0	0
Totalmente en desacuerdo	0	0
Total	23	100

Información adaptada de la encuesta de la investigación, 2019. Elaborado por el autor

■ Totalmente de acuerdo ■ De acuerdo ■ En desacuerdo ■ Totalmente en desacuerdo



Distribución porcentual de las opiniones de los estudiantes en relación con la adquisición de conocimientos en la materia. Información tomada directamente del autor. Elaborado por el autor.

El 70% de los estudiantes está totalmente de acuerdo que la proporción entre clases prácticas y teóricas son las correctas para que los estudiantes puedan adquirir nuevos conocimientos dentro de la catedra, mientras que el 30% también se encuentra de acuerdo.

Anexo 3

Código de la aplicación del sistema de modulaciones digitales

```

%% ASK
function [ outData ] = askModDemod( mode, data, Fc, Fs, Nc, Ac )

    if nargin == 6
        if strcmpi( mode, 'mod' ) || strcmpi( mode, 'modulate' ) ||
strcmpi( mode, 'm' ) % Modular
            [ outData ] = askMod( data, Fc, Fs, Nc, Ac );
        elseif strcmpi( mode, 'dem' ) || strcmpi( mode, 'demodulate' ) ||
strcmpi( mode, 'd' ) % Demodular
            [ outData ] = askDemod( data, Fc, Fs, Nc, Ac );
        else
            [ outData ] = initASK( Fc, Fs, Nc, Ac );
        end
    else
        [ outData ] = initASK( 1e2, 1e4, 1, [5, 10] );
    end

end

%% INIT ASK
function [ obj ] = initASK( Fc, Fs, Nc, Ac )
    obj = struct( 'Fc',Fc, 'Fs',Fs, 'Nc',Nc, 'Ac',Ac );
    obj.mod      = @(bits)      askMod( bits, Fc, Fs, Nc, Ac );
    obj.dem      = @(rxSignal)   askDemod( rxSignal, Fc, Fs, Nc, Ac );
    [ carrier, ~, dt ] = askCarrier( Fc, Fs, Nc );
    obj.carrier = carrier;
    obj.dt = dt;
end

%% MODULACION ASK
function [ txSignal ] = askMod( bits, Fc, Fs, Nc, Ac )

    nBits = length(bits);
    [ carrier, cSize, ~ ] = askCarrier( Fc, Fs, Nc );
    N = cSize * nBits;

    txSignal = zeros( N, 1 );
    for i = 1:nBits
        index = (i-1)*cSize+1 : i*cSize;
        txSignal( index ) = Ac( bits(i) + 1 ) * carrier;
    end

end

%% DEMODULACION ASK
function [ bits ] = askDemod( rxSignal, Fc, Fs, Nc, Ac )

    As = mean(Ac);
    N = length( rxSignal );
    [ carrier, cSize, dt ] = askCarrier( Fc, Fs, Nc );
    nBits = N / cSize;

    bits = zeros( 1, nBits );

```

```

for i = 1:nBits
    index = (i-1)*cSize+1 : i*cSize;
    dSignal = rxSignal( index ) .* carrier;
    z = 2*trapz( dt, dSignal )/dt(end);
    bits(i) = z > As;
end

end

%% GENERACION DE LA PORTADORA ASK
function [ carrier, cSize, dt ] = askCarrier( Fc, Fs, Nc )

    dt = linspace( 0, Nc/Fc - 1/Fs, Fs/Fc*Nc )';
    carrier = cos( 2*pi*Fc*dt );
    cSize = length( carrier );

end
-----
--
%% BPSK
function [ outData ] = bpskModDemod( mode, data, Fc, Fs, Ac, Nc )

    if nargin == 6
        if strcmpi( mode, 'mod' ) || strcmpi( mode, 'modulate' ) ||
strcmpi( mode, 'm' ) % Modular
            [ outData ] = bpskMod( data, Fc, Fs, Ac, Nc );
        elseif strcmpi( mode, 'dem' ) || strcmpi( mode, 'demodulate' ) ||
strcmpi( mode, 'd' ) % Demodular
            [ outData ] = bpskDemod( data, Fc, Fs, Ac, Nc );
        else
            [ outData ] = initBPSK( Fc, Fs, Ac, Nc );
        end
    else
        [ outData ] = initBPSK( 1e2, 1e4, 1, 1 );
    end

end

end

%% INIT BPSK
function [ obj ] = initBPSK( Fc, Fs, Ac, Nc )
    obj = struct( 'Fc',Fc, 'Fs',Fs, 'Ac',Ac, 'Nc',Nc );
    obj.mod      = @(bits)      bpskMod( bits, Fc, Fs, Ac, Nc );
    obj.dem      = @(rxSignal)  bpskDemod( rxSignal, Fc, Fs, Ac, Nc );
    [ carrier, ~, dt ] = bpskCarrier( Fc, Fs, Ac, Nc );
    obj.carrier = carrier;
    obj.dt = dt;

end

%% MODULACION BPSK
function [ txSignal ] = bpskMod( bits, Fc, Fs, Ac, Nc )

    [ carrier, cSize, ~ ] = bpskCarrier( Fc, Fs, Ac, Nc );
    nBits = length(bits);
    N = cSize * nBits;

    s = double( bits );
    txSignal = zeros( N, 1 );
    for i = 1:nBits
        index = (i-1)*cSize+1 : i*cSize;

```

```

txSignal( index ) = sign( s(i) - 0.5 ) * carrier;
end

end

%% DEMODULACION BPSK
function [ bits ] = bpskDemod( rxSignal, Fc, Fs, Ac, Nc )

    [ carrier, cSize, dt ] = bpskCarrier( Fc, Fs, Ac, Nc );
    N = length( rxSignal );
    nBits = N / cSize;

    bits = zeros( 1, nBits );
    for i = 1:nBits
        index = (i-1)*cSize+1 : i*cSize;
        ds = rxSignal( index ) .* carrier;
        z = 2*trapz( dt, ds )/dt(end);
        bits(i) = z > Ac/2;
    end

end

%% GENERACION DE LA PORTADORA BPSK
function [ carrier, cSize, dt ] = bpskCarrier( Fc, Fs, Ac, Nc )

    dt = linspace( 0, Nc/Fc, Nc*Fs/Fc )';
    carrier = Ac*cos( 2*pi*Fc*dt );
    cSize = length( carrier );

end
-----
--
%% FSK
function [ outData ] = fskModDemod( mode, data, Fc, Fs, Ac )

    if nargin == 5
        if strcmpi( mode, 'mod' ) || strcmpi( mode, 'modulate' ) ||
        strcmpi( mode, 'm' ) % Modular
            [ outData ] = fskMod( data, Fc, Fs, Ac );
        elseif strcmpi( mode, 'dem' ) || strcmpi( mode, 'demodulate' ) ||
        strcmpi( mode, 'd' ) % Demodular
            [ outData ] = fskDemod( data, Fc, Fs, Ac );
        else
            [ outData ] = initFSK( Fc, Fs, Ac );
        end
    else
        [ outData ] = initFSK( [ 1e3, 2e3 ], 1e4, 1 );
    end

end

%% INIT FSK
function [ obj ] = initFSK( Fc, Fs, Ac )
    obj = struct( 'Fc',Fc, 'Fs',Fs, 'Ac',Ac );
    obj.mod      = @(bits)      fskMod( bits, Fc, Fs, Ac );
    obj.dem      = @(rxSignal)  fskDemod( rxSignal, Fc, Fs, Ac );
    [ carrier, ~, dt ] = fskCarrier( Fc, Fs, Ac );
    obj.carrier = carrier;
    obj.dt = dt;

```

end

%% MODULACION FSK

```
function [ txSignal ] = fskMod( bits, Fc, Fs, Ac )

    [ carrier, cSize, ~ ] = fskCarrier( Fc, Fs, Ac );
    nBits = length(bits);
    N = cSize * nBits;

    txSignal = zeros( N, 1 );
    for i = 1:nBits
        index = (i-1)*cSize+1 : i*cSize;
        txSignal( index ) = carrier( :, bits(i) + 1 );
    end
```

end

%% DEMODULACION FSK

```
function [ bits ] = fskDemod( rxSignal, Fc, Fs, Ac )

    [ carrier, cSize, dt ] = fskCarrier( Fc, Fs, Ac );
    N = length( rxSignal );
    nBits = N / cSize;

    bits = zeros( 1, nBits );
    for i = 1:nBits
        index = (i-1)*cSize+1 : i*cSize;

        ds1 = rxSignal( index ) .* carrier( :, 1 );
        ds2 = rxSignal( index ) .* carrier( :, 2 );

        z1 = 2*trapz( dt, ds1 )/dt(end);
        z2 = 2*trapz( dt, ds2 )/dt(end);

        if z1 > Ac/2
            bits(i) = 0;
        elseif z2 > Ac/2
            bits(i) = 1;
        end
    end
end
```

end

%% GENERACION DE LA PORTADORA FSK

```
function [ carrier, cSize, dt ] = fskCarrier( Fc, Fs, Ac )

    Nc = gcd( Fc(1), Fc(2) );
    dt = linspace( 0, 1/Nc, Fs/Nc )';
    c1 = cos( 2*pi*Fc(1)*dt );
    c2 = cos( 2*pi*Fc(2)*dt );
    carrier = Ac * [ c1, c2 ];
    cSize = size( carrier, 1 );
```

end

--

%% QPSK

```
function [ outData ] = qpskModDemod( mode, data, Fc, Fs
```

```

if nargin == 4
    if strcmpi( mode, 'mod' ) || strcmpi( mode, 'modulate' ) ||
strcmpi( mode, 'm' ) % Modular
        [ outData ] = qpskMod( data, Fc, Fs );
    elseif strcmpi( mode, 'dem' ) || strcmpi( mode, 'demodulate' ) ||
strcmpi( mode, 'd' ) % Demodular
        [ outData ] = qpskDemod( data, Fc, Fs );
    else
        [ outData ] = initQPSK( Fc, Fs );
    end
else
    [ outData ] = initQPSK( 1e2, 1e4 );
end

end

%% INIT QPSK
function [ obj ] = initQPSK( Fc, Fs )
    obj = struct( 'Fc',Fc, 'Fs',Fs );
    obj.mod      = @(bits)      qpskMod( bits, Fc, Fs );
    obj.dem      = @(rxSignal)  qpskDemod( rxSignal, Fc, Fs );
    [ phCarrier, qdCarrier, ~, dt ] = qpskCarrier( Fc, Fs );
    obj.carrier = [ phCarrier, qdCarrier ];
    obj.dt = dt;
end

%% MODULACION QPSK
function [ txSignal ] = qpskMod( bits, Fc, Fs )

    [ phCarrier, qdCarrier, cSize, ~ ] = qpskCarrier( Fc, Fs );
    nBits = length(bits);
    N = cSize * nBits / 2;

    s = double( bits );
    phSignal = zeros( N, 1 );
    qdSignal = zeros( N, 1 );
    for i = 1:2:nBits-1
        j = ( i - 1 )/2 + 1;
        index = (j-1)*cSize+1 : j*cSize;
        phSignal( index ) = sign( s( i ) - 0.5 ) * phCarrier; % sei; % al
    en phase
        qdSignal( index ) = sign( s(i+1) - 0.5 ) * qdCarrier; % sei; % al
    en quadratura
    end
    txSignal = phSignal + qdSignal;
end

%% DEMODULACION QPSK
function [ bits ] = qpskDemod( rxSignal, Fc, Fs )

    [ phCarrier, qdCarrier, cSize, dt ] = qpskCarrier( Fc, Fs );
    N = length( rxSignal );
    nBits = 2 * N / cSize;

    bits = zeros( 1, nBits );
    for i = 1:2:nBits-1
        j = ( i - 1 )/2 + 1;
        index = (j-1)*cSize+1 : j*cSize;

```

```

    phds = rxSignal( index ) .* phCarrier;

    qdds = rxSignal( index ) .* qdCarrier;

    phz = 2*trapz( dt, phds )/dt(end);
    qdz = 2*trapz( dt, qdds )/dt(end);

    bits( i ) = phz > 0;
    bits(i+1) = qdz > 0;
end

end

%% GENERACION DE LA PORTADORA QPSK
function [ phCarrier, qdCarrier, cSize, dt ] = qpskCarrier( Fc, Fs )

    dt = linspace( 0, 1/Fc, Fs/Fc )';
    phCarrier = cos( 2*pi*Fc*dt );
    qdCarrier = sin( 2*pi*Fc*dt );
    cSize = length(dt);

end

-----
--
%% QAM
function [ outData ] = qamModDemod( mode, data, Fc, Fs, M )

    if nargin == 5
        if strcmpi( mode, 'mod' ) || strcmpi( mode, 'modulate' ) ||
strcmpi( mode, 'm' ) % Modular
            [ outData ] = qamMod( data, Fc, Fs, M );
        elseif strcmpi( mode, 'dem' ) || strcmpi( mode, 'demodulate' ) ||
strcmpi( mode, 'd' ) % Demodular
            [ outData ] = qamDemod( data, Fc, Fs, M );
        else
            [ outData ] = initQAM( Fc, Fs, M );
        end
    else
        [ outData ] = initQAM( 1e3, 1e4, 4 );
    end

end

end

%% INIT QAM
function [ obj ] = initQAM( Fc, Fs, M )
    obj = struct( 'Fc',Fc, 'Fs',Fs, 'M',M );
    obj.mod      = @(bits)      qamMod( bits, Fc, Fs, M );
    obj.dem      = @(rxSignal)  qamDemod( rxSignal, Fc, Fs, M );
    [ phCarrier, qdCarrier, ~, dt ] = qamCarrier( Fc, Fs );
    obj.carrier = [ phCarrier, qdCarrier ];
    obj.dt = dt;

end

%% MODULACION QAM
function [ txSignal ] = qamMod( bits, Fc, Fs, M )

```

```

nBits = length(bits);
m = log2(M);

rest = mod( nBits, m );
if rest > 0
    bits = [ bits, zeros( 1, m-rest ) ];
    nBits = length(bits);
end

sortBits = reshape( bits, m, nBits/m )';
symbols = bi2de( sortBits );
constellation = qammod( symbols, M );
phConst = real( constellation );
qdConst = imag( constellation );

[ phCarrier, qdCarrier, cSize, ~ ] = qamCarrier( Fc, Fs );
nSimbols = length(phConst);
N = cSize * nSimbols;

phSignal = zeros( N, 1 );
qdSignal = zeros( N, 1 );
for i = 1:nSimbols
    index = (i-1)*cSize+1 : i*cSize;
    phSignal(index) = phConst(i) * phCarrier;
    qdSignal(index) = qdConst(i) * qdCarrier;
end
txSignal = phSignal + qdSignal;
end

%% DEMODULACION QAM
function [ bits ] = qamDemod( rxSignal, Fc, Fs, M )

m = log2(M);
N = length( rxSignal );
[ phCarrier, qdCarrier, cSize, dt ] = qamCarrier( Fc, Fs );
nSimbols = N / cSize;

constellation = zeros( nSimbols, 1 );
for i = 1:nSimbols
    index = (i-1)*cSize+1 : i*cSize;

    phds = rxSignal( index ) .* phCarrier;
    qdds = rxSignal( index ) .* qdCarrier;

    phz = 2*trapz( dt, phds )/dt(end);
    qdz = 2*trapz( dt, qdds )/dt(end);

    constellation(i) = round( phz + 1i*qdz );
end

symbols = qamdemod( constellation, M );
sortBits = de2bi( symbols, m );
bits = reshape( sortBits', 1, nSimbols*m );

end

%% GENERACION DE LA PORTADORA QAM
function [ phCarrier, qdCarrier, cSize, dt ] = qamCarrier( Fc, Fs )
    dt = linspace( 0, 1/Fc, Fs/Fc )';

```

```

phCarrier = cos( 2*pi*Fc*dt );
qdCarrier = sin( 2*pi*Fc*dt );
cSize = length(dt);
end
-----
--
function [ outStr ] = initModDemod( tipo, obj, varargin )

    if nargin == 1

        if strcmpi( tipo, 'ask' )
            [ outStr ] = askModDemod();
        elseif strcmpi( tipo, 'fsk' )
            [ outStr ] = fskModDemod();
        elseif strcmpi( tipo, 'bpsk' )
            [ outStr ] = bpskModDemod();
        elseif strcmpi( tipo, 'qpsk' )
            [ outStr ] = qpskModDemod();
        elseif strcmpi( tipo, 'qam' )
            [ outStr ] = qamModDemod();
        else
            [ outStr ] = struct();
        end

    elseif nargin == 2

        if strcmpi( tipo, 'ask' )
            [ outStr ] = askModDemod( '', [], obj.Fc, obj.Fs, obj.Nc,
obj.Ac );
        elseif strcmpi( tipo, 'fsk' )
            [ outStr ] = fskModDemod( '', [], obj.Fc, obj.Fs, obj.Ac );
        elseif strcmpi( tipo, 'bpsk' )
            [ outStr ] = bpskModDemod( '', [], obj.Fc, obj.Fs, obj.Ac,
obj.Nc );
        elseif strcmpi( tipo, 'qpsk' )
            [ outStr ] = qpskModDemod( '', [], obj.Fc, obj.Fs );
        elseif strcmpi( tipo, 'qam' )
            [ outStr ] = qamModDemod( '', [], obj.Fc, obj.Fs, obj.M );
        else
            [ outStr ] = struct();
        end

    elseif nargin >= 3

        Fc = obj;
        Fs = varargin{1};

        if strcmpi( tipo, 'ask' )
            if nargin == 5
                Ac = varargin{2};
                Nc = varargin{3};
            else
                Ac = [ 5, 10 ];
                Nc = 1;
            end
            [ outStr ] = askModDemod( '', [], Fc, Fs, Nc, Ac );

        elseif strcmpi( tipo, 'fsk' )
            if nargin == 4
                Ac = varargin{2};

```



```

        else
            Ac = 1;
        end
        [ outStr ] = fskModDemod( '', [], Fc, Fs, Ac );

elseif strcmpi( tipo, 'bpsk' )
    if nargin == 5
        Ac = varargin{2};
        Nc = varargin{3};
    else
        Ac = [ 5, 10 ];
        Nc = 1;
    end
    [ outStr ] = bpskModDemod( '', [], Fc, Fs, Ac, Nc );

elseif strcmpi( tipo, 'qpsk' )
    [ outStr ] = qpskModDemod( '', [], Fc, Fs );

elseif strcmpi( tipo, 'qam' )
    if nargin == 4
        M = varargin{2};
    else
        M = 4;
    end
    [ outStr ] = qamModDemod( '', [], Fc, Fs, M );

end

else
    [ outStr ] = struct();
end

end

-----
--

```

```

classdef transmisor < matlab.apps.AppBase
    % Properties that correspond to app components
    properties (Access = public)
        TransmisorUIFigure    matlab.ui.Figure
        TabGroup               matlab.ui.container.TabGroup
        ConfiguracionesTab     matlab.ui.container.Tab
        TabGroup2              matlab.ui.container.TabGroup
        ASKTab                 matlab.ui.container.Tab
        FckHzSpinnerLabel      matlab.ui.control.Label
        askFc                  matlab.ui.control.Spinner
        FskHzSpinnerLabel      matlab.ui.control.Label
        askFs                  matlab.ui.control.Spinner
        askA0Label             matlab.ui.control.Label
        askA0                  matlab.ui.control.Knob
        askA1Label             matlab.ui.control.Label
        askA1                  matlab.ui.control.Knob
        askNcLabel             matlab.ui.control.Label
        askNc                  matlab.ui.control.Knob
        FSKTab                 matlab.ui.container.Tab
        fskF0Label             matlab.ui.control.Label
    end
end

```

fskF0	matlab.ui.control.Knob
fskF1Label	matlab.ui.control.Label
fskF1	matlab.ui.control.Knob
FskHzSpinnerLabel_2	matlab.ui.control.Label
fskFs	matlab.ui.control.Spinner
BPSKTab	matlab.ui.container.Tab
bpskNcLabel	matlab.ui.control.Label
bpskNc	matlab.ui.control.Knob
bpskFcLabel	matlab.ui.control.Label
bpskFc	matlab.ui.control.Knob
FskHzSpinnerLabel_3	matlab.ui.control.Label
bpskFs	matlab.ui.control.Spinner
QPSKTab	matlab.ui.container.Tab
qpskFcLabel	matlab.ui.control.Label
qpskFc	matlab.ui.control.Knob
FskHzSpinnerLabel_4	matlab.ui.control.Label
qpskFs	matlab.ui.control.Spinner
QAMTab	matlab.ui.container.Tab
qamFcLabel	matlab.ui.control.Label
qamFc	matlab.ui.control.Knob
qamFs	matlab.ui.control.Spinner
MQAMKnobLabel	matlab.ui.control.Label
qamM	matlab.ui.control.DiscreteKnob
FskHzSpinnerLabel_5	matlab.ui.control.Label
UIAxes	matlab.ui.control.UIAxes
ModulacinPanel	matlab.ui.container.Panel
KnobLabel	matlab.ui.control.Label
Knob	matlab.ui.control.DiscreteKnob
Apply	matlab.ui.control.StateButton
SensoresTab	matlab.ui.container.Tab
CLIENTETCPIPPanel	matlab.ui.container.Panel
IPLabel	matlab.ui.control.Label
tcpipIP	matlab.ui.control.EditField
PuertoEditFieldLabel	matlab.ui.control.Label
tcpipPort	matlab.ui.control.NumericEditField
tcpipBtn	matlab.ui.control.StateButton
TiempoSpinnerLabel	matlab.ui.control.Label
tcpipTime	matlab.ui.control.Spinner
SENSORESPanel	matlab.ui.container.Panel
sensAxes	matlab.ui.control.UIAxes
TemperaturaLabel	matlab.ui.control.Label
sensTemp	matlab.ui.control.Slider
SW2SwitchLabel	matlab.ui.control.Label
sensSW2	matlab.ui.control.RockerSwitch
SW3SwitchLabel	matlab.ui.control.Label
sensSW3	matlab.ui.control.ToggleSwitch
PotenciaKnobLabel	matlab.ui.control.Label
sensPower	matlab.ui.control.Knob
TextoEditFieldLabel	matlab.ui.control.Label
sensTxt	matlab.ui.control.EditField
SW1Label	matlab.ui.control.Label
sensSw1	matlab.ui.control.Switch

```

Label                                matlab.ui.control.Label
end
properties (Access = private)
    modObj        % Objeto para modular
    param         % parametros de modulacion
    recObj        % objeto que graba el audio
    sensores      % Vector de los sensores
    tcpObj        % Objeto para la comunicacion TCP/IP
    audioFs       % Frecuencia de muestreo del Audio
    txtMaxSize    % cantidad maxima de caracteres en el texto
    nSensors      % Cantidad de sensores
    time          % Tiempo inicial
end
methods (Access = private)

    function modPreview( app, type, param )
        obj = initModDemod( type, param );
        data = [ 0 1 0 1 0 1 1 0 0 1 ];
        testSignal = obj.mod( data );
        tSignal = linspace( 0, length(testSignal)/obj.Fs,
length(testSignal) ) * 1e3;
        bits = [ data, data(end) ];
        tBits = linspace( 0, tSignal(end), length(bits) );
        plot( app.UIAxes, tSignal, testSignal )
        hold( app.UIAxes, 'on' )
        stairs( app.UIAxes, tBits, bits )
        hold( app.UIAxes, 'off' )
        legend( app.UIAxes, 'Mod', 'Org' )
        axis( app.UIAxes, [ 0, tSignal(end),
min([min(bits),min(testSignal)])*1.1,
max([max(bits),max(testSignal)])*1.1 ] )
    end

    function showAsk(app)
        app.askA0Label.Text = sprintf( 'A[0] = %.1f',
app.param.ask.Ac(1) );
        app.askA1Label.Text = sprintf( 'A[1] = %.1f',
app.param.ask.Ac(2) );
        app.askNcLabel.Text = sprintf( 'Nc = %.0f', app.param.ask.Nc
);
        app.modPreview( 'ask', app.param.ask )
    end

    function readAsk(app)
        app.param.ask.Fs = app.askFs.Value * 1e3;
        app.param.ask.Fc = app.askFc.Value * 1e3;
        app.param.ask.Ac = [ app.askA0.Value, app.askA1.Value ];
        app.param.ask.Nc = round( app.askNc.Value );
    end

    function enableAsk(app, value)
        app.askA0.Enable = value;
        app.askA1.Enable = value;

```

```

app.askNc.Enable = value;
    app.askFc.Enable = value;
    app.askFs.Enable = value;
end

function showFsk(app)
    app.fskF0Label.Text = sprintf( 'F[0] = %.1f [kHz]',
app.param.fsk.Fc(1)/1e3 );
    app.fskF1Label.Text = sprintf( 'F[1] = %.1f [kHz]',
app.param.fsk.Fc(2)/1e3 );
    app.modPreview( 'fsk', app.param.fsk )
end

function readFsk(app)
    app.param.fsk.Fc = round([ app.fskF0.Value, app.fskF1.Value
]) * 1e3;
    app.param.fsk.Fs = round(app.fskFs.Value) * 1e3;
    app.param.fsk.Ac = 1;
end

function enableFsk(app,value)
    app.fskF0.Enable = value;
    app.fskF1.Enable = value;
    app.fskFs.Enable = value;
end

function showBpsk(app)
    app.bpskFcLabel.Text = sprintf( 'Fc = %.1f [kHz]',
app.param.bpsk.Fc/1e3 );
    app.bpskNcLabel.Text = sprintf( 'Nc = %.0f',
app.param.bpsk.Nc );
    app.modPreview( 'bpsk', app.param.bpsk );
end

function readBpsk(app)
    app.param.bpsk.Fc = round( app.bpskFc.Value ) * 1e3;
    app.param.bpsk.Nc = round( app.bpskNc.Value );
    app.param.bpsk.Fs = round( app.bpskFs.Value ) * 1e3;
    app.param.bpsk.Ac = 1;
end

function enableBpsk(app,value)
    app.bpskFc.Enable = value;
    app.bpskFs.Enable = value;
    app.bpskNc.Enable = value;
end

function showQpsk(app)
    app.qpskFcLabel.Text = sprintf( 'Fc = %.1f [kHz]',
app.param.qpsk.Fc/1e3 );
    app.modPreview( 'qpsk', app.param.qpsk );
end

```

```

function readQpsk(app)
    app.param.qpsk.Fc = round( app.qpskFc.Value ) * 1e3;
    app.param.qpsk.Fs = round( app.qpskFs.Value ) * 1e3;
end

function enableQpsk(app,value)
    app.qpskFc.Enable = value;
    app.qpskFs.Enable = value;
end

function showQam(app)
    app.qamFcLabel.Text = sprintf( 'Fc = %.1f [kHz]',
app.param.qam.Fc/1e3 );
    app.modPreview( 'qam', app.param.qam );
end

function readQam(app)
    app.param.qam.Fc = round( app.qamFc.Value ) * 1e3;
    app.param.qam.Fs = round( app.qamFs.Value ) * 1e3;
    app.param.qam.M = str2double( app.qamM.Value );
end

function enableQam(app,value)
    app.qamFc.Enable = value;
    app.qamFs.Enable = value;
    app.qamM.Enable = value;
end

function temporizador( app, obj, event )
    x = getaudiodata(app.recObj,'uint8');
    l = length( x );
    Fs = 8000;

    timeShow = app.tcpiptime.Value;
    if l / Fs > timeShow
        nShow = timeShow * Fs;
        x = x(end-nShow+1:end);
        l = length( x );
    end

    bits = app.readSensors(x);
    modSignal = app.modObj.mod( bits );
    txSignal = ( modSignal/max(abs(modSignal)) + 1 ) * 255 / 2;
    fwrite( app.tcpObj, txSignal );

    t = linspace( 0, l ./ Fs, l );
    plot( app.sensAxes, t, x );
    axis( app.sensAxes, [ 0, timeShow, 0, 255 ] );

    T = datetime('now') - app.time;
    app.Label.Text = datestr( T, 'HH:MM:SS' );
end

```

```

function bits = num2bin(app, data)
    b = de2bi( data, 8 );
    [r,c] = size(b);
    bits = reshape( b, [r*c, 1] );
end

function num = bin2num( app, bits )
    c = 8;
    r = app.N / c;
    num = bi2de( reshape( bits, [r, c] ) );
end

function bits = readSensors(app, audio)
    slide = uint8( app.sensTemp.Value );
    knob = uint8( app.sensPower.Value );

    aux = uint8( app.sensTxt.Value );
    if length(aux) > app.txtMaxSize
        txt = aux( 1:app.txtMaxSize );
    else
        txt = zeros( app.txtMaxSize, 1 );
        txt( 1:length(aux) ) = aux;
    end

    if strcmpi( app.sensSw1.Value, 'on' )
        sw1 = uint8(1);
    else
        sw1 = uint8(0);
    end
    if strcmpi( app.sensSw2.Value, 'on' )
        sw2 = uint8(1);
    else
        sw2 = uint8(0);
    end
    if strcmpi( app.sensSw3.Value, 'on' )
        sw3 = uint8(1);
    else
        sw3 = uint8(0);
    end
    byteArray = [ audio; slide; knob; sw1; sw2; sw3; txt ];
    bits = app.num2bin(byteArray);
end

function nBytes = calcNBytes(app)
    audioLen = app.tcpipTime.Value * app.audioFs; % [bytes]
    nBytes = ( audioLen + app.txtMaxSize + app.nSensors ) * 8 *
length( app.modObj.dt );

    if strcmpi( app.Knob.Value, 'qpsk' )
        nBytes = nBytes / 2;
    elseif strcmpi( app.Knob.Value, 'qam' )
        nBytes = nBytes / log2( app.modObj.M );
    end

```

```

nBytes = round(nBytes);
end

function recepcion( app, obj, event )
    disp('Recepcion Exitosa')
end

end
methods (Access = private)
    % Code that executes after component creation
    function startupFcn(app)
        app.param.ask = struct();
        app.param.fsk = struct();
        app.param.bpsk = struct();
        app.param.qpsk = struct();
        app.param.qam = struct();

        app.sensores = struct();

        app.readAsk();
        app.readFsk();
        app.readBpsk();
        app.readQpsk();
        app.readQam();

        app.showAsk();

        app.audioFs = 8000;
        app.txtMaxSize = 100;
        app.nSensors = 5;
    end
    % Value changing function: askA0
    function askA0ValueChanging(app, event)
        changingValue = event.Value;
        app.readAsk();
        app.param.ask.Ac(1) = changingValue;
        app.showAsk();
    end
    % Value changing function: askA1
    function askA1ValueChanging(app, event)
        changingValue = event.Value;
        app.readAsk();
        app.param.ask.Ac(2) = changingValue;
        app.showAsk();
    end
    % Value changing function: askNc
    function askNcValueChanging(app, event)
        changingValue = event.Value;
        app.readAsk();
        app.param.ask.Nc = round( changingValue );
        app.showAsk();
    end
    % Value changed function: askFc

```

```

function askFcValueChanged(app, event)
    app.readAsk();
    app.showAsk();
end
% Value changed function: askFs
function askFsValueChanged(app, event)
    app.readAsk();
    app.showAsk();
end
% Value changing function: fskF0
function fskF0ValueChanging(app, event)
    changingValue = event.Value;
    app.readFsk();
    app.param.fsk.Fc(1) = round( changingValue ) * 1e3;
    app.showFsk();
end
% Value changing function: fskF1
function fskF1ValueChanging(app, event)
    changingValue = event.Value;
    app.readFsk();
    app.param.fsk.Fc(2) = round( changingValue ) * 1e3;
    app.showFsk();
end
% Value changed function: fskFs
function fskFsValueChanged(app, event)
    value = app.fskFs.Value;
    app.readFsk();
    app.param.fsk.Fs = round( value ) * 1e3;
    app.showFsk();
end
% Value changing function: bpskFc
function bpskFcValueChanging(app, event)
    changingValue = event.Value;
    app.readBpsk();
    app.param.bpsk.Fc = round( changingValue ) * 1e3;
    app.showBpsk();
end
% Value changing function: bpskNc
function bpskNcValueChanging(app, event)
    changingValue = event.Value;
    app.readBpsk();
    app.param.bpsk.Nc = round( changingValue );
    app.showBpsk();
end
% Value changed function: bpskFs
function bpskFsValueChanged(app, event)
    value = app.bpskFs.Value;
    app.readBpsk();
    app.param.bpsk.Fs = round( value ) * 1e3;
    app.showBpsk();
end
% Value changing function: qpskFc
function qpskFcValueChanging(app, event)

```



```

        changingValue = event.Value;
        app.readQpsk();
        app.param.qpsk.Fc = round( changingValue ) * 1e3;
        app.showQpsk();
    end
    % Value changed function: qpskFs
    function qpskFsValueChanged(app, event)
        value = app.qpskFs.Value;
        app.readQpsk();
        app.param.qpsk.Fs = round( value ) * 1e3;
        app.showQpsk();
    end
    % Value changing function: qamFc
    function qamFcValueChanging(app, event)
        changingValue = event.Value;
        app.readQam();
        app.param.qam.Fc = round( changingValue ) * 1e3;
        app.showQam();
    end
    % Value changed function: qamM
    function qamMValueChanged(app, event)
        value = app.qamM.Value;
        app.readQam();
        app.param.qam.M = str2double( value );
        app.showQam();
    end
    % Value changed function: qamFs
    function qamFsValueChanged(app, event)
        value = app.qamFs.Value;
        app.readQam();
        app.param.qam.Fs = round( value ) * 1e3;
        app.showQam();
    end
    % Value changed function: Knob
    function KnobValueChanged(app, event)
        value = app.Knob.Value;
        app.modObj = initModDemod( value, app.param.(lower(value)) );
        if strcmpi( value, 'ask' )
            app.showAsk();
        elseif strcmpi( value, 'fsk' )
            app.showFsk();
        elseif strcmpi( value, 'bpsk' )
            app.showBpsk();
        elseif strcmpi( value, 'qpsk' )
            app.showQpsk();
        elseif strcmpi( value, 'qam' )
            app.showQam();
        end
    end

    end
    % Value changed function: Apply
    function ApplyValueChanged(app, event)
        value = app.Apply.Value;

```

```

if value
    app.Apply.Text = 'Cambiar';
    en = 'off';
    app.tcpipBtn.Enable = 'on';
else
    app.Apply.Text = 'Seleccionar';
    en = 'on';
    app.tcpipBtn.Enable = 'off';
end

value = app.Knob.Value;
app.modObj = initModDemod( value, app.param.(lower(value)) );

app.enableAsk(en);
app.enableFsk(en);
app.enableBpsk(en);
app.enableQam(en);
app.enableQpsk(en);
app.Knob.Enable = en;
end
% Value changed function: tcpipBtn
function tcpipBtnValueChanged(app, event)
    value = app.tcpipBtn.Value;
    if value

        disp('Conectandose al Servidor')
        nBytes = app.calcNBytes();
        app.tcpObj = tcpip( app.tcpipIP.Value,
app.tcpipPort.Value, 'NetworkRole', 'client' );
        app.tcpObj.InputBufferSize = 2*nBytes;
        app.tcpObj.OutputBufferSize = 2*nBytes;
        app.tcpObj.BytesAvailableFcnCount = nBytes;
        app.tcpObj.BytesAvailableFcnMode = 'byte';
        app.tcpObj.BytesAvailableFcn = @app.recepcion;
        fopen(app.tcpObj);

        app.recObj = audiorecorder( app.audioFs, 8, 1 );
        app.recObj.TimerFcn = @app.temporizador;
        app.recObj.TimerPeriod = app.tcpipTime.Value;
        record( app.recObj );
        disp('Conectado con el Servidor')

        app.time = datetime('now');
        app.TransmisorUIFigure.Name = 'Transmitiendo...';
        app.tcpipBtn.Text = 'Desconectar';
        en = 'off';
    else
        stop( app.recObj );
        fclose(app.tcpObj);

        app.TransmisorUIFigure.Name = 'Transmisor';
        app.tcpipBtn.Text = 'Conectar';
        en = 'on';
    end
end

```

```

disp('Conexion Cerrada')
end

app.tcpipIP.Enable = en;
app.tcpipPort.Enable = en;
app.tcpipTime.Enable = en;
app.Apply.Enable = en;
end
end
% App initialization and construction
methods (Access = private)
% Create UIFigure and components
function createComponents(app)
% Create TransmisorUIFigure
app.TransmisorUIFigure = uifigure;
app.TransmisorUIFigure.Position = [100 100 640 480];
app.TransmisorUIFigure.Name = 'Transmisor';
app.TransmisorUIFigure.Resize = 'off';
% Create TabGroup
app.TabGroup = uitabgroup(app.TransmisorUIFigure);
app.TabGroup.TabLocation = 'bottom';
app.TabGroup.Position = [1 1 640 480];
% Create ConfiguracionesTab
app.ConfiguracionesTab = uitab(app.TabGroup);
app.ConfiguracionesTab.Title = 'Configuraciones';
% Create TabGroup2
app.TabGroup2 = uitabgroup(app.ConfiguracionesTab);
app.TabGroup2.TabLocation = 'left';
app.TabGroup2.Position = [11 225 620 221];
% Create ASKTab
app.ASKTab = uitab(app.TabGroup2);
app.ASKTab.Title = 'ASK';
% Create FckHzSpinnerLabel
app.FckHzSpinnerLabel = uilabel(app.ASKTab);
app.FckHzSpinnerLabel.HorizontalAlignment = 'right';
app.FckHzSpinnerLabel.FontWeight = 'bold';
app.FckHzSpinnerLabel.Position = [78 30 56 22];
app.FckHzSpinnerLabel.Text = 'Fc [kHz]:';
% Create askFc
app.askFc = uispinner(app.ASKTab);
app.askFc.Limits = [1 Inf];
app.askFc.ValueDisplayFormat = '%.2f';
app.askFc.ValueChangedFcn = createCallbackFcn(app,
@askFcValueChanged, true);
app.askFc.HorizontalAlignment = 'center';
app.askFc.FontWeight = 'bold';
app.askFc.Position = [149 30 100 22];
app.askFc.Value = 1;
% Create FskHzSpinnerLabel
app.FskHzSpinnerLabel = uilabel(app.ASKTab);
app.FskHzSpinnerLabel.HorizontalAlignment = 'right';
app.FskHzSpinnerLabel.FontWeight = 'bold';
app.FskHzSpinnerLabel.Position = [327 30 56 22];

```

```

app.FskHzSpinnerLabel.Text = 'Fs [kHz]:';
% Create askFs
app.askFs = uispinner(app.ASKTab);
app.askFs.Step = 10;
app.askFs.Limits = [10 Inf];
app.askFs.ValueDisplayFormat = '%.2f';
app.askFs.ValueChangedFcn = createCallbackFcn(app,
@askFsValueChanged, true);
app.askFs.HorizontalAlignment = 'center';
app.askFs.FontWeight = 'bold';
app.askFs.Position = [398 30 100 22];
app.askFs.Value = 10;
% Create askA0Label
app.askA0Label = uilabel(app.ASKTab);
app.askA0Label.HorizontalAlignment = 'center';
app.askA0Label.FontWeight = 'bold';
app.askA0Label.Position = [81 80 100 22];
app.askA0Label.Text = 'A[0] = 5';
% Create askA0
app.askA0 = uiknob(app.ASKTab, 'continuous');
app.askA0.Limits = [0 20];
app.askA0.ValueChangingFcn = createCallbackFcn(app,
@askA0ValueChanging, true);
app.askA0.Position = [98 128 60 60];
app.askA0.Value = 5;
% Create askA1Label
app.askA1Label = uilabel(app.ASKTab);
app.askA1Label.HorizontalAlignment = 'center';
app.askA1Label.FontWeight = 'bold';
app.askA1Label.Position = [241 80 100 22];
app.askA1Label.Text = 'A[1] = 10';
% Create askA1
app.askA1 = uiknob(app.ASKTab, 'continuous');
app.askA1.Limits = [0 20];
app.askA1.ValueChangingFcn = createCallbackFcn(app,
@askA1ValueChanging, true);
app.askA1.Position = [258 128 60 60];
app.askA1.Value = 10;
% Create askNcLabel
app.askNcLabel = uilabel(app.ASKTab);
app.askNcLabel.HorizontalAlignment = 'center';
app.askNcLabel.FontWeight = 'bold';
app.askNcLabel.Position = [411 80 80 22];
app.askNcLabel.Text = 'Nc = 1';
% Create askNc
app.askNc = uiknob(app.ASKTab, 'continuous');
app.askNc.Limits = [1 10];
app.askNc.ValueChangingFcn = createCallbackFcn(app,
@askNcValueChanging, true);
app.askNc.Position = [418 128 60 60];
app.askNc.Value = 1;
% Create FSKTab
app.FSKTab = uitab(app.TabGroup2);

```

```

app.FSKTab.Title = 'FSK';
% Create fskF0Label
app.fskF0Label = uilabel(app.FSKTab);
app.fskF0Label.HorizontalAlignment = 'center';
app.fskF0Label.FontWeight = 'bold';
app.fskF0Label.Position = [118 73 120 22];
app.fskF0Label.Text = 'F[0] = 1.0 [kHz]';
% Create fskF0
app.fskF0 = uiknob(app.FSKTab, 'continuous');
app.fskF0.Limits = [1 10];
app.fskF0.ValueChangingFcn = createCallbackFcn(app,
@fskF0ValueChanging, true);
app.fskF0.Position = [148 129 60 60];
app.fskF0.Value = 1;
% Create fskF1Label
app.fskF1Label = uilabel(app.FSKTab);
app.fskF1Label.HorizontalAlignment = 'center';
app.fskF1Label.FontWeight = 'bold';
app.fskF1Label.Position = [338 73 120 22];
app.fskF1Label.Text = 'F[1] = 2.0 [kHz]';
% Create fskF1
app.fskF1 = uiknob(app.FSKTab, 'continuous');
app.fskF1.Limits = [1 10];
app.fskF1.ValueChangingFcn = createCallbackFcn(app,
@fskF1ValueChanging, true);
app.fskF1.Position = [368 129 60 60];
app.fskF1.Value = 2;
% Create FskHzSpinnerLabel_2
app.FskHzSpinnerLabel_2 = uilabel(app.FSKTab);
app.FskHzSpinnerLabel_2.HorizontalAlignment = 'right';
app.FskHzSpinnerLabel_2.FontWeight = 'bold';
app.FskHzSpinnerLabel_2.Position = [191 20 56 22];
app.FskHzSpinnerLabel_2.Text = 'Fs [kHz]:';
% Create fskFs
app.fskFs = uispinner(app.FSKTab);
app.fskFs.Step = 10;
app.fskFs.Limits = [10 Inf];
app.fskFs.ValueDisplayFormat = '%.2f';
app.fskFs.ValueChangedFcn = createCallbackFcn(app,
@fskFsValueChanged, true);
app.fskFs.HorizontalAlignment = 'center';
app.fskFs.FontWeight = 'bold';
app.fskFs.Position = [262 20 100 22];
app.fskFs.Value = 10;
% Create BPSKTab
app.BPSKTab = uitab(app.TabGroup2);
app.BPSKTab.Title = 'BPSK';
% Create bpskNcLabel
app.bpskNcLabel = uilabel(app.BPSKTab);
app.bpskNcLabel.HorizontalAlignment = 'center';
app.bpskNcLabel.FontWeight = 'bold';
app.bpskNcLabel.Position = [354 80 80 22];
app.bpskNcLabel.Text = 'Nc = 1';

```

```

% create bpskNc
    app.bpskNc = uiknob(app.BPSKTab, 'continuous');
    app.bpskNc.Limits = [1 10];
    app.bpskNc.ValueChangingFcn = createCallbackFcn(app,
@bpskNcValueChanging, true);
    app.bpskNc.Position = [361 129 60 60];
    app.bpskNc.Value = 1;
    % Create bpskFcLabel
    app.bpskFcLabel = uilabel(app.BPSKTab);
    app.bpskFcLabel.HorizontalAlignment = 'center';
    app.bpskFcLabel.FontWeight = 'bold';
    app.bpskFcLabel.Position = [118 80 120 22];
    app.bpskFcLabel.Text = 'Fc = 1.0 [kHz]';
    % Create bpskFc
    app.bpskFc = uiknob(app.BPSKTab, 'continuous');
    app.bpskFc.Limits = [1 10];
    app.bpskFc.ValueChangingFcn = createCallbackFcn(app,
@bpskFcValueChanging, true);
    app.bpskFc.Position = [148 129 60 60];
    app.bpskFc.Value = 1;
    % Create FskHzSpinnerLabel_3
    app.FskHzSpinnerLabel_3 = uilabel(app.BPSKTab);
    app.FskHzSpinnerLabel_3.HorizontalAlignment = 'right';
    app.FskHzSpinnerLabel_3.FontWeight = 'bold';
    app.FskHzSpinnerLabel_3.Position = [191 20 56 22];
    app.FskHzSpinnerLabel_3.Text = 'Fs [kHz]';
    % Create bpskFs
    app.bpskFs = uispinner(app.BPSKTab);
    app.bpskFs.Step = 10;
    app.bpskFs.Limits = [10 Inf];
    app.bpskFs.ValueDisplayFormat = '%.2f';
    app.bpskFs.ValueChangedFcn = createCallbackFcn(app,
@bpskFsValueChanged, true);
    app.bpskFs.HorizontalAlignment = 'center';
    app.bpskFs.FontWeight = 'bold';
    app.bpskFs.Position = [262 20 100 22];
    app.bpskFs.Value = 10;
    % Create QPSKTab
    app.QPSKTab = uitab(app.TabGroup2);
    app.QPSKTab.Title = 'QPSK';
    % Create qpskFcLabel
    app.qpskFcLabel = uilabel(app.QPSKTab);
    app.qpskFcLabel.HorizontalAlignment = 'center';
    app.qpskFcLabel.FontWeight = 'bold';
    app.qpskFcLabel.Position = [217 80 120 22];
    app.qpskFcLabel.Text = 'Fc = 1.0 [kHz]';
    % Create qpskFc
    app.qpskFc = uiknob(app.QPSKTab, 'continuous');
    app.qpskFc.Limits = [1 10];
    app.qpskFc.ValueChangingFcn = createCallbackFcn(app,
@qpskFcValueChanging, true);
    app.qpskFc.Position = [247 129 60 60];
    app.qpskFc.Value = 1;

```

```

% Create FskHzSpinnerLabel_4
app.FskHzSpinnerLabel_4 = uilabel(app.QPSKTab);
app.FskHzSpinnerLabel_4.HorizontalAlignment = 'right';
app.FskHzSpinnerLabel_4.FontWeight = 'bold';
app.FskHzSpinnerLabel_4.Position = [191 20 56 22];
app.FskHzSpinnerLabel_4.Text = 'Fs [kHz]:';
% Create qpskFs
app.qpskFs = uispinner(app.QPSKTab);
app.qpskFs.Step = 10;
app.qpskFs.Limits = [10 Inf];
app.qpskFs.ValueDisplayFormat = '%.2f';
app.qpskFs.ValueChangedFcn = createCallbackFcn(app,
@qpskFsValueChanged, true);
app.qpskFs.HorizontalAlignment = 'center';
app.qpskFs.FontWeight = 'bold';
app.qpskFs.Position = [262 20 100 22];
app.qpskFs.Value = 10;
% Create QAMTab
app.QAMTab = uitab(app.TabGroup2);
app.QAMTab.Title = 'QAM';
% Create qamFcLabel
app.qamFcLabel = uilabel(app.QAMTab);
app.qamFcLabel.HorizontalAlignment = 'center';
app.qamFcLabel.FontWeight = 'bold';
app.qamFcLabel.Position = [118 80 120 22];
app.qamFcLabel.Text = 'Fc = 1.0 [kHz]';
% Create qamFc
app.qamFc = uiknob(app.QAMTab, 'continuous');
app.qamFc.Limits = [1 10];
app.qamFc.ValueChangingFcn = createCallbackFcn(app,
@qamFcValueChanging, true);
app.qamFc.Position = [148 129 60 60];
app.qamFc.Value = 1;
% Create qamFs
app.qamFs = uispinner(app.QAMTab);
app.qamFs.Step = 10;
app.qamFs.Limits = [10 Inf];
app.qamFs.ValueDisplayFormat = '%.2f';
app.qamFs.ValueChangedFcn = createCallbackFcn(app,
@qamFsValueChanged, true);
app.qamFs.HorizontalAlignment = 'center';
app.qamFs.FontWeight = 'bold';
app.qamFs.Position = [262 20 100 22];
app.qamFs.Value = 10;
% Create MQAMKnobLabel
app.MQAMKnobLabel = uilabel(app.QAMTab);
app.MQAMKnobLabel.HorizontalAlignment = 'center';
app.MQAMKnobLabel.FontWeight = 'bold';
app.MQAMKnobLabel.Position = [368 80 47 22];
app.MQAMKnobLabel.Text = 'M-QAM';
% Create qamM
app.qamM = uiknob(app.QAMTab, 'discrete');
app.qamM.Items = {'2', '4', '16', '32'};

```

```

app.qamM.ValueChangedFcn = createCallbackFcn(app, @qamMValueChanged,
true);
    app.qamM.Position = [360 125 60 60];
    app.qamM.Value = '4';
    % Create FskHzSpinnerLabel_5
    app.FskHzSpinnerLabel_5 = uilabel(app.QAMTab);
    app.FskHzSpinnerLabel_5.HorizontalAlignment = 'right';
    app.FskHzSpinnerLabel_5.FontWeight = 'bold';
    app.FskHzSpinnerLabel_5.Position = [197 20 56 22];
    app.FskHzSpinnerLabel_5.Text = 'Fs [kHz]:';
    % Create UIAxes
    app.UIAxes = uiaxes(app.ConfiguracionesTab);
    title(app.UIAxes, 'Vista Previa')
    xlabel(app.UIAxes, 'Tiempo [mseg]')
    ylabel(app.UIAxes, 'Amplitud')
    app.UIAxes.Box = 'on';
    app.UIAxes.XGrid = 'on';
    app.UIAxes.YGrid = 'on';
    app.UIAxes.Position = [181 16 450 200];
    % Create ModulacinPanel
    app.ModulacinPanel = uipanel(app.ConfiguracionesTab);
    app.ModulacinPanel.Title = 'Modulación:';
    app.ModulacinPanel.FontWeight = 'bold';
    app.ModulacinPanel.Position = [11 6 160 210];
    % Create KnobLabel
    app.KnobLabel = uilabel(app.ModulacinPanel);
    app.KnobLabel.HorizontalAlignment = 'center';
    app.KnobLabel.Position = [68 43 25 22];
    app.KnobLabel.Text = '';
    % Create Knob
    app.Knob = uiknob(app.ModulacinPanel, 'discrete');
    app.Knob.Items = {'ASK', 'FSK', 'BPSK', 'QPSK', 'QAM'};
    app.Knob.ValueChangedFcn = createCallbackFcn(app,
@KnobValueChanged, true);
    app.Knob.FontWeight = 'bold';
    app.Knob.Position = [49 80 60 60];
    app.Knob.Value = 'ASK';
    % Create Apply
    app.Apply = uibutton(app.ModulacinPanel, 'state');
    app.Apply.ValueChangedFcn = createCallbackFcn(app,
@ApplyValueChanged, true);
    app.Apply.Text = 'Seleccionar';
    app.Apply.FontWeight = 'bold';
    app.Apply.Position = [29 19 100 22];
    % Create SensoresTab
    app.SensoresTab = uitab(app.TabGroup);
    app.SensoresTab.Title = 'Sensores';
    % Create CLIENTETCPIPPanel
    app.CLIENTETCPIPPanel = uipanel(app.SensoresTab);
    app.CLIENTETCPIPPanel.Title = 'CLIENTE TCP/IP:';
    app.CLIENTETCPIPPanel.FontWeight = 'bold';
    app.CLIENTETCPIPPanel.Position = [11 386 620 60];
    % Create IPLabel

```



```

app.IPLabel = uilabel(app.CLIENTETCPIPPanel);
app.IPLabel.HorizontalAlignment = 'right';
app.IPLabel.Position = [11 8 25 22];
app.IPLabel.Text = 'IP: ';
% Create tcpipIP
app.tcpipIP = uieditfield(app.CLIENTETCPIPPanel, 'text');
app.tcpipIP.HorizontalAlignment = 'center';
app.tcpipIP.Position = [41 8 100 22];
app.tcpipIP.Value = 'localhost';
% Create PuertoEditFieldLabel
app.PuertoEditFieldLabel = uilabel(app.CLIENTETCPIPPanel);
app.PuertoEditFieldLabel.HorizontalAlignment = 'right';
app.PuertoEditFieldLabel.Position = [181 8 44 22];
app.PuertoEditFieldLabel.Text = 'Puerto: ';
% Create tcpipPort
app.tcpipPort = uieditfield(app.CLIENTETCPIPPanel,
'numeric');
app.tcpipPort.Limits = [0 Inf];
app.tcpipPort.ValueDisplayFormat = '%.0f';
app.tcpipPort.HorizontalAlignment = 'center';
app.tcpipPort.Position = [231 8 69 22];
app.tcpipPort.Value = 30000;
% Create tcpipBtn
app.tcpipBtn = uibutton(app.CLIENTETCPIPPanel, 'state');
app.tcpipBtn.ValueChangedFcn = createCallbackFcn(app,
@tcpipBtnValueChanged, true);
app.tcpipBtn.Enable = 'off';
app.tcpipBtn.Text = 'Conectar';
app.tcpipBtn.Position = [511 8 100 22];
% Create TiempoSpinnerLabel
app.TiempoSpinnerLabel = uilabel(app.CLIENTETCPIPPanel);
app.TiempoSpinnerLabel.HorizontalAlignment = 'right';
app.TiempoSpinnerLabel.Position = [339 8 48 22];
app.TiempoSpinnerLabel.Text = 'Tiempo: ';
% Create tcpipTime
app.tcpipTime = uispinner(app.CLIENTETCPIPPanel);
app.tcpipTime.Step = 0.05;
app.tcpipTime.Limits = [0.05 1];
app.tcpipTime.HorizontalAlignment = 'center';
app.tcpipTime.Position = [399 8 73 22];
app.tcpipTime.Value = 0.25;
% Create SENSORESPanel
app.SENSORESPanel = uipanel(app.SensoresTab);
app.SENSORESPanel.Title = 'SENSORES: ';
app.SENSORESPanel.FontWeight = 'bold';
app.SENSORESPanel.Position = [10 6 621 371];
% Create sensAxes
app.sensAxes = uiaxes(app.SENSORESPanel);
title(app.sensAxes, 'Muestra de Audio')
xlabel(app.sensAxes, 'Tiempo [seg]')
ylabel(app.sensAxes, 'Amplitud')
app.sensAxes.Box = 'on';
app.sensAxes.XGrid = 'on';

```

```

app.sensAxes.YGrid = 'on';
app.sensAxes.Position = [151 6 460 155];
% Create TemperaturaLabel
app.TemperaturaLabel = uilabel(app.SENSORESPanel);
app.TemperaturaLabel.HorizontalAlignment = 'right';
app.TemperaturaLabel.Position = [11 319 76 22];
app.TemperaturaLabel.Text = 'Temperatura: ';
% Create sensTemp
app.sensTemp = uislider(app.SENSORESPanel);
app.sensTemp.Orientation = 'vertical';
app.sensTemp.Position = [48 58 3 246];
% Create SW2SwitchLabel
app.SW2SwitchLabel = uilabel(app.SENSORESPanel);
app.SW2SwitchLabel.HorizontalAlignment = 'center';
app.SW2SwitchLabel.Position = [471 219 32 22];
app.SW2SwitchLabel.Text = 'SW2';
% Create sensSW2
app.sensSW2 = uiswitch(app.SENSORESPanel, 'rocker');
app.sensSW2.Position = [477 277 20 45];
% Create SW3SwitchLabel
app.SW3SwitchLabel = uilabel(app.SENSORESPanel);
app.SW3SwitchLabel.HorizontalAlignment = 'center';
app.SW3SwitchLabel.Position = [551 219 32 22];
app.SW3SwitchLabel.Text = 'SW3';
% Create sensSW3
app.sensSW3 = uiswitch(app.SENSORESPanel, 'toggle');
app.sensSW3.Position = [557 277 20 45];
% Create PotenciaKnobLabel
app.PotenciaKnobLabel = uilabel(app.SENSORESPanel);
app.PotenciaKnobLabel.HorizontalAlignment = 'center';
app.PotenciaKnobLabel.Position = [202 219 52 22];
app.PotenciaKnobLabel.Text = 'Potencia';
% Create sensPower
app.sensPower = uiknob(app.SENSORESPanel, 'continuous');
app.sensPower.Position = [199 257 60 60];
% Create TextoEditFieldLabel
app.TextoEditFieldLabel = uilabel(app.SENSORESPanel);
app.TextoEditFieldLabel.HorizontalAlignment = 'right';
app.TextoEditFieldLabel.Position = [181 179 38 22];
app.TextoEditFieldLabel.Text = 'Texto: ';
% Create sensTxt
app.sensTxt = uieditfield(app.SENSORESPanel, 'text');
app.sensTxt.HorizontalAlignment = 'center';
app.sensTxt.Position = [231 179 359 22];
app.sensTxt.Value = 'Texto de Prueba';
% Create SW1Label
app.SW1Label = uilabel(app.SENSORESPanel);
app.SW1Label.HorizontalAlignment = 'center';
app.SW1Label.Position = [363 244 32 22];
app.SW1Label.Text = 'SW1';
% Create sensSw1
app.sensSw1 = uiswitch(app.SENSORESPanel, 'slider');
app.sensSw1.Position = [355 281 45 20];

```

```

% Create Label
    app.Label = uilabel(app.SENSORESPanel);
    app.Label.HorizontalAlignment = 'center';
    app.Label.FontSize = 16;
    app.Label.FontWeight = 'bold';
    app.Label.Position = [4 9 137 22];
    app.Label.Text = '--:--:--';
end
end
methods (Access = public)
    % Construct app
    function app = transmisor
        % Create and configure components
        createComponents(app)
        % Register the app with App Designer
        registerApp(app, app.TransmisorUIFigure)
        % Execute the startup function
        runStartupFcn(app, @startupFcn)
        if nargin == 0
            clear app
        end
    end
    % Code that executes before app deletion
    function delete(app)
        % Delete UIFigure when app is deleted
        delete(app.TransmisorUIFigure)
    end
end
end
-----
--
classdef receptor < matlab.apps.AppBase
    % Properties that correspond to app components
    properties (Access = public)
        ReceptorUIFigure          matlab.ui.Figure
        TabGroup                   matlab.ui.container.TabGroup
        ConfiguracionesTab         matlab.ui.container.Tab
        TabGroup2                  matlab.ui.container.TabGroup
        ASKTab                     matlab.ui.container.Tab
        FckHzSpinnerLabel          matlab.ui.control.Label
        askFc                      matlab.ui.control.Spinner
        FskHzSpinnerLabel          matlab.ui.control.Label
        askFs                      matlab.ui.control.Spinner
        askA0Label                 matlab.ui.control.Label
        askA0                      matlab.ui.control.Knob
        askA1Label                 matlab.ui.control.Label
        askA1                      matlab.ui.control.Knob
        askNcLabel                 matlab.ui.control.Label
        askNc                      matlab.ui.control.Knob
        FSKTab                     matlab.ui.container.Tab
        fskF0Label                 matlab.ui.control.Label
        fskF0                      matlab.ui.control.Knob
        fskF1Label                 matlab.ui.control.Label
    end
end

```

fskF1	matlab.ui.control.Knob
FskHzSpinnerLabel_2	matlab.ui.control.Label
fskFs	matlab.ui.control.Spinner
BPSKTab	matlab.ui.container.Tab
bpskNcLabel	matlab.ui.control.Label
bpskNc	matlab.ui.control.Knob
bpskFcLabel	matlab.ui.control.Label
bpskFc	matlab.ui.control.Knob
FskHzSpinnerLabel_3	matlab.ui.control.Label
bpskFs	matlab.ui.control.Spinner
QPSKTab	matlab.ui.container.Tab
qpskFcLabel	matlab.ui.control.Label
qpskFc	matlab.ui.control.Knob
FskHzSpinnerLabel_4	matlab.ui.control.Label
qpskFs	matlab.ui.control.Spinner
QAMTab	matlab.ui.container.Tab
qamFcLabel	matlab.ui.control.Label
qamFc	matlab.ui.control.Knob
qamFs	matlab.ui.control.Spinner
MQAMKnobLabel	matlab.ui.control.Label
qamM	matlab.ui.control.DiscreteKnob
FskHzSpinnerLabel_5	matlab.ui.control.Label
UIAxes	matlab.ui.control.UIAxes
ModulacinPanel	matlab.ui.container.Panel
KnobLabel	matlab.ui.control.Label
Knob	matlab.ui.control.DiscreteKnob
Apply	matlab.ui.control.StateButton
SensoresTab	matlab.ui.container.Tab
SERVIDORTCIPPanel	matlab.ui.container.Panel
IPLabel	matlab.ui.control.Label
tcPIP	matlab.ui.control.EditField
PuertoEditFieldLabel	matlab.ui.control.Label
tcPIPPort	matlab.ui.control.NumericEditField
tcPIPBtn	matlab.ui.control.StateButton
TiempoSpinnerLabel	matlab.ui.control.Label
tcPIPTime	matlab.ui.control.Spinner
DISPLAYSPanel	matlab.ui.container.Panel
sensAxes	matlab.ui.control.UIAxes
TextoEditFieldLabel	matlab.ui.control.Label
sensTxt	matlab.ui.control.EditField
TemperaturaLabel	matlab.ui.control.Label
sensTemp	matlab.ui.control.LinearGauge
Label	matlab.ui.control.Label
sensPower	matlab.ui.control.Gauge
LED1Label	matlab.ui.control.Label
sensSW1	matlab.ui.control.Lamp
LED2Label	matlab.ui.control.Label
sensSW2	matlab.ui.control.Lamp
LED3Label	matlab.ui.control.Label
sensSW3	matlab.ui.control.Lamp
TiempoLabel	matlab.ui.control.Label
PotenciaLabel	matlab.ui.control.Label
TiempodeTransmisionLabel	matlab.ui.control.Label

```

end

properties (Access = private)
    modObj      % Objeto para modular
    param      % parametros de modulacion
    recObj      % objeto que graba el audio
    sensores    % Vector de los sensores
    tcpObj      % Objeto para la comunicacion TCP/IP
    audioFs     % Frecuencia de muestreo del Audio
    txtMaxSize  % cantidad maxima de caracteres en el texto
    nSensors    % Cantidad de sensores
    time        % Tiempo transcurrido
end
methods (Access = private)

    function modPreview( app, type, param )
        obj = initModDemod( type, param );
        data = [ 0 1 0 1 0 1 1 0 0 1 ];
        testSignal = obj.mod( data );
        tSignal = linspace( 0, length(testSignal)/obj.Fs,
length(testSignal) ) * 1e3;
        bits = [ data, data(end) ];
        tBits = linspace( 0, tSignal(end), length(bits) );
        plot( app.UIAxes, tSignal, testSignal )
        hold( app.UIAxes, 'on' )
        stairs( app.UIAxes, tBits, bits )
        hold( app.UIAxes, 'off' )
        legend( app.UIAxes, 'Mod', 'Org' )
        axis( app.UIAxes, [ 0, tSignal(end),
min([min(bits),min(testSignal)])*1.1,
max([max(bits),max(testSignal)])*1.1 ] )
    end

    function showAsk(app)
        app.askA0Label.Text = sprintf( 'A[0] = %.1f',
app.param.ask.Ac(1) );
        app.askA1Label.Text = sprintf( 'A[1] = %.1f',
app.param.ask.Ac(2) );
        app.askNcLabel.Text = sprintf( 'Nc = %.0f', app.param.ask.Nc
);
        app.modPreview( 'ask', app.param.ask )
    end

    function readAsk(app)
        app.param.ask.Fs = app.askFs.Value * 1e3;
        app.param.ask.Fc = app.askFc.Value * 1e3;
        app.param.ask.Ac = [ app.askA0.Value, app.askA1.Value ];
        app.param.ask.Nc = round( app.askNc.Value );
    end

    function enableAsk(app, value)
        app.askA0.Enable = value;
        app.askA1.Enable = value;
        app.askNc.Enable = value;

```

```

app.askFc.Enable = value;
    app.askFs.Enable = value;
end

function showFsk(app)
    app.fskF0Label.Text = sprintf( 'F[0] = %.1f [kHz]',
app.param.fsk.Fc(1)/1e3 );
    app.fskF1Label.Text = sprintf( 'F[1] = %.1f [kHz]',
app.param.fsk.Fc(2)/1e3 );
    app.modPreview( 'fsk', app.param.fsk )
end

function readFsk(app)
    app.param.fsk.Fc = round([ app.fskF0.Value, app.fskF1.Value
]) * 1e3;
    app.param.fsk.Fs = round(app.fskFs.Value) * 1e3;
    app.param.fsk.Ac = 1;
end

function enableFsk(app,value)
    app.fskF0.Enable = value;
    app.fskF1.Enable = value;
    app.fskFs.Enable = value;
end

function showBpsk(app)
    app.bpskFcLabel.Text = sprintf( 'Fc = %.1f [kHz]',
app.param.bpsk.Fc/1e3 );
    app.bpskNcLabel.Text = sprintf( 'Nc = %.0f',
app.param.bpsk.Nc );
    app.modPreview( 'bpsk', app.param.bpsk );
end

function readBpsk(app)
    app.param.bpsk.Fc = round( app.bpskFc.Value ) * 1e3;
    app.param.bpsk.Nc = round( app.bpskNc.Value );
    app.param.bpsk.Fs = round( app.bpskFs.Value ) * 1e3;
    app.param.bpsk.Ac = 1;
end

function enableBpsk(app,value)
    app.bpskFc.Enable = value;
    app.bpskFs.Enable = value;
    app.bpskNc.Enable = value;
end

function showQpsk(app)
    app.qpskFcLabel.Text = sprintf( 'Fc = %.1f [kHz]',
app.param.qpsk.Fc/1e3 );
    app.modPreview( 'qpsk', app.param.qpsk );
end

function readQpsk(app)

```

```

app.param.qpsk.Fc = round( app.qpskFc.Value ) * 1e3;
    app.param.qpsk.Fs = round( app.qpskFs.Value ) * 1e3;
end

function enableQpsk(app,value)
    app.qpskFc.Enable = value;
    app.qpskFs.Enable = value;
end

function showQam(app)
    app.qamFcLabel.Text = sprintf( 'Fc = %.1f [kHz]',
app.param.qam.Fc/1e3 );
    app.modPreview( 'qam', app.param.qam );
end

function readQam(app)
    app.param.qam.Fc = round( app.qamFc.Value ) * 1e3;
    app.param.qam.Fs = round( app.qamFs.Value ) * 1e3;
    app.param.qam.M = str2double( app.qamM.Value );
end

function enableQam(app,value)
    app.qamFc.Enable = value;
    app.qamFs.Enable = value;
    app.qamM.Enable = value;
end

function temporizador( app, obj, event )
    x = getaudiodata(app.recObj,'uint8');
    l = length( x );
    Fs = 8000;

    timeShow = app.tcpiTime.Value;
    if l / Fs > timeShow
        nShow = timeShow * Fs;
        x = x(end-nShow+1:end);
        l = length( x );
    end

    bits = app.readSensors(x);
    modSignal = app.modObj.mod( bits );
    txSignal = ( modSignal/max(abs(modSignal)) + 1 ) * 255 / 2;
    fwrite( app.tcpObj, txSignal );

    t = linspace( 0, l ./ Fs, l );
    plot( app.sensAxes, t, x );
    axis( app.sensAxes, [ 0, timeShow, 0, 255 ] );
end

function bits = num2bin(app, data)
    b = de2bi( data, 8 );
    [r,c] = size(b);
    bits = reshape( b, [r*c, 1] );

```

```

end

function num = bin2num( app, bits )
    c = 8;
    r = length(bits) / c;
    num = bi2de( reshape( bits, [r, c] ) );
end
function bits = readSensors(app,audio)
%     slide = uint8( app.sensTemp.Value );
%     knob =  uint8( app.sensPower.Value );

    aux = uint8( app.sensTxt.Value )';
    if length(aux) > app.txtMaxSize
        txt = aux( 1:app.txtMaxSize );
    else
        txt = zeros( app.txtMaxSize, 1 );
        txt( 1:length(aux) ) = aux;
    end

%         if strcmpi( app.sensSw1.Value, 'on' )
%             sw1 = uint8(1);
%         else
%             sw1 = uint8(0);
%         end
%         if strcmpi( app.sensSw2.Value, 'on' )
%             sw2 = uint8(1);
%         else
%             sw2 = uint8(0);
%         end
%         if strcmpi( app.sensSw3.Value, 'on' )
%             sw3 = uint8(1);
%         else
%             sw3 = uint8(0);
%         end
    byteArray = [ audio; slide; knob; sw1; sw2; sw3; txt ];
    bits = app.num2bin(byteArray);
end

function nBytes = calcNBytes(app)
    audioLen = app.tcpipTime.Value * app.audioFs; % [bytes]
    nBytes = ( audioLen + app.txtMaxSize + app.nSensors ) * 8 *
length( app.modObj.dt );

    if strcmpi( app.Knob.Value, 'qpsk' )
        nBytes = nBytes / 2;
    elseif strcmpi( app.Knob.Value, 'qam' )
        nBytes = nBytes / log2( app.modObj.M );
    end
    nBytes = round(nBytes);
end

```



```

Function recepcion (app, obj, event)

    T = datetime('now') - app.time;
    app.TiempoLabel.Text = datestr( T, 'HH:MM:SS' );

    if app.tcpObj.BytesAvailable == app.calcNBytes()
        rxSignal = fread( app.tcpObj, app.tcpObj.BytesAvailable
    );

        if strcmpi( 'ask', app.Knob.Value )
            Ac = app.modObj.Ac(2);
        elseif strcmpi( 'fsk', app.Knob.Value )
            Ac = app.modObj.Ac;
        elseif strcmpi( 'qam', app.Knob.Value )
            switch app.modObj.M
                case 4
                    Ac = sqrt(2);
                case 16
                    Ac = 4.242639375961311;
                case 8
                    Ac = 3.162276118809079;
                case 32
                    Ac = 5.830951236124022;
                case 64
                    Ac = 9.899491877243058;
                case 128
                    Ac = 13.038401826991130;
                otherwise
                    Ac = 1;
            end
        else
            Ac = 1;
        end

        processSignal = ( double(rxSignal)*2/255 - 1 ) * Ac;

        bits = app.modObj.dem( processSignal );
        byteArray = app.bin2num( bits );
        audiolen = app.tcpipTime.Value * app.audioFs;

        audio = byteArray( 1:audiolen );
        txt = byteArray( end-app.txtMaxSize+1:end )';
        txt( txt == 0 ) = [];
        txt = char(txt);

        slide = byteArray( audiolen + 1 );
        knob = byteArray( audiolen + 2 );
        sw1 = byteArray( audiolen + 3 );
        sw2 = byteArray( audiolen + 4 );
        sw3 = byteArray( audiolen + 5 );

        app.sensTemp.Value = slide;
        app.sensPower.Value = knob;
    end
end

```

```

        app.sensSW1.Color = double ([0, sw1, 0]);
        app.sensSW2.Color = double ([sw2, 0, 0]);
        app.sensSW3.Color = double([ sw3, sw3, 0 ]);
        app.sensTxt.Value = txt;

        soundsc( double(audio), app.audioFs );

        t = linspace( 0, audiolen ./ app.audioFs, audiolen );
        plot( app.sensAxes, t, audio )
        axis( app.sensAxes, [ 0, t(end), 0, 255 ] );
    else
        if app.tcpObj.BytesAvailable > 0
            fread( app.tcpObj, app.tcpObj.BytesAvailable );
        end
    end
end
end

end
methods (Access = private)
% Code that executes after component creation
function startupFcn(app)
    app.param.ask = struct();
    app.param.fsk = struct();
    app.param.bpsk = struct();
    app.param.qpsk = struct();
    app.param.qam = struct();

    app.sensores = struct();

    app.readAsk();
    app.readFsk();
    app.readBpsk();
    app.readQpsk();
    app.readQam();

    app.showAsk();

    app.audioFs = 8000;
    app.txtMaxSize = 100;
    app.nSensors = 5;
end
% Value changing function: askA0
function askA0ValueChanging(app, event)
    changingValue = event.Value;
    app.readAsk();
    app.param.ask.Ac(1) = changingValue;
    app.showAsk();
end
% Value changing function: askA1
function askA1ValueChanging(app, event)
    changingValue = event.Value;
    app.readAsk();

```

```

app.param.ask.Ac(2) = changingValue;
    app.showAsk();
end
% Value changing function: askNc
function askNcValueChanging(app, event)
    changingValue = event.Value;
    app.readAsk();
    app.param.ask.Nc = round( changingValue );
    app.showAsk();
end
% Value changed function: askFc
function askFcValueChanged(app, event)
    app.readAsk();
    app.showAsk();
end
% Value changed function: askFs
function askFsValueChanged(app, event)
    app.readAsk();
    app.showAsk();
end
% Value changing function: fskF0
function fskF0ValueChanging(app, event)
    changingValue = event.Value;
    app.readFsk();
    app.param.fsk.Fc(1) = round( changingValue ) * 1e3;
    app.showFsk();
end
% Value changing function: fskF1
function fskF1ValueChanging(app, event)
    changingValue = event.Value;
    app.readFsk();
    app.param.fsk.Fc(2) = round( changingValue ) * 1e3;
    app.showFsk();
end
% Value changed function: fskFs
function fskFsValueChanged(app, event)
    value = app.fskFs.Value;
    app.readFsk();
    app.param.fsk.Fs = round( value ) * 1e3;
    app.showFsk();
end
% Value changing function: bpskFc
function bpskFcValueChanging(app, event)
    changingValue = event.Value;
    app.readBpsk();
    app.param.bpsk.Fc = round( changingValue ) * 1e3;
    app.showBpsk();
end
% Value changing function: bpskNc
function bpskNcValueChanging(app, event)
    changingValue = event.Value;
    app.readBpsk();
    app.param.bpsk.Nc = round( changingValue );

```

```

        app.showBpsk();
    end
    % Value changed function: bpskFs
    function bpskFsValueChanged(app, event)
        value = app.bpskFs.Value;
        app.readBpsk();
        app.param.bpsk.Fs = round( value ) * 1e3;
        app.showBpsk();
    end
    % Value changing function: qpskFc
    function qpskFcValueChanging(app, event)
        changingValue = event.Value;
        app.readQpsk();
        app.param.qpsk.Fc = round( changingValue ) * 1e3;
        app.showQpsk();
    end
    % Value changed function: qpskFs
    function qpskFsValueChanged(app, event)
        value = app.qpskFs.Value;
        app.readQpsk();
        app.param.qpsk.Fs = round( value ) * 1e3;
        app.showQpsk();
    end
    % Value changing function: qamFc
    function qamFcValueChanging(app, event)
        changingValue = event.Value;
        app.readQam();
        app.param.qam.Fc = round( changingValue ) * 1e3;
        app.showQam();
    end
    % Value changed function: qamM
    function qamMValueChanged(app, event)
        value = app.qamM.Value;
        app.readQam();
        app.param.qam.M = str2double( value );
        app.showQam();
    end
    % Value changed function: qamFs
    function qamFsValueChanged(app, event)
        value = app.qamFs.Value;
        app.readQam();
        app.param.qam.Fs = round( value ) * 1e3;
        app.showQam();
    end
    % Value changed function: Knob
    function KnobValueChanged(app, event)
        value = app.Knob.Value;
        app.modObj = initModDemod( value, app.param.(lower(value)) );
        if strcmpi( value, 'ask' )
            app.showAsk();
        elseif strcmpi( value, 'fsk' )
            app.showFsk();
        elseif strcmpi( value, 'bpsk' )

```

```

app.showBpsk();
    elseif strcmpi( value, 'qpsk' )

        app.showQpsk();
    elseif strcmpi( value, 'qam' )
        app.showQam();
    end

end

% Value changed function: Apply
function ApplyValueChanged(app, event)
    value = app.Apply.Value;
    if value
        app.Apply.Text = 'Cambiar';
        en = 'off';
        app.tcpipBtn.Enable = 'on';
    else
        app.Apply.Text = 'Seleccionar';
        en = 'on';
        app.tcpipBtn.Enable = 'off';
    end

    value = app.Knob.Value;
    app.modObj = initModDemod( value, app.param.(lower(value)) );

    app.enableAsk(en);
    app.enableFsk(en);
    app.enableBpsk(en);
    app.enableQam(en);
    app.enableQpsk(en);
    app.Knob.Enable = en;
end

% Value changed function: tcpipBtn
function tcpipBtnValueChanged(app, event)
    value = app.tcpipBtn.Value;
    if value

        disp('Abriendo Servidor')
        nBytes = app.calcNBytes();
        app.tcpObj = tcpip( app.tcpipIP.Value,
app.tcpipPort.Value, 'NetworkRole', 'server' );
        app.tcpObj.InputBufferSize = 2*nBytes;
        app.tcpObj.OutputBufferSize = 2*nBytes;
        app.tcpObj.BytesAvailableFcnCount = nBytes;
        app.tcpObj.BytesAvailableFcnMode = 'byte';
        app.tcpObj.BytesAvailableFcn = @app.recepcion;
        fopen(app.tcpObj);
        disp('Cliente conectado')

        app.time = datetime('now');
        app.ReceptorUIFigure.Name = 'Recibiendo...';
        app.tcpipBtn.Text = 'Desconectar';
        en = 'off';
    end
end

```

```

else
    fclose(app.tcpObj);
    app.ReceptorUIFigure.Name = 'Receptor';
    app.tcpipBtn.Text = 'Conectar';
    en = 'on';
    disp('Conexion Cerrada')
end

app.tcpipIP.Enable = en;
app.tcpipPort.Enable = en;
app.tcpipTime.Enable = en;
app.Apply.Enable = en;
end

end
% App initialization and construction
methods (Access = private)
% Create UIFigure and components
function createComponents(app)
    % Create ReceptorUIFigure
    app.ReceptorUIFigure = uifigure;
    app.ReceptorUIFigure.Position = [100 100 640 480];
    app.ReceptorUIFigure.Name = 'Receptor';
    app.ReceptorUIFigure.Resize = 'off';
    % Create TabGroup
    app.TabGroup = uitabgroup(app.ReceptorUIFigure);
    app.TabGroup.TabLocation = 'bottom';
    app.TabGroup.Position = [1 1 640 480];
    % Create ConfiguracionesTab
    app.ConfiguracionesTab = uitab(app.TabGroup);
    app.ConfiguracionesTab.Title = 'Configuraciones';
    % Create TabGroup2
    app.TabGroup2 = uitabgroup(app.ConfiguracionesTab);
    app.TabGroup2.TabLocation = 'left';
    app.TabGroup2.Position = [11 225 620 221];
    % Create ASKTab
    app.ASKTab = uitab(app.TabGroup2);
    app.ASKTab.Title = 'ASK';
    % Create FckHzSpinnerLabel
    app.FckHzSpinnerLabel = uilabel(app.ASKTab);
    app.FckHzSpinnerLabel.HorizontalAlignment = 'right';
    app.FckHzSpinnerLabel.FontWeight = 'bold';
    app.FckHzSpinnerLabel.Position = [78 30 56 22];
    app.FckHzSpinnerLabel.Text = 'Fc [kHz]:';
    % Create askFc
    app.askFc = uispinner(app.ASKTab);
    app.askFc.Limits = [1 Inf];
    app.askFc.ValueDisplayFormat = '%.2f';
    app.askFc.ValueChangedFcn = createCallbackFcn(app,
@askFcValueChanged, true);
    app.askFc.HorizontalAlignment = 'center';
    app.askFc.FontWeight = 'bold';
    app.askFc.Position = [149 30 100 22];
    app.askFc.Value = 1;

```

```

% Create FskHzSpinnerLabel
app.FskHzSpinnerLabel = uilabel(app.ASKTab);
app.FskHzSpinnerLabel.HorizontalAlignment = 'right';
app.FskHzSpinnerLabel.FontWeight = 'bold';
app.FskHzSpinnerLabel.Position = [327 30 56 22];
app.FskHzSpinnerLabel.Text = 'Fs [kHz]:';
% Create askFs
app.askFs = uispinner(app.ASKTab);
app.askFs.Step = 10;
app.askFs.Limits = [10 Inf];
app.askFs.ValueDisplayFormat = '%.2f';
app.askFs.ValueChangedFcn = createCallbackFcn(app,
@askFsValueChanged, true);
app.askFs.HorizontalAlignment = 'center';
app.askFs.FontWeight = 'bold';
app.askFs.Position = [398 30 100 22];
app.askFs.Value = 10;
% Create askA0Label
app.askA0Label = uilabel(app.ASKTab);
app.askA0Label.HorizontalAlignment = 'center';
app.askA0Label.FontWeight = 'bold';
app.askA0Label.Position = [81 80 100 22];
app.askA0Label.Text = 'A[0] = 5';
% Create askA0
app.askA0 = uiknob(app.ASKTab, 'continuous');
app.askA0.Limits = [0 20];
app.askA0.ValueChangingFcn = createCallbackFcn(app,
@askA0ValueChanging, true);
app.askA0.Position = [98 128 60 60];
app.askA0.Value = 5;
% Create askA1Label
app.askA1Label = uilabel(app.ASKTab);
app.askA1Label.HorizontalAlignment = 'center';
app.askA1Label.FontWeight = 'bold';
app.askA1Label.Position = [241 80 100 22];
app.askA1Label.Text = 'A[1] = 10';
% Create askA1
app.askA1 = uiknob(app.ASKTab, 'continuous');
app.askA1.Limits = [0 20];
app.askA1.ValueChangingFcn = createCallbackFcn(app,
@askA1ValueChanging, true);
app.askA1.Position = [258 128 60 60];
app.askA1.Value = 10;
% Create askNcLabel
app.askNcLabel = uilabel(app.ASKTab);
app.askNcLabel.HorizontalAlignment = 'center';
app.askNcLabel.FontWeight = 'bold';
app.askNcLabel.Position = [411 80 80 22];
app.askNcLabel.Text = 'Nc = 1';
% Create askNc
app.askNc = uiknob(app.ASKTab, 'continuous');
app.askNc.Limits = [1 10];
app.askNc.ValueChangingFcn = createCallbackFcn(app,

```

```

@askNcValueChanging, true);
    app.askNc.Position = [418 128 60 60];
    app.askNc.Value = 1;
    % Create FSKTab
    app.FSKTab = uitab(app.TabGroup2);
    app.FSKTab.Title = 'FSK';
    % Create fskF0Label
    app.fskF0Label = uilabel(app.FSKTab);
    app.fskF0Label.HorizontalAlignment = 'center';
    app.fskF0Label.FontWeight = 'bold';
    app.fskF0Label.Position = [118 73 120 22];
    app.fskF0Label.Text = 'F[0] = 1.0 [kHz]';
    % Create fskF0
    app.fskF0 = uiknob(app.FSKTab, 'continuous');
    app.fskF0.Limits = [1 10];
    app.fskF0.ValueChangingFcn = createCallbackFcn(app,
@fskF0ValueChanging, true);
    app.fskF0.Position = [148 129 60 60];
    app.fskF0.Value = 1;
    % Create fskF1Label
    app.fskF1Label = uilabel(app.FSKTab);
    app.fskF1Label.HorizontalAlignment = 'center';
    app.fskF1Label.FontWeight = 'bold';
    app.fskF1Label.Position = [338 73 120 22];
    app.fskF1Label.Text = 'F[1] = 2.0 [kHz]';
    % Create fskF1
    app.fskF1 = uiknob(app.FSKTab, 'continuous');
    app.fskF1.Limits = [1 10];
    app.fskF1.ValueChangingFcn = createCallbackFcn(app,
@fskF1ValueChanging, true);
    app.fskF1.Position = [368 129 60 60];
    app.fskF1.Value = 2;
    % Create FskHzSpinnerLabel_2
    app.FskHzSpinnerLabel_2 = uilabel(app.FSKTab);
    app.FskHzSpinnerLabel_2.HorizontalAlignment = 'right';
    app.FskHzSpinnerLabel_2.FontWeight = 'bold';
    app.FskHzSpinnerLabel_2.Position = [191 20 56 22];
    app.FskHzSpinnerLabel_2.Text = 'Fs [kHz]:';
    % Create fskFs
    app.fskFs = uispinner(app.FSKTab);
    app.fskFs.Step = 10;
    app.fskFs.Limits = [10 Inf];
    app.fskFs.ValueDisplayFormat = '%.2f';
    app.fskFs.ValueChangedFcn = createCallbackFcn(app,
@fskFsValueChanged, true);
    app.fskFs.HorizontalAlignment = 'center';
    app.fskFs.FontWeight = 'bold';
    app.fskFs.Position = [262 20 100 22];
    app.fskFs.Value = 10;
    % Create BPSKTab
    app.BPSKTab = uitab(app.TabGroup2);
    app.BPSKTab.Title = 'BPSK';
    % Create bpskNcLabel

```



```

app.bpskNcLabel = uilabel(app.BPSKTab);
app.bpskNcLabel.HorizontalAlignment = 'center';
app.bpskNcLabel.FontWeight = 'bold';
app.bpskNcLabel.Position = [354 80 80 22];
app.bpskNcLabel.Text = 'Nc = 1';
% Create bpskNc
app.bpskNc = uiknob(app.BPSKTab, 'continuous');
app.bpskNc.Limits = [1 10];
app.bpskNc.ValueChangingFcn = createCallbackFcn(app,
@bpskNcValueChanging, true);
app.bpskNc.Position = [361 129 60 60];
app.bpskNc.Value = 1;
% Create bpskFcLabel
app.bpskFcLabel = uilabel(app.BPSKTab);
app.bpskFcLabel.HorizontalAlignment = 'center';
app.bpskFcLabel.FontWeight = 'bold';
app.bpskFcLabel.Position = [118 80 120 22];
app.bpskFcLabel.Text = 'Fc = 1.0 [kHz]';
% Create bpskFc
app.bpskFc = uiknob(app.BPSKTab, 'continuous');
app.bpskFc.Limits = [1 10];
app.bpskFc.ValueChangingFcn = createCallbackFcn(app,
@bpskFcValueChanging, true);
app.bpskFc.Position = [148 129 60 60];
app.bpskFc.Value = 1;
% Create FskHzSpinnerLabel_3
app.FskHzSpinnerLabel_3 = uilabel(app.BPSKTab);
app.FskHzSpinnerLabel_3.HorizontalAlignment = 'right';
app.FskHzSpinnerLabel_3.FontWeight = 'bold';
app.FskHzSpinnerLabel_3.Position = [191 20 56 22];
app.FskHzSpinnerLabel_3.Text = 'Fs [kHz]:';
% Create bpskFs
app.bpskFs = uispinner(app.BPSKTab);
app.bpskFs.Step = 10;
app.bpskFs.Limits = [10 Inf];
app.bpskFs.ValueDisplayFormat = '%.2f';
app.bpskFs.ValueChangedFcn = createCallbackFcn(app,
@bpskFsValueChanged, true);
app.bpskFs.HorizontalAlignment = 'center';
app.bpskFs.FontWeight = 'bold';
app.bpskFs.Position = [262 20 100 22];
app.bpskFs.Value = 10;
% Create QPSKTab
app.QPSKTab = uitab(app.TabGroup2);
app.QPSKTab.Title = 'QPSK';
% Create qpskFcLabel
app.qpskFcLabel = uilabel(app.QPSKTab);
app.qpskFcLabel.HorizontalAlignment = 'center';
app.qpskFcLabel.FontWeight = 'bold';
app.qpskFcLabel.Position = [217 80 120 22];
app.qpskFcLabel.Text = 'Fc = 1.0 [kHz]';
% Create qpskFc
app.qpskFc = uiknob(app.QPSKTab, 'continuous');

```

```

app.qpskFc.Limits = [1 10];
    app.qpskFc.ValueChangingFcn = createCallbackFcn (app,
@qpskFcValueChanging, true);
    app.qpskFc.Position = [247 129 60 60];
    app.qpskFc.Value = 1;
    % Create FskHzSpinnerLabel_4
    app.FskHzSpinnerLabel_4 = uilabel(app.QPSKTab);
    app.FskHzSpinnerLabel_4.HorizontalAlignment = 'right';
    app.FskHzSpinnerLabel_4.FontWeight = 'bold';
    app.FskHzSpinnerLabel_4.Position = [191 20 56 22];
    app.FskHzSpinnerLabel_4.Text = 'Fs [kHz]:';
    % Create qpskFs
    app.qpskFs = uispinner(app.QPSKTab);
    app.qpskFs.Step = 10;
    app.qpskFs.Limits = [10 Inf];
    app.qpskFs.ValueDisplayFormat = '%.2f';
    app.qpskFs.ValueChangedFcn = createCallbackFcn(app,
@qpskFsValueChanged, true);
    app.qpskFs.HorizontalAlignment = 'center';
    app.qpskFs.FontWeight = 'bold';
    app.qpskFs.Position = [262 20 100 22];
    app.qpskFs.Value = 10;
    % Create QAMTab
    app.QAMTab = uitab(app.TabGroup2);
    app.QAMTab.Title = 'QAM';
    % Create qamFcLabel
    app.qamFcLabel = uilabel(app.QAMTab);
    app.qamFcLabel.HorizontalAlignment = 'center';
    app.qamFcLabel.FontWeight = 'bold';
    app.qamFcLabel.Position = [118 80 120 22];
    app.qamFcLabel.Text = 'Fc = 1.0 [kHz]';
    % Create qamFc
    app.qamFc = uiknob(app.QAMTab, 'continuous');
    app.qamFc.Limits = [1 10];
    app.qamFc.ValueChangingFcn = createCallbackFcn(app,
@qamFcValueChanging, true);
    app.qamFc.Position = [148 129 60 60];
    app.qamFc.Value = 1;
    % Create qamFs
    app.qamFs = uispinner(app.QAMTab);
    app.qamFs.Step = 10;
    app.qamFs.Limits = [10 Inf];
    app.qamFs.ValueDisplayFormat = '%.2f';
    app.qamFs.ValueChangedFcn = createCallbackFcn(app,
@qamFsValueChanged, true);
    app.qamFs.HorizontalAlignment = 'center';
    app.qamFs.FontWeight = 'bold';
    app.qamFs.Position = [262 21 100 22];
    app.qamFs.Value = 10;
    % Create MQAMKnobLabel
    app.MQAMKnobLabel = uilabel(app.QAMTab);
    app.MQAMKnobLabel.HorizontalAlignment = 'center';
    app.MQAMKnobLabel.FontWeight = 'bold';

```

```

app.MQAMKnobLabel.Position = [368 80 47 22];
app.MQAMKnobLabel.Text = 'M-QAM';
% Create qamM
app.qamM = uiknob(app.QAMTab, 'discrete');
app.qamM.Items = {'2', '4', '16', '32'};
app.qamM.ValueChangedFcn = createCallbackFcn(app,
@qamMValueChanged, true);
app.qamM.Position = [360 125 60 60];
app.qamM.Value = '4';
% Create FskHzSpinnerLabel_5
app.FskHzSpinnerLabel_5 = uilabel(app.QAMTab);
app.FskHzSpinnerLabel_5.HorizontalAlignment = 'right';
app.FskHzSpinnerLabel_5.FontWeight = 'bold';
app.FskHzSpinnerLabel_5.Position = [197 21 56 22];
app.FskHzSpinnerLabel_5.Text = 'Fs [kHz]';
% Create UIAxes
app.UIAxes = uiaxes(app.ConfiguracionesTab);
title(app.UIAxes, 'Vista Previa')
xlabel(app.UIAxes, 'Tiempo [mseg]')
ylabel(app.UIAxes, 'Amplitud')
app.UIAxes.Box = 'on';
app.UIAxes.XGrid = 'on';
app.UIAxes.YGrid = 'on';
app.UIAxes.Position = [181 16 450 200];
% Create ModulacinPanel
app.ModulacinPanel = uipanel(app.ConfiguracionesTab);
app.ModulacinPanel.Title = 'Modulación: ';
app.ModulacinPanel.FontWeight = 'bold';
app.ModulacinPanel.Position = [11 6 160 210];
% Create KnobLabel
app.KnobLabel = uilabel(app.ModulacinPanel);
app.KnobLabel.HorizontalAlignment = 'center';
app.KnobLabel.Position = [68 43 25 22];
app.KnobLabel.Text = '';
% Create Knob
app.Knob = uiknob(app.ModulacinPanel, 'discrete');
app.Knob.Items = {'ASK', 'FSK', 'BPSK', 'QPSK', 'QAM'};
app.Knob.ValueChangedFcn = createCallbackFcn(app,
@KnobValueChanged, true);
app.Knob.FontWeight = 'bold';
app.Knob.Position = [49 80 60 60];
app.Knob.Value = 'ASK';
% Create Apply
app.Apply = uibutton(app.ModulacinPanel, 'state');
app.Apply.ValueChangedFcn = createCallbackFcn(app,
@ApplyValueChanged, true);
app.Apply.Text = 'Seleccionar';
app.Apply.FontWeight = 'bold';
app.Apply.Position = [29 19 100 22];
% Create SensoresTab
app.SensoresTab = uitab(app.TabGroup);
app.SensoresTab.Title = 'Sensores';
% Create SERVIDORTCPIPPanel

```

```

app.SERVIDORTCPIPPanel = uipanel (app.SensoresTab);
    app.SERVIDORTCPIPPanel.Title = 'SERVIDOR TCP/IP: ';
    app.SERVIDORTCPIPPanel.FontWeight = 'bold';
    app.SERVIDORTCPIPPanel.Position = [11 386 620 60];
    % Create IPLabel
    app.IPLabel = uilabel(app.SERVIDORTCPIPPanel);
    app.IPLabel.HorizontalAlignment = 'right';
    app.IPLabel.Position = [11 8 25 22];
    app.IPLabel.Text = 'IP: ';
    % Create tcpipIP
    app.tcpipIP = uieditfield(app.SERVIDORTCPIPPanel, 'text');
    app.tcpipIP.HorizontalAlignment = 'center';
    app.tcpipIP.Position = [41 8 100 22];
    app.tcpipIP.Value = '0.0.0.0';
    % Create PuertoEditFieldLabel
    app.PuertoEditFieldLabel = uilabel(app.SERVIDORTCPIPPanel);
    app.PuertoEditFieldLabel.HorizontalAlignment = 'right';
    app.PuertoEditFieldLabel.Position = [181 8 44 22];
    app.PuertoEditFieldLabel.Text = 'Puerto: ';
    % Create tcpipPort
    app.tcpipPort = uieditfield(app.SERVIDORTCPIPPanel,
'numeric');
    app.tcpipPort.Limits = [0 Inf];
    app.tcpipPort.ValueDisplayFormat = '%.0f';
    app.tcpipPort.HorizontalAlignment = 'center';
    app.tcpipPort.Position = [231 8 69 22];
    app.tcpipPort.Value = 30000;
    % Create tcpipBtn
    app.tcpipBtn = uibutton(app.SERVIDORTCPIPPanel, 'state');
    app.tcpipBtn.ValueChangedFcn = createCallbackFcn(app,
@tcpipBtnValueChanged, true);
    app.tcpipBtn.Enable = 'off';
    app.tcpipBtn.Text = 'Conectar';
    app.tcpipBtn.Position = [511 8 100 22];
    % Create TiempoSpinnerLabel
    app.TiempoSpinnerLabel = uilabel(app.SERVIDORTCPIPPanel);
    app.TiempoSpinnerLabel.HorizontalAlignment = 'right';
    app.TiempoSpinnerLabel.Position = [339 8 48 22];
    app.TiempoSpinnerLabel.Text = 'Tiempo: ';
    % Create tcpipTime
    app.tcpipTime = uispinner(app.SERVIDORTCPIPPanel);
    app.tcpipTime.Step = 0.05;
    app.tcpipTime.Limits = [0.05 1];
    app.tcpipTime.HorizontalAlignment = 'center';
    app.tcpipTime.Position = [399 8 73 22];
    app.tcpipTime.Value = 0.25;
    % Create DISPLAYSPanel
    app.DISPLAYSPanel = uipanel(app.SensoresTab);
    app.DISPLAYSPanel.Title = 'DISPLAYS: ';
    app.DISPLAYSPanel.FontWeight = 'bold';
    app.DISPLAYSPanel.Position = [10 6 621 371];
    % Create sensAxes
    app.sensAxes = uiaxes(app.DISPLAYSPanel);

```

```

Title (app.sensAxes, 'Audio Recibido')
Xlabel (app.sensAxes, 'Tiempo [seg]')
ylabel(app.sensAxes, 'Amplitud')
app.sensAxes.Box = 'on';
app.sensAxes.XGrid = 'on';
app.sensAxes.YGrid = 'on';
app.sensAxes.Position = [151 6 460 155];
% Create TextEditFieldLabel
app.TextEditFieldLabel = uilabel(app.DISPLAYSPanel);
app.TextEditFieldLabel.HorizontalAlignment = 'right';
app.TextEditFieldLabel.Position = [181 179 38 22];
app.TextEditFieldLabel.Text = 'Texto: ';
% Create sensTxt
app.sensTxt = uieditfield(app.DISPLAYSPanel, 'text');
app.sensTxt.Editable = 'off';
app.sensTxt.HorizontalAlignment = 'center';
app.sensTxt.Position = [231 179 359 22];
app.sensTxt.Value = 'Texto de Prueba';
% Create TemperaturaLabel
app.TemperaturaLabel = uilabel(app.DISPLAYSPanel);
app.TemperaturaLabel.HorizontalAlignment = 'center';
app.TemperaturaLabel.Position = [21 298 76 22];
app.TemperaturaLabel.Text = 'Temperatura: ';
% Create sensTemp
app.sensTemp = uigauge(app.DISPLAYSPanel, 'linear');
app.sensTemp.Orientation = 'vertical';
app.sensTemp.Position = [37 41 79 249];
% Create Label
app.Label = uilabel(app.DISPLAYSPanel);
app.Label.HorizontalAlignment = 'center';
app.Label.Position = [200 209 25 22];
app.Label.Text = '';
% Create sensPower
app.sensPower = uigauge(app.DISPLAYSPanel, 'circular');
app.sensPower.Position = [165 221 99 99];
% Create LED1Label
app.LED1Label = uilabel(app.DISPLAYSPanel);
app.LED1Label.HorizontalAlignment = 'center';
app.LED1Label.FontWeight = 'bold';
app.LED1Label.Position = [321 277 36 22];
app.LED1Label.Text = 'LED1';
% Create sensSW1
app.sensSW1 = uilamp(app.DISPLAYSPanel);
app.sensSW1.Position = [330 307 20 20];
% Create LED2Label
app.LED2Label = uilabel(app.DISPLAYSPanel);
app.LED2Label.HorizontalAlignment = 'center';
app.LED2Label.FontWeight = 'bold';
app.LED2Label.Position = [410 277 36 22];
app.LED2Label.Text = 'LED2';
% Create sensSW2
app.sensSW2 = uilamp(app.DISPLAYSPanel);
app.sensSW2.Position = [419 307 20 20];

```

```

app.sensSW2.Color = [1 0 0];
% Create LED3Label
app.LED3Label = uilabel(app.DISPLAYSPanel);
app.LED3Label.HorizontalAlignment = 'center';
app.LED3Label.FontWeight = 'bold';
app.LED3Label.Position = [512 277 36 22];
app.LED3Label.Text = 'LED3';
% Create sensSW3
app.sensSW3 = uilamp(app.DISPLAYSPanel);
app.sensSW3.Position = [521 307 20 20];
app.sensSW3.Color = [1 1 0];
% Create TiempoLabel
app.TiempoLabel = uilabel(app.DISPLAYSPanel);
app.TiempoLabel.HorizontalAlignment = 'center';
app.TiempoLabel.FontSize = 16;
app.TiempoLabel.FontWeight = 'bold';
app.TiempoLabel.Position = [340 211 154 33];
app.TiempoLabel.Text = '---:---:---';
% Create PotencialLabel
app.PotencialLabel = uilabel(app.DISPLAYSPanel);
app.PotencialLabel.Position = [187 319 56 22];
app.PotencialLabel.Text = 'Potencia ';
% Create TiempodeTransmisionLabel
app.TiempodeTransmisionLabel = uilabel(app.DISPLAYSPanel);
app.TiempodeTransmisionLabel.Position = [349 243 140 22];
app.TiempodeTransmisionLabel.Text = 'Tiempo de Transmision';
end
end
methods (Access = public)
% Construct app
function app = receptor
    % Create and configure components
    createComponents(app)
    % Register the app with App Designer
    registerApp(app, app.ReceptorUIFigure)
    % Execute the startup function
    runStartupFcn(app, @startupFcn)
    if nargin == 0
        clear app
    end
end
% Code that executes before app deletion
function delete(app)
    % Delete UIFigure when app is deleted
    delete(app.ReceptorUIFigure)
end
end
end

```

Anexo 4

Formato de la encuesta



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA DE INGENIERÍA EN TELEINFORMÁTICA.

La siguiente encuesta está orientada al proyecto de Tesis que tiene como título “IMPLEMENTACIÓN EN SOFTWARE DE LOS SISTEMAS DE MODULACIONES DIGITALES”, para recopilar información que pretende medir la fiabilidad de respuesta y la facilidad de realizar prácticas dentro del aula de clases con el software MATLAB, datos que se obtendrán a través de la práctica que fue realizada por los estudiantes del 8vo Semestre de la Carrera de Ingeniería en Teleinformática.

Cada pregunta tiene una ponderación que puede ir de lo más aceptable a lo menos aceptable, dependiendo de la pregunta, se les recomienda que lean detenidamente y elijan la opción que esté acorde a su respuesta.

Por Favor marcar con una “X” en el casillero que se encuentra en cada opción de respuesta.

- 1. ¿Las prácticas dentro del laboratorio sirven de apoyo para el proceso de enseñanza de los estudiantes?**

Totalmente de acuerdo.	
De acuerdo.	
En Desacuerdo.	
Totalmente en desacuerdo	

- 2. ¿Las prácticas dentro del laboratorio, permiten reducir tiempo, costos, tanto a los estudiantes como a docentes?**

Totalmente de acuerdo.	
De acuerdo.	
En Desacuerdo.	
Totalmente en desacuerdo	

- 3. ¿Existe necesidad de que los estudiantes dominen el uso de software para la elaboración de prácticas?**

Totalmente de acuerdo.	
De acuerdo.	
En Desacuerdo.	
Totalmente en desacuerdo	

- 4. ¿El uso del software MATLAB es sencillo y más práctico que el uso de otro hardware?**

Totalmente de acuerdo.	
De acuerdo.	
En Desacuerdo.	
Totalmente en desacuerdo	

5. **¿El uso de la herramienta App Designer del software MATLAB, incluye componentes necesarios para el desarrollo de un sistema de modulaciones digitales?**

Totalmente de acuerdo.	
De acuerdo.	
En Desacuerdo.	
Totalmente en desacuerdo	

6. **¿Posee conocimientos necesarios para utilizar la aplicación de los sistemas de modulaciones digitales realizadas en la herramienta App Designer del Software Matlab?**

Totalmente de acuerdo.	
De acuerdo.	
En Desacuerdo.	
Totalmente en desacuerdo	

7. **¿Cree usted que la aplicación desarrollada en la herramienta App Designer del software MATLAB, es de fácil manejo y permite comprender los fundamentos teóricos?**

Totalmente de acuerdo.	
De acuerdo.	
En Desacuerdo.	
Totalmente en desacuerdo	

8. **¿Considera usted que dentro de la implementación de los sistemas de modulaciones digitales en la herramienta App Designer del software MATLAB, redujo el tiempo para la adquisición de conocimiento de las clases?**

Totalmente de acuerdo	
De acuerdo	
En desacuerdo.	
Totalmente en desacuerdo	

9. **¿Cree usted necesario la implementación de nuevos programas para realizar sistemas que utilicen señales digitales?**

Totalmente de acuerdo.	
De acuerdo.	
En desacuerdo.	
Totalmente en desacuerdo	

10. **¿La proporción entre clases prácticas y teóricas son las correctas para que los estudiantes puedan adquirir nuevos conocimientos dentro de la catedra?**

Totalmente de acuerdo.	
De acuerdo.	
En desacuerdo.	
Totalmente en desacuerdo	

Anexo 5

Fotos de la implementación del sistema de modulaciones digitales



Implementación del Sistema. Información tomada directamente por el autor. Elaborado por el autor



Estudiantes visualizando el funcionamiento del sistema propuesto. Información tomada directamente por el autor. Elaborado por el autor.

Bibliografía

- Real Academia de Ingeniería de España. (2014). Sitio web. Teoría de la Señal. [http://www.raing.es/es/acad-micos/excmo-sr-d-ram-n-agust-comes/Teoría de la Señal.html](http://www.raing.es/es/acad-micos/excmo-sr-d-ram-n-agust-comes/Teoria%20de%20la%20Se%C3%B1al.html).
- Texto científicos. (2005). Sitio web. Redes/Modulación PSK. <https://www.textoscientificos.com/redes/modulacion/psk.html>
- Álvarez-Álvarez, C. (2015). Artículo. Teoría frente a práctica educativa: algunos problemas y propuestas de solución. *Perfiles educativos*, 37(148). http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0185-26982015000200011.pdf
- Amador, M. G. (2009). Sitio web. La entrevista en investigación. <http://manuelgalan.blogspot.com/2009/05/la-entrevista-en-investigacion.html>
- Arias, F. (2016). PDF. Importancia de la metodología de la investigación en la educación universitaria. <http://otrasvoceseneducacion.org/archivos/77516.pdf>
- Asamblea Nacional Constituyente. (2008). PDF. Constitución de la República del Ecuador. <https://www.wipo.int/edocs/lexdocs/laws/es/ec/ec030es.pdf>
- Asamblea Nacional Constituyente. (2018). PDF. Ley Orgánica Reformatoria a la Ley Orgánica de Educación Superior. <https://educacion.gob.ec/wp-content/uploads/downloads/2015/09/Ley-Organica-Reformatoria-a-la-Ley-Organica-de-Educacion-Intercultural-LOEI.pdf>
- Casas Anguita, J., Repullo Labrador, J., & Donado Campos, J. (2003). Artículo. La encuesta como técnica de investigación. Elaboración de cuestionarios y tratamiento estadístico de los datos (I). <https://medes.com/publication/9654.pdf>
- Cataldi, Z., Lage, F., & Dominighini, C. (2013). Artículo. Fundamentos para el uso de simulaciones en la enseñanza. *Revista Informática Educativa y Medio Audiovisuales*, 10(17), 8-16.pdf. <http://laboratorios.fi.uba.ar/lie/Revista/Articulos/101017/A2mar2013.pdf>
- CCM.Net. (2018). Sitio web.TCP/IP. <https://es.ccm.net/contents/282-tcp-ip.html>.
- CEP. (2016). Libro pag. 224.Guardia Civil. Escala de Cabos y Guardias (Vol. II). Madrid: CEP.pdf.
- Consejo de Educación Superior. (2017). Sitio web. Universidades y escuelas politécnicas. http://www.ces.gob.ec/index.php?option=com_sobipro&sid=69:Universidades-y-escuelas-politecnicas&Itemid=0.html.
- Crispín, M., Gómez, T., Ramírez, J., & Ulloa, J. (2012). PDF. Guía del docente para el desarrollo de competencias. (G. Ulloa, Ed.) Ciudad de México: Universidad Iberoamericana CM. http://www.iberomx.mx/formaciondeprofesores/Apoyos%20generales/Guia_docente_desarrollo_competencias.pdf
- Durango, P. (2015). Repositorio UNAL. Las prácticas de laboratorio como una estrategia didáctica alternativa para desarrollar las competencias básicas en el proceso enseñanza-aprendizaje de la química <http://www.bdigital.unal.edu.co/49497/1/43905291.2015.pdf>
- EcuRed. (2012). Sitio web. Sistema de telecomunicaciones. https://www.ecured.cu/Sistema_de_telecomunicaciones#Servicios_de_telecomunicaciones.html.
- Gaitán, A., Restrepo, A., & Cáceres, M. (2013). Artículo. Estrategias de acompañamiento para el mejoramiento de los programas de formación para el ejercicio de la docencia

- ue desarrollan las instituciones de educación superior y escuelas normales superiores oficiales. https://www.mineduacion.gov.co/1759/articles-345822_ANEXO_21.pdf
- Garcés, E., Garcés, E., & Alcívar, O. (2016). Artículo. Las tecnologías de la información en el cambio de la educación superior en el siglo XXI: reflexiones para la práctica. *Revista Universidad y Sociedad*, 8(4). doi: ISSN 2218-3620
- Gargiulo, S., & Gómez, M. (2016). PDF. Algunas experiencias con simuladores educativos. Parte I. Docentes en línea. Recuperado de https://www.researchgate.net/publication/327977646_Algunas_experiencias_con_simuladores_educativos_Parte_I_Diciembre_2016.html.
- Gómez, J. (2016). Repositorio UPNA. Desarrollo de una interfaz gráfica de usuario para el control de analizadores de espectros ópticos mediante Matlab. Tesis Grado en Ingeniería en Tecnologías Industriales. <https://academica-e.unavarra.es/bitstream/handle/2454/21691/Desarrollo%20de%20una%20interfaz%20gr%C3%ADfica%20de%20usuario%20para%20el%20control%20de%20analizadores%20de%20espectros%20%2B%C2%A6pticos%20mediante%20Matlab.pdf?sequence=1&isAllowed=y.pdf>.
- Gomez, M. (2013). Artículo. La importancia de los laboratorios en la enseñanza de la ingeniería. *Ingenierías* (61). http://eprints.uanl.mx/10531/1/61_editorial.pdf
- Guanoluisa, L. (2012). Repositorio EPN. Implementación de sistemas de telecomunicaciones para las redes de datos de las empresas públicas y privadas del país. Tesis. <http://bibdigital.epn.edu.ec/bitstream/15000/4470/1/CD-4085.pdf>
- Gunipa-Pérez, M. (2012). Artículo. Elementos del proceso de comunicación que orientan el discurso de los profesores universitarios. *VivatAcademia*, 118, 33-60. <https://www.redalyc.org/pdf/5257/525752945003.pdf>
- Hernando, M. (2012). Sitio web. MatLab y Robótica Introducción a Matlab y la ToolBox de Robótica. https://www.academia.edu/6553163/MatLab_y_Rob%C3%B3tica_Introducci%C3%B3n_a_Matlab_y_la_ToolBox_de_Rob%C3%B3tica.pdf
- IBM. (2017). Sitio web. Protocolos TCP/IP. https://www.ibm.com/support/knowledgecenter/es/ssw_aix_72/com.ibm.aix.networkcomm/tcpip_protocols.html
- Lanuza, F., Rizo, M., & Saavedra, L. (2018). Artículo. Uso y aplicación de las TIC en el proceso de enseñanza- aprendizaje. *Revista Científica de FAREM-Estelí*, 7(25). <https://www.lamjol.info/index.php/FAREM/article/view/5667/5380.html>.
- Laoerden, E. (2012). Repositorio UPM. Descripción, comparación y ejemplos de uso de las funciones de la toolbox de procesamiento digital de imágenes de MATLAB. http://oa.upm.es/14016/2/PFC_EDUARDO_LAORDEN_FITER_B.pdf
- López, A., & Tamayo, Ó. (2012). Artículo. Las prácticas de laboratorio en la enseñanza de las ciencias naturales. *Revista Latinoamericana de Estudios Educativos*, 8(1), 145-166. <https://www.redalyc.org/pdf/1341/134129256008.pdf>
- López, S. (2011). Artículo. El aprendizaje a través de la participación del estudiante en actividades prácticas. Barcelona. Recuperado de <http://www.cite2011.com/Comunicaciones/Escuela/044.pdf>
- Maquilón, J. (2011). PDF. La formación del profesorado en el siglo XXI. Propuestas ante los cambios económicos, sociales y culturales (1era ed.). (A. U. Profesorado, Ed.) Universidad de Murcia, Servicio de Publicaciones. http://aufop.com/aufop/uploaded_files/revistas/14189331764.pdf

- Matas, A. (2018). Artículo. Diseño del formato de escalas tipo Likert: un estado de la cuestión. *Revista Electrónica de Investigación*, 20 (1), 38.47.pdf.
- MathWorks. (2018). Sitio web. MATLAB. https://es.mathworks.com/help/matlab/learn_matlab/scripts-and-functions.html
- Mosquera, D., & Sacoto, E. (2012). Repositorio UPS. Diseño de un laboratorio para el desarrollo de prácticas en las asignaturas de formación profesional para la carrera de ingeniería industrial de la Universidad Politécnica Salesiana Sede Cuenca. <https://dspace.ups.edu.ec/bitstream/123456789/5204/1/UPS-CT002745.pdf>
- Niño, J., Garcia, J., Alvarado Juan, Barragán, G., Urrea, G., & Hazbón, O. (2013). Artículo. Del aula a la libertad. La Importancia de los laboratorios en la formación del ingeniero. caso de estudio: Ingeniería aeronáutica. Cartagena. <https://www.acofipapers.org/index.php/acofipapers/2013/paper/viewFile/502/145>
- Pillou, J.-F. (2017). PDF. Transmisión de datos - Transmisión digital. <https://es.ccm.net/contents/684-transmission-de-datos-transmission-digital.pdf>
- Sánchez, J. (2013). PDF. Introducción a Matlab. <http://blog.utp.edu.co/jnsanchez/files/2011/03/Matlab-Introducci%C3%B3n-clase16.pdf>
- Schaull, S. (2011). PDF. Software: concepto, tipos y usos. *Ing. USBMed*, 2(2), 6-9. <http://web.usbmed.edu.co/usbmed/fing/v2n2/v2n2a1.pdf>
- Secretaría de la Educación Superior, Ciencia, Tecnología e Innovación. (sf). Sitio web. SNIIESE. Información actualizada sobre la educación superior del Ecuador. <https://infoeducacionsuperior.gob.ec/#/>
- Soubal, S. (2012). Artículo. La gestión del aprendizaje. *Polis. Revista Latinoamericana*. <https://journals.openedition.org/polis/2955>
- Tamayo, M., & Tamayo. (2012). Libro pag. 180. *El Proceso De La Investigacion Cientifica*. Mexico: Limusa S.A. http://biblioteca.unach.edu.ec/opac_css/index.php?lvl=notice_display&id=12950
- Torres, F. (2016). Repositorio UPEC. La importancia de los laboratorios en el proceso de aprendizaje. *InformaticaUPEC* (64), 2-6.pdf. http://www.upec.edu.ec/index.php?option=com_docman&task=doc_download&gid=4033&Itemid=86. pdf.