



FACULTAD DE INGENIERIAS Y CIENCIAS AGROPECUARIAS

DISEÑO E IMPLEMENTACIÓN DE PROTOTIPO DE SIMULADOR DE  
OFDM UTILIZANDO LA HERRAMIENTA MATLAB

AUTOR

FERNANDO XAVIER MOLINA LUCAS

AÑO

2018



FACULTAD DE INGENIERIAS Y CIENCIAS AGROPECUARIAS

DISEÑO E IMPLEMENTACIÓN DE PROTOTIPO DE SIMULADOR DE OFDM  
UTILIZANDO LA HERRAMIENTA MATLAB.

Trabajo de titulación presentado en conformidad con los requisitos establecidos  
para optar por el título de Ingeniero en Redes y Telecomunicaciones

Profesor Guía  
MSc. Iván Ricardo Sánchez Salazar

Autor  
Fernando Xavier Molina Lucas

Año  
2018

### **Declaración Profesor Guía**

"Declaro haber dirigido el trabajo, DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SIMULACIÓN DE OFDM UTILIZANDO LA HERRAMIENTA MATLAB, a través de reuniones periódicas con el estudiante Fernando Molina, en el Semestre 2018-1, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

---

Iván Ricardo Sánchez Salazar

Magister en Calidad Seguridad Y Ambiente

CC: 1803456142

### **Declaración Del Profesor Corrector**

“Declaro Haber revisado este trabajo, DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SIMULACIÓN DE OFDM UTILIZANDO LA HERRAMIENTA MATLAB del estudiante Fernando Xavier Molina Lucas, en el semestre 2018-1, dando cumplimiento a todas las disposiciones vigentes que regulan los trabajos de titulación “.

---

Carlos Marcelo Molina Colcha

Magister en Gestión de las Comunicaciones y Tecnologías de la Información.

CC: 1709624215

## **DECLARACION DE AUTORIA DEL ESTUDIANTE**

“Declaro que este trabajo es original, de mi autoría, que se ha citado las fuentes correspondientes y que en su ejecución se ha respetado las disposiciones legales que protegen los derechos de autor vigentes.”

---

Fernando Xavier Molina Lucas

CI: 1720799319

## **AGRADECIMIENTOS**

En el siguiente trabajo de tesis quiero agradecer principalmente a Dios, por acompañarme y guiarme en todo momento, por ponerme en un seno familiar único, llenar mi vida de momentos de tristeza, dolor y felicidad, mismos que me han permitido llegar hasta donde estoy en este momento, por ser mi fortaleza y mi pilar fundamental para recorrer mi camino.

A mis padres Fernando Molina y Letty Lucas por apoyarme en todo momento, enseñarme todos los valores morales y éticos que me permiten que sea un ser humano único e irrepetible y darme el mejor de los ejemplos de lucha y esfuerzo.

## **DEDICATORIA**

A mis padres, por ser ejemplo de liderazgo, disciplina, honestidad, lealtad y porque supieron hacer de mi un gran hombre.

A mi hija, por ser el motor que impulsa mis días, sobre todo por ser la mujer que me enseñó a amar incondicionalmente, permitiéndome ser su ejemplo a seguir.

A todos y todas las personas que estuvieron, están, y seguirán conmigo en mi camino, compartiendo malas y buenas experiencias, haciendo de este camino un lindo y grato vivir, gracias

## RESUMEN

El presente proyecto abarca el estudio, diseño e implementación de un prototipo de simulador, apoyado en un entorno de desarrollo de un sistema matemático (MATLAB), que permita explicar de manera didáctica el funcionamiento de la Multiplexación por División de Frecuencias Ortogonal (OFDM). El proyecto se ha conformado en seis capítulos, los cuales se distribuirán de la siguiente manera

Capítulo I: Se identifica y describe el problema objeto de estudio, así como los objetivos, alcance y justificación de la investigación.

Capítulo II: En este capítulo se analiza el sistema OFDM, recopilando información sobre éste y sus componentes. Además, se definirán los parámetros y variables que serán utilizados en el proceso de modulación OFDM.

Capítulo III: se diseña el diagrama de flujo para el desarrollo e implementación del software que permitirá visualizar las etapas de modulación OFDM.

Capítulo IV: Consiste en el desarrollo del prototipo de simulador OFDM en una herramienta informática. En este capítulo se configurará el prototipo mediante la herramienta informática de simulación.

Capítulo V: Finalmente, en este capítulo se realizan pruebas de funcionamiento del prototipo implementado, así como la verificación del correcto funcionamiento del simulador de OFDM "interfaz del usuario", además se propondrá una guía de laboratorio para la Universidad de Las Américas.

## **ABSTRACT**

The present project covers the study, design and implementation of a prototype simulator, supported in a development environment and mathematical software tool (MATLAB), which allows to explain the operation of Orthogonal Frequency Division Multiplexing (OFDM). The project has been divided into six chapters, which will be distributed as follows:

Chapter I: The problem object of study is identified and described as well as the objectives, scope and justification of the investigation

Chapter II: In this chapter OFDM is analyzed, collecting information about it and its components. Also, the parameters and variables that will be used in the OFDM modulation process are defined.

Chapter III: The design of the flow diagram for the development and implementation of the software that will allow visualizing the OFDM modulation stages is designed.

Chapter IV: It consists of the development of the OFDM simulator prototype in a computer tool. In this chapter, the prototype will be configured using the computer simulation tool.

Chapter V: Finally, in this chapter performance tests of the implemented prototype are carried out, as well as the verification of the correct functioning of the OFDM simulator "user interface", and a laboratory guide for the University of the Americas will be proposed.

# ÍNDICE

1. CAPITULO I. INTRODUCCIÓN I.....	1
1.1. Alcance .....	2
1.2. Justificación .....	2
1.3. Objetivos .....	2
1.3.1 Objetivo General.....	2
1.3.2. Objetivos específicos .....	3
2. CAPÍTULO II. MARCO TEÓRICO .....	3
2.1. Antecedentes Investigativos.....	3
2.2. Fundamentación Teórica.....	4
2.2.1. La Comunicación.....	4
2.2.2. Sistemas de comunicación.....	5
2.2.3. Modulaciones.....	5
2.2.4. Multiplexación por División de Frecuencias Ortogonales (OFDM)..	7
2.2.5. <i>Parámetros de las OFDM.</i> .....	9
2.2.6. <i>Aplicaciones.</i> .....	10
2.2.7. <i>Sistemas de Transmisión OFDM.</i> .....	10
2.2.8. <i>Transmisión con una portadora.</i> .....	10
2.2.9. <i>Transmisión con múltiples portadoras.</i> .....	11
2.2.10. <i>Transmisión OFDM.</i> .....	13
2.2.11. <i>Transmisión FMT.</i> .....	14
2.2.12. MATLAB. ....	15
2.2.13. <i>Entorno de desarrollo.</i> .....	16
2.2.14. <i>Entorno de Simulación.</i> .....	17
2.2.15. <i>Entorno de Interfaz de Usuario.</i> .....	17
3. CAPÍTULO III. COMPONENTES DEL OFDM.....	17
3.1 Análisis de Sistemas OFDM.....	17
3.2. Proceso de transmisión.....	19

3.2.1	Generación de la imagen.....	19
3.2.2	Codificador. ....	20
3.2.3	Inserción de los pilotos. ....	24
3.2.4	IFFT. ....	26
3.2.5	Inserción de intervalo de guarda.....	29
3.2.6	Conversor Paralelo-Serial.....	32
3.2.7	D/A.....	32
3.2.8	Transmisor.....	33
3.2.9	Canal. ....	33
<b>4.</b>	<b>CAPITULO IV. DIAGRAMA DE FLUJO.....</b>	<b>35</b>
4.1.	Diagrama OFDM generalizado.....	35
4.2.	Proceso de Transmisión. ....	36
4.3.	Generación de la imagen de bit. ....	36
4.4.	Conversor S/P. ....	37
4.5.	Modulación 16 QAM. ....	39
4.6.	Piloto.....	40
4.7.	IFFT. ....	41
4.8.	Prefijo cíclico.....	43
4.9.	Conversor P/S. ....	44
4.10.	Transmisión. ....	45
4.11.	Canal.....	46
4.12.	Sistema de transmisión.....	47
<b>5.</b>	<b>CAPÍTULO V. SIMULADOR OFDM.....</b>	<b>48</b>
5.1.	Código de inicio de carga de la ventana.....	50
5.2	Generación del código de imagen.....	50
5.3	Generación de la imagen. ....	51
5.4.	Conversión de serial a paralelo. ....	51
5.5.	Modulación 16 QAM.....	52
5.6.	Inserción de pilotos.....	52

5.7. Modulación IFFT.....	53
5.8. Inserción de la guarda.....	54
5.9. Conversión paralela a serial.....	54
5.10. Transmisión.....	55
5.11. Canal.....	55
5.12. Elementos visuales.....	56
<b>6. CAPÍTULO VI. PRUEBAS DE FUNCIONAMIENTO.....</b>	<b>58</b>
6.1. Datos de prueba.....	58
6.2. Generación de imágenes.....	58
6.3. Conversión de serial a paralelo.....	59
6.4. Modulación 16 QAM.....	60
6.5. Inserción de pilotos.....	61
6.6. Aplicación de IFFT.....	62
6.7. Aplicación de guarda.....	66
6.8. Conversión paralelo a serial.....	67
6.9. Transmisión.....	68
6.10. Canal.....	68
<b>7. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>70</b>
7.1 Conclusiones.....	70
7.2. Recomendaciones.....	71
<b>REFERENCIAS.....</b>	<b>72</b>
<b>ANEXOS.....</b>	<b>75</b>

## ÍNDICE DE FIGURAS

<i>Figura 1.</i> Esquema de Sistema de Comunicación.....	5
<i>Figura 2.</i> Clasificación de las modulaciones analógicas y digitales.....	6
<i>Figura 3.</i> Composición de la señal OFMD.....	9
<i>Figura 4.</i> Diagrama de bloque de un sistema OFDM .....	10
<i>Figura 5.</i> Diagrama de transmisión con múltiples portadoras .....	11
<i>Figura 6.</i> Sistema multicanal con detalle .....	12
<i>Figura 7.</i> Ejemplo de la portadora Nyquist .....	13
<i>Figura 8.</i> Aplicación de la transformada de Fourier.....	14
<i>Figura 9.</i> Sistemas de modulación FMT .....	15
<i>Figura 10.</i> Ejemplo del diseño de un circuito en MATLAB .....	16
<i>Figura 11.</i> Diagrama del OFDM .....	18
<i>Figura 12.</i> Grafica de la imagen binaria. ....	19
<i>Figura 13.</i> Símbolos de las imágenes de datos.....	20
<i>Figura 14.</i> Representación 3D de los símbolos.....	20
<i>Figura 15.</i> Mapeado base de la modulación 16QAM. ....	21
<i>Figura 16.</i> Mapa De Constelaciones .....	22
<i>Figura 17.</i> Visualización de la Constelación según MATLAB .....	23
<i>Figura 18.</i> Constelación de salida del ejemplo numérico. ....	24
<i>Figura 19.</i> Representación de la inserción de pilotos.....	25
<i>Figura 20.</i> Gráfica de la inserción de pilotos. ....	26
<i>Figura 21.</i> Tiempo y frecuencia de las portadoras IFFT.....	27
<i>Figura 22.</i> Aplicación de la IFFT del ejemplo citado utilizando MATLAB. ....	28
<i>Figura 23.</i> Aplicación del prefijo cíclico .....	29
<i>Figura 24.</i> Incorporación del prefijo cíclico del ejemplo citado. ....	31
<i>Figura 25.</i> Gráfica de la conversión Serial-Paralelo .....	32
<i>Figura 26.</i> Codificación de la señal .....	33
<i>Figura 27.</i> Representación de la señal OFDM realizado en MATLAB.....	33
<i>Figura 28.</i> Inclusión de ruido en la señal.....	34
<i>Figura 29.</i> Representación de inserción de ruido .....	34
<i>Figura 30.</i> Diagrama de modulación OFDM.....	35
<i>Figura 31.</i> Diagrama de flujo de la generación de la imagen de bit.....	37
<i>Figura 32.</i> Diagrama de flujo de la generación de la conversión S/P.....	38
<i>Figura 33.</i> Detalle de la conversión serial a paralelo.....	38
<i>Figura 34.</i> Diagrama de flujo del 16 QAM.....	39
<i>Figura 35.</i> Detalle de la modulación 16 QAM.....	40
<i>Figura 36.</i> Diagrama de flujo del 16 QAM .....	40
<i>Figura 37.</i> Detalle de la aplicación de los pilotos.....	41
<i>Figura 38.</i> Aplicación de la transformada rápida de Fourier inversa .....	41

<i>Figura 39.</i> Diagrama de flujo de la IFFT .....	42
<i>Figura 40.</i> Detalle de la aplicación del prefijo cíclico .....	43
<i>Figura 41.</i> Diagrama de flujo de la guarda .....	44
<i>Figura 42.</i> Diagrama de flujo de la conversión P/S. ....	45
<i>Figura 43.</i> Detalle de la conversión paralelo a serial.....	45
<i>Figura 44.</i> Diagrama de flujo de la transmisión. ....	46
<i>Figura 45.</i> Representación la transmisión. ....	46
<i>Figura 46.</i> Diagrama de flujo del canal.....	47
<i>Figura 47.</i> Representación de la inserción del ruido en el canal .....	47
<i>Figura 48.</i> Diagrama de bloque de la modulación 16 QAM y OFDM.....	48
<i>Figura 49.</i> Entorno de programación del MATLAB.....	48
<i>Figura 50.</i> Armado de la interfaz en GUIDE .....	49
<i>Figura 51.</i> Indicación del nombre de cada elemento.....	50
<i>Figura 52.</i> Bloque generador de código .....	50
<i>Figura 53.</i> Bloque de Imagen .....	51
<i>Figura 54.</i> Bloque conversión serial-paralelo .....	51
<i>Figura 55.</i> Bloque modulación 16 QAM.....	52
<i>Figura 56.</i> Bloque de la inserción del piloto.....	52
<i>Figura 57.</i> Bloque de modulación IFFT .....	53
<i>Figura 58.</i> Bloque de inserción de guarda.....	54
<i>Figura 59.</i> Bloque de conversión paralelo-serial. ....	54
<i>Figura 60.</i> Bloque de la transmisión .....	55
<i>Figura 61.</i> Bloque del canal.....	55
<i>Figura 62.</i> Sección de los elementos visuales.....	56
<i>Figura 63.</i> Diagrama de bloque del simulador.....	57
<i>Figura 64.</i> Estado de inicio del simulador.....	57

## ÍNDICE DE TABLAS

Tabla 1. <i>Reordenamiento de la imagen binaria por el conversor S/P.</i> .....	20
Tabla 2. <i>Modulación 16QAM por cada 4 bits.</i> .....	22
Tabla 3. <i>Tabla para codificación según modulación 16QAM</i> .....	23
Tabla 4. <i>Realización del mapeo de los símbolos.</i> .....	23
Tabla 5. <i>Imagen de salida de la modulación 16 QAM.</i> .....	24
Tabla 6. <i>Inserción de pilotos.</i> .....	25
Tabla 7. <i>Representación numérica de la inserción de pilotos</i> .....	26
Tabla 8. <i>Tabla de resultado de aplicación de IFFT</i> .....	27
Tabla 9. <i>Aplicación del prefijo cíclico.</i> .....	30
Tabla 10. <i>Parametros QPSK y 16 QAM.</i> .....	34
Tabla 11. <i>Listado de los elementos de interfaz utilizados</i> .....	49
Tabla 12. <i>Gráficas de ingreso de datos.</i> .....	58
Tabla 13. <i>Resultados de la conversión S/P.</i> .....	59
Tabla 14. <i>Constelaciones 16 QAM de las pruebas.</i> .....	61
Tabla 15. <i>Gráficas de inserción de pilotos.</i> .....	62
Tabla 16. <i>Tabla de aplicación de IFFT con variación de tiempo.</i> .....	63
Tabla 17. <i>Gráficas de aplicación IFFT con variación de la longitud</i> .....	63
Tabla 18. <i>Aplicación de IFFT con variación de las portadoras.</i> .....	64
Tabla 19. <i>Gráficas IFFT seleccionadas.</i> .....	65
Tabla 20. <i>Gráficas de la guarda.</i> .....	66
Tabla 21. <i>Gráficas de conversión P/S.</i> .....	67
Tabla 22. <i>Gráficas de la transmisión.</i> .....	68
Tabla 23. <i>Gráficas del paso por el canal.</i> .....	69

## 1. CAPITULO1. INTRODUCCIÓN I

La Universidad de las Américas (UDLA), con su programa de proyectos de investigación científica y tecnológica, contribuye a la solución de problemas específicos con un gran interés social. Para ello, se ha implementado una política de promoción continua de la investigación como parte del proceso educativo en todos los programas de grado y de postgrado en los últimos años, dando prioridad a la investigación en áreas científicas y sociales, creando departamentos de investigación, todo esto unido a la vinculación responsable con la sociedad, el desarrollo del talento humano mediante la labor docente y el desarrollo profesional; el propósito fundamental es la de solucionar problemáticas y suplir deficiencias o novedades que surgen por la propia necesidad

Es por esto que, mediante este proyecto, se busca solucionar una necesidad imperante en la UDLA, ya que muchas veces resulta difícil comprobar la teoría de los conocimientos obtenidos con la práctica. Es muy importante tener herramientas que permitan al estudiante y al docente sustentar las clases impartidas con ejemplos prácticos para preparar al estudiantado al campo laboral.

Para suplir esta necesidad en la carrera de Redes y Telecomunicaciones se propone el diseño de un simulador con codificación y presentación gráfica, analizando las más comunes herramientas computarizadas que generen una señal la cual pueda ser controlada y visualizada de una manera práctica y flexible. El objetivo de la comparación de diversos programas informáticos es la determinación de viabilidad de la selección de una como base del proyecto y crear una simulación adecuada de fácil entendimiento.

La importancia del uso de un simulador ha demostrado, a nivel académico e industrial, una amplitud de control de variables que facilita el manejo de las ondas y la efectividad de su recepción tomando en cuenta el entorno de transmisión; de esta manera es posible elaborar una programación que pueda simular la Multiplexación por División en Frecuencias Ortogonales (OFDM). La OFDM es un mecanismo de modulación de transmisión que divide un canal de frecuencia en sub canales de datos simultáneos o subportadoras; en la actualidad es muy utilizado por redes estándar de cuarta generación en Telefonía Móvil (LTE), redes WIFI, WIMAX e incluso puede emplearse en los Sistemas de Transmisión por Cable (ADSL o PLC).

Todos los criterios presentados serán analizados y estructurados con el objetivo de construir el prototipo simulador de OFDM, el cual se convertirá a su vez en unas nuevas herramientas para estudios posteriores en aulas y

laboratorios permitiendo así observar el comportamiento de este importante sistema.

### **1.1. Alcance**

El presente proyecto contempla la investigación, estudio y análisis del procesamiento de transmisión Multiplexación por División en Frecuencias Ortogonales (OFDM), identificando los diversos sistemas computarizados para diseñarlo, precisando los pros y los contras que derivan del uso de cada uno de ellos para elegir el más adecuado que se ajuste a los objetivos de ésta tesis.

Posteriormente se implementará un simulador de OFDM que permita a los estudiantes de la Universidad de las Américas observar y analizar el comportamiento de este sistema para mejorar sus conocimientos.

### **1.2. Justificación**

El Ingeniero en Redes y Telecomunicaciones debe poseer un conocimiento integro de electrónica y redes de comunicación, para ello debe comparar los conocimientos obtenidos en la teoría con prácticas que le permitan analizar y observar los procesos estudiados. Esto lo llevaría a afianzar sus conocimientos y confiar en ellos en su aplicación en el campo laboral.

La Universidad de las Américas (UDLA) actualmente no cuenta con un simulador que les permita llevar a cabo este proceso de aprendizaje en relación al funcionamiento del sistema OFDM, por lo que su implementación beneficiaría tanto a estudiantes como docentes ya que a los últimos les facilitaría su labor de enseñanza al llevar la teoría impartida a la práctica haciendo más didácticas las clases.

Además, la aplicación de este proyecto marcará un precedente para futuros proyectos llegando a ser una base fundamental en la investigación e implementación de la modulación OFDM en aporte fundamental para la Universidad de las Américas

### **1.3. Objetivos**

#### **1.3.1 Objetivo General**

Diseñar e implementar un prototipo de simulador de OFDM, apoyado en un entorno de desarrollo de un *software* matemático, el cual permita explicar de manera didáctica el funcionamiento de la Multiplexación por División de Frecuencias Ortogonal OFDM.

### 1.3.2. Objetivos específicos

- Analizar, todos los componentes y variables que intervienen en el proceso de modulación OFDM.
- Definir los parámetros y variables que serán utilizados en el proceso de modulación OFDM
- Diseñar el diagrama de flujo requerido para la implementación del software que permitirá visualizar las etapas de modulación OFDM.
- Desarrollar el prototipo de simulador OFDM en una herramienta informática.
- Realizar pruebas de funcionamiento del prototipo.
- Proponer la hoja guía de una práctica de laboratorio que involucre la herramienta desarrollada en el presente trabajo.

## 2. CAPÍTULO II. MARCO TEÓRICO

### 2.1. Antecedentes Investigativos.

La Multiplexación por División en Frecuencias Ortogonales (OFDM), siendo una de las aplicaciones tecnológicas de gran utilización en el entorno de las telecomunicaciones es un tema de análisis y estudio por diferentes profesionales tanto en Ecuador como a nivel mundial. Entre tantos documentos se pueden nombrar los siguientes:

En la India, entre los años 2005 y 2007, el D. Jagadeesh Roll desarrolló su maestría con la creación de un simulador capaz de generar OFDM con técnicas de transformación FFT, DHT y DCT con el objetivo de mejorar la calidad de las comunicaciones. Su simulación le permitió realizar modulaciones en BPSK y QPSK para realizar los estudios de interferencia y fuerza de la transmisión.

El norteamericano Steven Charles Hemple, en el 2012, desarrolló un simulador computarizado con presentaciones gráficas en el entorno de MATLAB, en el cual su simulador puede adquirir un conjunto de variables para ampliar y reforzar las investigaciones del OFDM como un futuro tecnológico. Utilizó el principio de transmisión de múltiples antenas y su simulador comprendido en varios archivos diferentes presentan gráficas con modulaciones variadas.

En España, Francisco Javier López Martínez utilizó un sistema MIMO-OFDM basado en QAM y un simulador computarizado con el objeto de determinar de manera más exacta los cálculos del BER y su control para mejorar la calidad de la transmisión, para esto presentó diversos puntos de estudio para crear los cálculos matemáticos en MATLAB y su representación del error y señales de transmisión.

A nivel nacional, el Ing. Juan Carlos Navarro Viveros desarrolló un modulador de OFDM y 64-QAM, usando el MATLAB y el OptiSystem como entorno gráfico controlado. Con dicho proyecto se obtuvieron cambios en el sistema de transmisión de señales RF por aplicación de fibra para generar un ahorro de energía del sistema vigente. Su simulador le permitió verificar la influencia de la longitud del canal de fibra como determinante en la presencia del ruido.

El Ing. Saúl Ignacio Ashqui Pagalo utilizó el Universal Software Radio Peripheral (USRP) y el LabView para diseñar una técnica de transmisión por radio con modulaciones digitales QPSK y 8PSK, junto con esto desarrolló la gráfica de comparación BER vs SNR. Entre sus resultados determinaron la importancia de la encriptación de la información realizada con un entorno computarizado que permite la programación de cada elemento.

Se debe tomar en cuenta que entre todos los entornos de simulación informáticas disponibles actualmente para desarrollar los simuladores de OFDM se han basado en la presentación gráfica del proceso de generación de la señal, dicho resultado se compara entre diversos métodos de generación los cuales son analizados y evaluados por el investigador.

## **2.2. Fundamentación Teórica.**

### **2.2.1. La Comunicación.**

Desde mucho antes de la existencia de la humanidad, la comunicación se presentó como la base inicial del enlace entre dos seres, pero a diferencia de otros organismos vivos, el ser humano ha dominado su entorno y lo adapta a su necesidad, así mismo como lo planteó con la comunicación por diversos medios y motivos ha logrado crear avances continuos en lo que conocemos como la era electrónica. Desde nuestro desarrollo como individuos hemos aprendido diversas formas de comunicarnos y por nuestras manos han pasado un conjunto de herramientas comunicativas, pero siempre el sistema de la comunicación permanece igual:

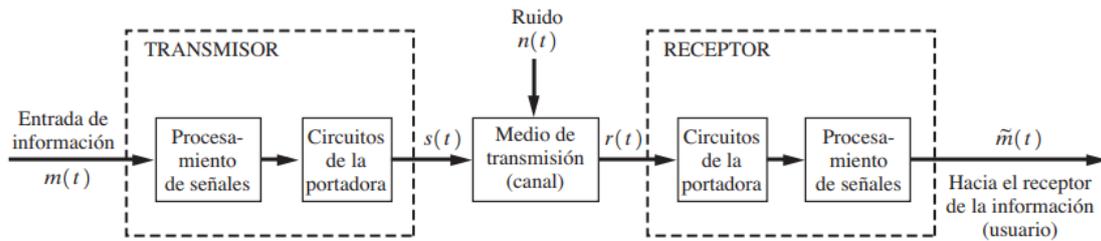


Figura 1. Esquema de Sistema de Comunicación.

Tomado de: (Couch, 2007, pág. 8)

Sin importar la estructura de un sistema de comunicaciones existen conjuntos de variables que pueden ser controladas y otras que no. En la actualidad la tendencia de la comunicación se fundamenta principalmente en la electrónica desde su aparición, esto ha traído consigo un conjunto de factores contribuyentes así mismo como problemas serios.

### 2.2.2. Sistemas de comunicación.

Indiferentemente del principio empleado, la comunicación abarca un sistema compuesto por los elementos básicos necesarios y muy conocidos, pero en el tema de Telecomunicaciones se deben tomar en cuenta dos de ellas:

- Sistemas Analógicos: Son aquellos que se establecen mediante el control de voltajes y corrientes en función al tiempo, incluyen los sonidos y ondas mecánicas.
- Sistemas digitales: Está formado por un conjunto de componentes físicos e intangibles que manejan información de manera discreta o no palpable.

### 2.2.3. Modulaciones.

Para asegurar la transmisión de una señal a lo largo del medio se ha requerido modificar dicha información para que en comportamiento permita el recorrido de la distancia o llegar al destino, del receptor. Para conseguir esto, las ondas analógicas y las digitales han requerido de la aplicación de cambios en sus características y su preparación para el envío.

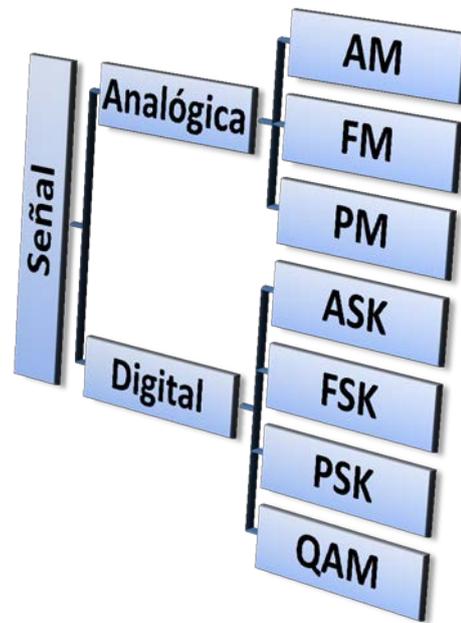


Figura 2. Clasificación de las modulaciones analógicas y digitales.

Cuando se tiene una ecuación del voltaje como una función senoidal que pueda ser utilizada como una señal portadora de alta frecuencia:

$$v(t) = V \cdot \text{Sen}(2\pi \cdot f \cdot t + \theta) \quad (\text{Ecuación1})$$

Donde  $V(t)$  corresponde al voltaje,  $V$  es la amplitud,  $f$  la frecuencia y  $\theta$  el desplazamiento de fase.

Se pueden obtener diversas modulaciones de dicha portadora para transmitir la información:

- **AM**, *amplitude modulation*, se refiere a la modulación que modifica la amplitud de la señal.
- **FM**, *frequency modulation*, genera la modulación de la frecuencia.
- **PM**, *phase modulation*, esta modulación gestiona un acondicionamiento de la fase de la señal.
- **ASK**, *amplitud shift keying*, está determinada por la modulación de amplitud de señales digitales.
- **FSK**, *frequency shift keying*, engloba la modulación de la frecuencia de la señal digital.
- **PSK**, *phase shift keying*, establece la modulación de la fase de una señal digital.
- **QAM**, *quadrature amplitude modulation*, esta modulación digital realiza una variación en la amplitud y en la fase.

#### **2.2.4. Multiplexación por División de Frecuencias Ortogonales (OFDM).**

Las redes de telecomunicaciones crean medios dedicados los cuales permiten el ahorro de tiempo, haciendo más efectivas las comunicaciones, evitando el desplazamiento físico de un mensajero a lo largo del recorrido entre el emisor del mensaje y el receptor, proporcionando una comunicación eficiente. La principal razón de su desarrollo es debido al costo de establecer un enlace dedicado entre dos usuarios de una red, el cual sería elevadísimo sin redes de comunicación, considerando que no todo el tiempo, los usuarios se comunican entre sí.

Históricamente “las primeras redes de telecomunicación propiamente dichas surgen con la aparición de la telegrafía óptica” (Estepa, 2004), con las cuales se pudo sustituir la mensajería tradicional por la trasmisión de todo tipo de mensaje, mejorar las condiciones y calidad de vida del ser humano y optimizar el proceso de la comunicación a distancia.

Es importante considerar determinados descubrimientos cuyos fundamentos constituyen la raíz de cambios y avances tecnológicos de relevancia para la humanidad, uno de ellos se ilustra en 1821, cuando Michael Faraday descubre la inducción eléctrica, logrando producir corriente eléctrica a través del movimiento de un conductor alrededor de un imán permanente, éste dispositivo es lo que conocemos como un generador y crea electricidad a partir del movimiento mecánico; también aparece Maxwell quien estableció las relaciones de cambio de la radiación electromagnética respecto a la frecuencia. Estos descubrimientos representan la piedra fundacional de las comunicaciones alámbricas e inalámbricas y por tanto las teorías que la avalan debe ser tomada en cuenta en la elaboración del presente trabajo de investigación.

La comprensión del tema en cuestión es posible cuando se establece la funcionalidad teórica y práctica de la Multiplexación por División de Frecuencias Ortogonales (OFDM), la cual es una técnica de transmisión que consiste en la multiplexación de múltiples portadoras establecidas a diferentes frecuencias, donde cada una transporta una información. Constituye una solución a los problemas de interferencia entre símbolos que se originan por la propagación multitrayecto de los sistemas inalámbricos, por lo cual esta técnica de modulación es globalmente utilizada.

Desde su aparición se le ha incluido en sistemas de propagación como lo son Wireless por las ventajas que ofrece; actualmente se estudia con más detenimiento para poder encontrar mejoras y desarrollar aplicaciones más exitosas. Su primera patente fue escrita en 1966 en los laboratorios Bell e introducidos por primera vez en la década de los 60.

Su prototipo era un diseño analógico por lo cual, el uso de grandes bancos de generadores sinusoidales para realizar el procesamiento de las señales de los múltiples sub-canales, hacía de la técnica multiportadora una forma no viable. Para los años 1971 con el avance y aporte para OFDM de Ebert y Weinstein, los cuales proponen como parte del sistema de modulación y demodulación la aplicación de la Transformada Discreta de Fourier e implementando un intervalo de guarda dentro del tiempo del símbolo, se eliminarían la ISI.

En la década de los 80, Paled y Ruiz incorporan el prefijo cíclico, así evitando dejar el intervalo de guarda vacío, comienzan a ser posibles los diseños digitales, implementando de manera más eficiente la DTF con la Transformada Rápida de Fourier (FFT), y con esta implementación se eliminaba los sistemas analógicos mismos que no eran viables por el uso de los bancos de generadores sinusoidales y demoduladores de los sistemas iniciales.

En este sentido, la viabilidad de OFDM no fue posible hasta la década de los 90 cuando los avances de los circuitos integrados VLSI (*Very-Large-Scale Integration*) lograron abaratar los costes de fabricación, pues para calcular la FFT era necesario un procesador de señal DSP (*Digital Signal Processor*), y de igual manera al abaratar el coste de estos circuitos y también de incrementar su capacidad de computo de una manera muy notable permitió que OFDM sea comercialmente posible. Una vez, definido la historia del concepto en estudio, se procede a explicar los fundamentos y componentes de la OFDM.

La transmisión de una información mediante el uso de señales electromagnéticas posee un desarrollo equivalente al avance tecnológico actual, aun así, las bases de generación de dicha modulación conservan el esquema casi constante de preparación de la señal. La modulación es prácticamente la aplicación de una señal portadora, generalmente sinusoidal, la cual es modificada para la transmisión de la información, de aquí se derivan las diversas modulaciones conocidas, entre la diversa gama se encuentra la OFDM.

“La OFDM es un sistema de modulación multitrayecto (*multipath Modulation system*, MCM) que utiliza transformadoras rápidas de Fourier (FFT) y transformadores inversos de Fourier (IFFT)”. (Orthogonal Frequency Division Multiplexing System with Frequency Offset Estimation, s.f.)

Una representación básica de la señal con OFDM es la siguiente:

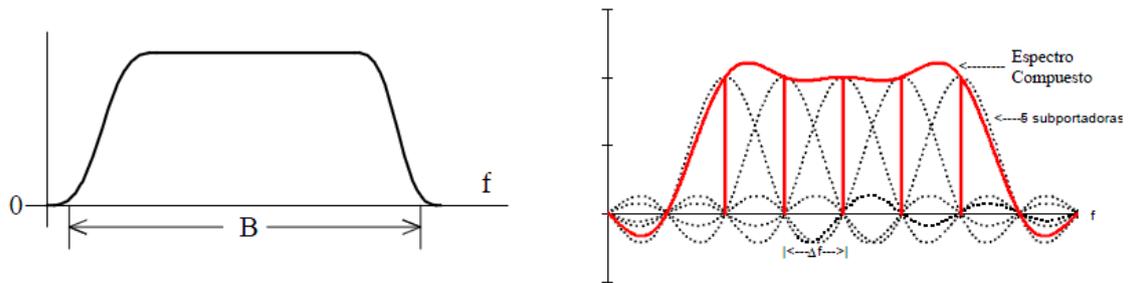


Figura 3. Composición de la señal OFDM.

Tomado de: (Briceño, 2012, pág. 582)

Esta imagen permite visualizar la constitución de la OFDM como la combinación de múltiples portadoras convirtiéndose en una técnica de modulación en excelencia para las comunicaciones inalámbricas motivado por su capacidad de transportar grandes cantidades de información. Esto ha motivado al desarrollo de empresas y grupos de investigaciones a nivel mundial con el simple objeto de desarrollar mejoras al OFDM a pesar de las incontables aplicaciones actuales.

### 2.2.5. Parámetros de las OFDM.

Al generar la OFDM se obtiene una señal de salida con las siguientes características:

1. Tasa de datos: De 6 Mbps a 48 Mbps
2. Tipo de modulación: BPSK, QPSK , 16 QAM y 64 QAM
3. Codificación: Convolutional concatenado con Reed Solomon.
4. Periodo de la FFT:  $1/\Delta = 3.2 \mu\text{seg}$
5. Tamaño de la FFT: 64 (58 - 48 para datos y 4 para señales piloto)
6. Separación de frecuencia entre subportadoras: 20 Mhz divididos en 64 portadoras de 0.3125 Mhz.
7. Duración del periodo de guarda: Un cuarto de símbolo (0.8  $\mu\text{seg}$ ).
8. Tiempo de símbolo: 4  $\mu\text{seg}$ .

### 2.2.6. Aplicaciones.

Desde su desarrollo a nivel comercial en las comunicaciones a mediados de los 80's, OFDM se ha expandido en cuanto a sus aplicaciones por su comportamiento en el manejo de las bandas anchas lo cual ha generado un incremento en el desarrollo de dicho sistema. Estos son unos ejemplos de los usos actuales del OFDM:

- La televisión digital DVB-T.
- Radio digital DAB y de baja frecuencia DRM.
- Protocolos de enlace ADSL.
- Protocolo de red de área local IEEE 802.11A/g (LAN e HIPERLAN2).
- Transmisión inalámbrica de datos WiMax.

### 2.2.7. Sistemas de Transmisión OFDM.

Los sistemas de transmisión de información, no solo deben cumplir con un conjunto de parámetros establecidos por los entes reguladores en materia de Telecomunicaciones, sino también presentan una diversidad de mecanismos de transmisión entre los cuales podremos comparar tres:

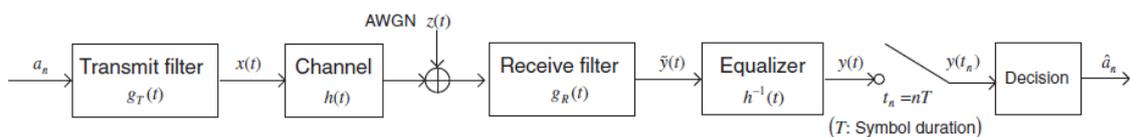


Figura 4. Diagrama de bloque de un sistema OFDM.

Tomado de: (Cho, 2010, pág. 112)

### 2.2.8. Transmisión con una portadora.

Este sistema está contemplado por la aplicación de una banda de un canal  $h(t)$  y de un ancho de banda  $W$ , por el cual se envía una información con una función  $\{a_n\}$  y un período  $T$ , esto corresponde a una tasa de información  $R = 1/T$ . Dicha señal en función a un pulso de tiempo es filtrado por el transmisor en función de  $g_T(t)$ . Durante la transmisión la señal pasa el canal como una función  $x(t)$  hasta ser recibido como  $g_R(t)$  por el filtro donde se aplica la función  $g_R(t)$ , el ecualizador estabiliza la señal con una función  $h^{-1}(t)$  finalizando con una respuesta:

$$y(t) = \sum_{m=-\infty}^{\infty} a_m \cdot g(t - m \cdot T) + z(t) \quad (\text{Ecuación2})$$

Las consideraciones que debemos de tomar en cuenta son que:  $z(t)$  representa al ruido obtenido durante la transmisión y  $g(t)$  es el impulso de respuesta desde el inicio hasta el final del proceso siendo:

$$g(t) = g_T(t) \cdot h(t) \cdot g_R(t) \cdot h^{-1}(t) \quad (\text{Ecuación3})$$

Este efecto causado por el canal es totalmente asumido por el filtro del cual se puede idealizar que suprime completamente la interferencia recogida por la señal y a su vez tomando en cuenta el último dato de la información correspondiente por el término  $a_n$ , se tiene la ecuación decodificada:

$$y(t_n) = a_T \cdot g(0) \sum_{m=-\infty, m \neq n}^{\infty} a_m \cdot g(n - m)T \quad (\text{Ecuación4})$$

### 2.2.9. Transmisión con múltiples portadoras.

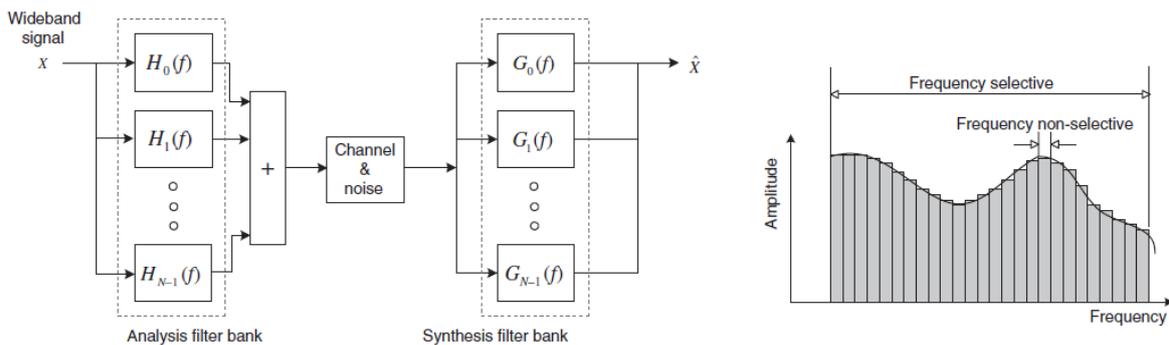


Figura 5. Diagrama de transmisión con múltiples portadoras.

Tomado de: (Cho, 2010, pág. 116)

Con éste proceso se seleccionan múltiples portadoras para incrementar la tasa de transmisión como se muestra en la figura. A través del filtro múltiple  $H_K(f)$  se analiza el ancho de banda para ser fragmentado en el transmisor y posterior a

la transmisión será sintetizado por el filtro  $\mathbf{G}_K(\mathbf{f})$  del receptor y sigue el proceso de decodificación de modo similar que en la codificación.

Una característica principal de éste proceso de transmisión es que al no poderse seleccionar las bandas estrechas se reduce la complejidad en la ecualización y a su vez que mantiene la ortogonalidad de los subcanales.

Para los sistemas de múltiples canales se aplica la división del ancho de banda en N cantidad de subcanales con frecuencias  $\mathbf{f}_K$ , donde  $\mathbf{K} = 0, 1, 2, \dots, N-1$ . En la figura 6 se contempla solo uno de los canales, el conjunto completo de los canales forma la transmisión ortogonal. El término  $\mathbf{X}_i[\mathbf{k}]$  representa a la señal transmitida,  $\mathbf{Y}_i[\mathbf{k}]$  la señal recibida, y  $\mathbf{f}_k$  es la frecuencia portadora en el intervalo  $\mathbf{f}_k$ .

El espectro de frecuencia obtenida por éste tipo de transmisión constituido por todos los subcanales se centran en las diferentes frecuencias de las portadoras junto con el ordenamiento descrito en la figura 6 se convierte en una transmisión FMT (*Filtered Multi-Tone*), éste tipo de transmisión es más complejo y exigente debido a la demanda de componentes requeridos como filtros, codificadores, decodificadores y al incremento del número de portadoras.

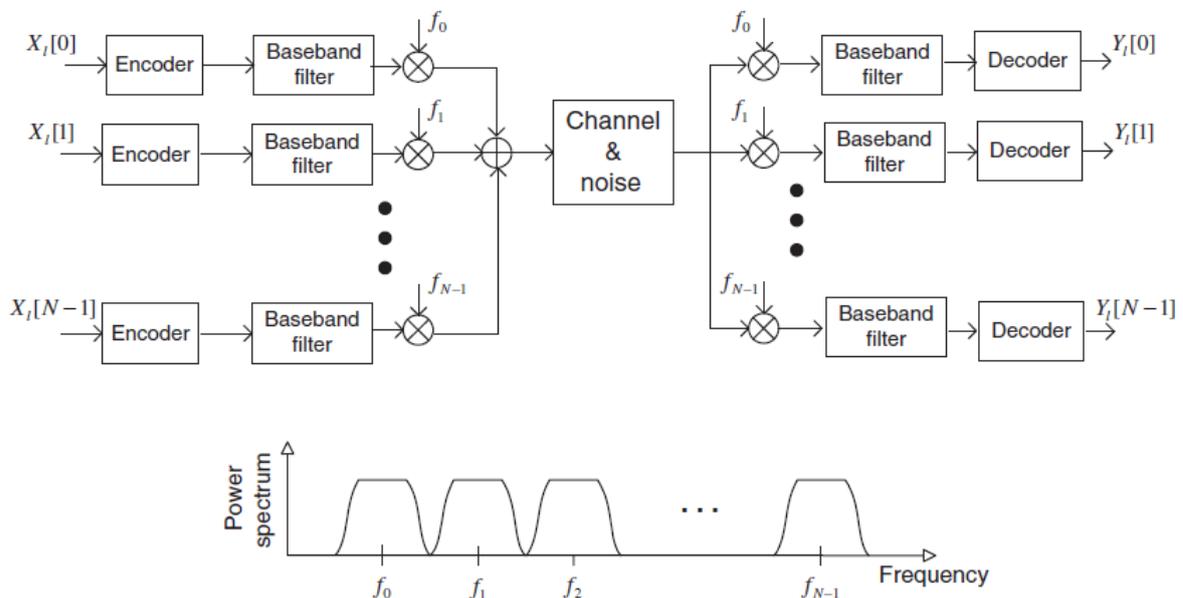


Figura 6. Sistema multicanal con detalle.

Tomado de: (Cho, 2010, pág. 116)

### 2.2.10. Transmisión OFDM.

Todas las transmisiones OFDM se manejan de un modo diferente como en los sistemas FMT, se utilizan señales múltiples ortogonales como portadoras que son solapadas en el espectro por medio del uso de una portadora Nyquist.

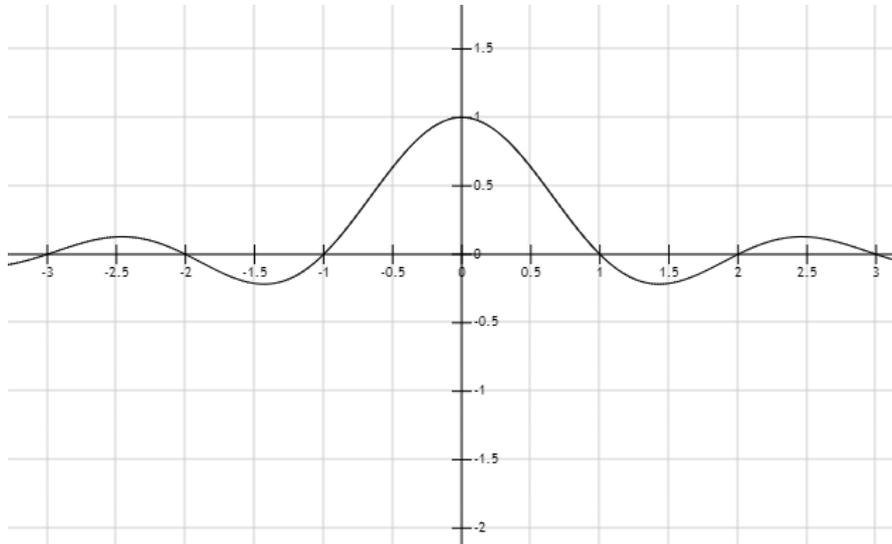


Figura 7. Ejemplo de la portadora Nyquist.

La base del uso de la portadora de Nyquist es la aplicación de la transformada de Fourier con la ecuación:

$$\sum_{i=-\infty}^{\infty} G\left(f - \frac{i}{T}\right) = T \quad (\text{Ecuación 5})$$

Donde  $\mathbf{G(f)}$  es la transformada de Fourier aplicada en  $\mathbf{g(t)}$ , conocido como el impulso de respuesta. Por lo tanto, éste tipo de transmisiones usan la transformada discreta de Fourier (DFT) y su inversa (IDFT) que es fundamental para el manejo de las señales ortogonales a través de los transformadores rápidos de Fourier (FFT) y su inverso (IFFT).

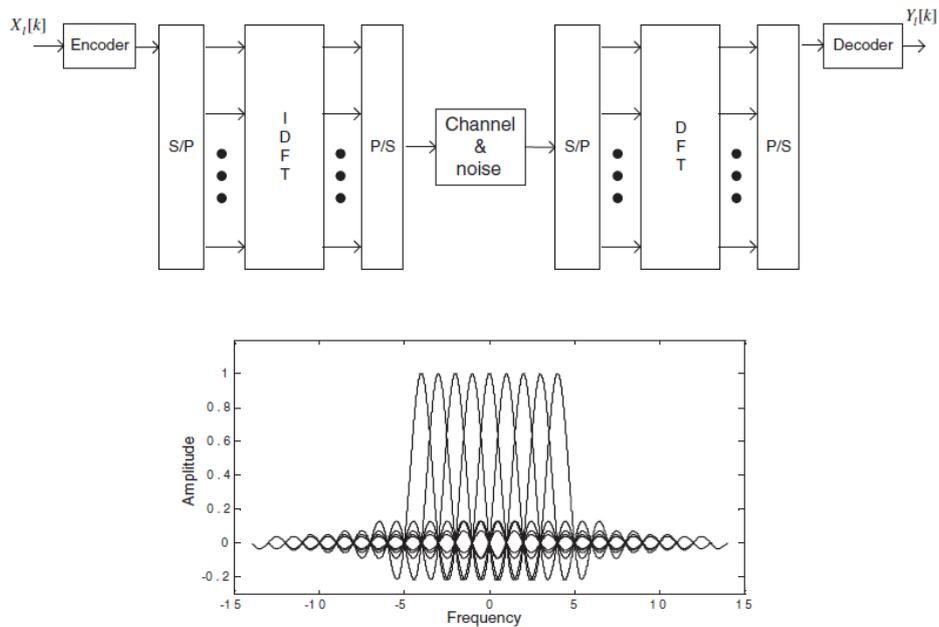


Figura 8. Aplicación de la transformada de Fourier.

Tomado de: (Cho, 2010, pág. 118)

Se debe detallar que se toma la información  $x[n]$  y se realiza la división del ancho de banda en  $N$  partes que serán tomadas por el transmisor como

$$\{X_l[k]\}_{k=0}^{N-1} \quad (\text{Ecuación6})$$

Para generar

$$\{x[n]\}_{n=0}^{N-1} \quad (\text{Ecuación7})$$

El receptor obtiene la señal en la forma

$$\{y[n]\}_{n=0}^{N-1} \quad (\text{Ecuación8})$$

El cual implica el ruido obtenido durante la transmisión

$$y[n] = x[n] + w[n] \quad (\text{Ecuación9})$$

Siendo  $w[n]$  el ruido agregado y la ecuación del receptor:

$$\{Y_l[k]\}_{k=0}^{N-1} \quad (\text{Ecuación10})$$

El resultado es prácticamente comparado con la repetición o suma de la portadora Nyquist que se encuentra espaciada por una razón  $1/T$ .

### 2.2.11. Transmisión FMT.

Éste sistema es de modulación multicanal como se puede ver en la figura 9.

$$\{X_l[k]\}_{k=0}^{N-1} \quad (\text{Ecuación11})$$

La ecuación anterior está solapada  $(\mathbf{N}+\mathbf{v})/T$  veces y es tomada por un filtro de paso de banda (BPF) donde  $\mathbf{v}$  es un entero no negativo y  $\mathbf{h}(\mathbf{t})$  es el impulso de respuesta del filtro. Al entrar la información al sistema, el filtro recibe  $\mathbf{k}$  cantidad de subcanales y lo entrega al transmisor en la forma

$$h_k(t) = h(t) \cdot e^{j \cdot 2\pi \cdot f_k \cdot t} \quad (\text{Ecuación 12})$$

Donde  $f_k$  es la frecuencia central del último subcanal. Debido a la sobreexposición en el transmisor, la frecuencia de exposición es  $(\mathbf{N}+\mathbf{v})/T$  y a su vez,  $f_k = (\mathbf{N}+\mathbf{v}) \cdot k / (\mathbf{N}T)$ . Al recibir la señal, cada filtro ajustado con  $\mathbf{h}_k(\mathbf{t})$  y  $(\mathbf{N}+\mathbf{v})/T$  debe procesar la sobreexposición para obtener la señal:

$$\{Y_l[k]\}_{k=0}^{N-1} \quad \text{Ecuación (13)}$$

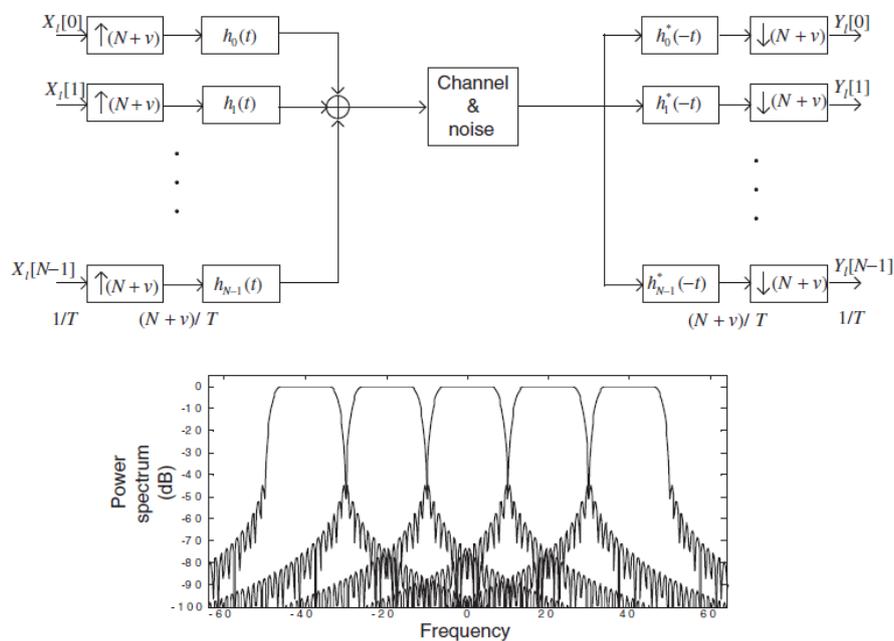


Figura 9. Sistemas de modulación FMT.

Tomado de (Cho, 2010, pág. 119)

## 2.2.12. MATLAB.

Para el desarrollo de la simulación se debe elegir la adecuada herramienta informática que cumpla con los requerimientos necesarios para su correcta implementación. Un CAS es un programa que permite al usuario realizar cálculos matemáticos complejos a través del álgebra computacional o cálculo simbólico, es decir, usando variables y constantes no cuantificadas; estos

*softwares* tienen un amplio uso todas las áreas científicas y de ingeniería, desde la enseñanza, la investigación y la aplicación a nivel industrial.

Entre las funciones de estos programas matemáticos están: Operaciones matemáticas complejas tales como cálculo matricial, vectores, álgebra lineal, etc., estadística, visualización y creación de gráficos en 2D y 3D, diseño e implementación de simuladores de sistemas de control y sistemas dinámicos, entre otros.

Para el desarrollo de este proyecto se ha elegido el MATLAB (Por sus siglas en inglés *Matrix Laboratory*, "laboratorio de matrices) el cual es uno de los CAS más usado y conocido por científicos e ingenieros de diferentes áreas. De su página oficial: "MATLAB combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente." (Mathworks).

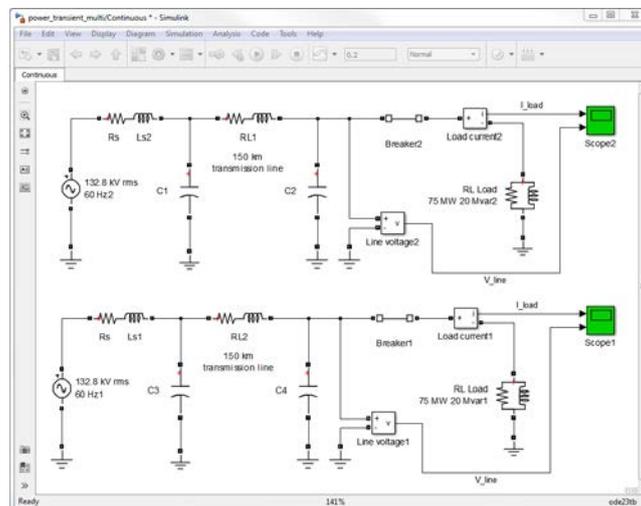


Figura 10. Ejemplo del diseño de un circuito en MATLAB.

### 2.2.13. Entorno de desarrollo.

MATLAB tiene una gran variedad de funciones tanto generales como de especialización, incluyendo la elaboración de gráficas lo que permite la amplitud de su aplicación. Posee un lenguaje de programación propio, similar a lenguajes como Basic o Lenguaje C, esto permite la creación de funciones y la gestión de archivos. Cuenta con una amplia librería con funciones de cálculo y muchas de ellas son comunes a las usadas en C o Fortran. Presenta además los llamados Toolboxes, que son paquetes de programación para diversos fines.

#### **2.2.14. Entorno de Simulación.**

MATLAB cuenta con un sistema propio llamado Simulink, sirve para modelar y simular sistemas de control y sistemas físicos usando diagramas de bloques. Su utilidad está en la facilidad de visualización del comportamiento de los sistemas y para ello dispone de herramientas de análisis y almacenamiento de los resultados. La interfaz gráfica de Simulink es amigable por lo que es de fácil uso, se puede modelar usando el mouse, personalizando los bloques con los que se trabajara.

#### **2.2.15. Entorno de Interfaz de Usuario.**

MATLAB provee al programador de una herramienta para crear interfaz de usuario final que necesiten ingresar continuamente datos para su evaluación, este sistema se llama GUIDE (*Graphical User Interface Development Environment*) esto quiere decir entorno de desarrollo gráfico de la interfaz de usuario. Su forma de uso es idéntica al usado en lenguajes de programación tales como Visual C.

### **3. CAPÍTULO III. COMPONENTES DEL OFDM**

#### **3.1 Análisis de Sistemas OFDM.**

Como se ha podido apreciar desde el capítulo anterior, la Multiplexación por División en Frecuencias Ortogonales (OFDM), es un método de modulación de señal digital en el cual un conjunto de información es separada en una cantidad definida de canales y cada una posee una frecuencia diferente entre sí para mejorar la transferencia de la información reduciendo la interferencia y el ruido del ambiente. Un esquema del sistema con mayor cantidad de procesos se puede representar de la siguiente manera:

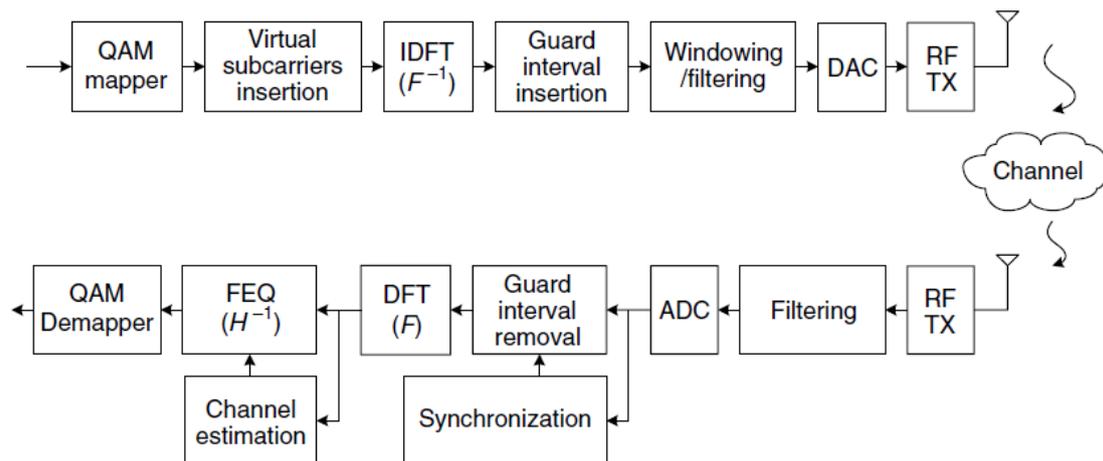


Figura 11. Diagrama del OFDM.

Tomado de: **(Kaise, 2008, pág. 209)**

Los avances en las telecomunicaciones han causado el incremento en las exigencias en cuanto al proceso de la información, los desarrollos en la telefonía móvil es un mejor ejemplo. Éste tipo de sistemas opera mediante la aplicación de OFDM y generalmente con modulación 16 QAM, por lo tanto, en base al creciente desempeño de ésta tecnología el presente proyecto será enfocado a ésta modulación

Entre los tipos de modulaciones digitales, la Modulación por Cuadratura de Amplitud (QAM), desarrolla la modificación de la fase y la amplitud de una onda conservando la frecuencia de manera constante, a su vez que ha permitido mejorar las transmisiones minimizando las interferencias del canal de transmisión. La aplicación del QAM es también representado como la modernización de la modulación BPSK (modulación binaria por desplazamiento de fase) y QPSK (modulación cuaternaria por desplazamiento de fase), las modulaciones QAM están diferenciadas por la cantidad de imágenes que son utilizadas en ella: 2 QAM (BPSK), 4 QAM (QPSK), 8 QAM, 16 QAM, 32 QAM, 64 QAM y 256 QAM.

A partir del 16 QAM el proceso ha sido considerado como de mejor efectividad, por otro lado, las modulaciones mayores requieren la implementación de mejores equipos de mayor precisión. Respecto a la captación de interferencia, existen estudios previos donde han experimentado con la tasa de error de diferentes modulaciones y con múltiples variaciones de datos.

La facilidad del cálculo y la viabilidad de su implementación permiten seleccionar al 16 QAM como el procedimiento seleccionado para la realización de éste estudio en combinación con el OFDM, por lo tanto, a partir de ésta sección se realiza el análisis de sólo el proceso de transmisión de una señal utilizando la modulación 16 QAM con OFDM.

### 3.2. Proceso de transmisión.

#### 3.2.1 Generación de la imagen.

Se debe tener en cuenta que un símbolo es toda información digital que ingresa al sistema OFDM para su modulación y envío, dicha imagen está constituida por número binarios ordenado aleatoriamente; éste conjunto debe ser suministrado por un generador de imagen o un software debidamente programado.

El siguiente símbolo puede ser tomado como ejemplo:

$$d_n = 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0$$

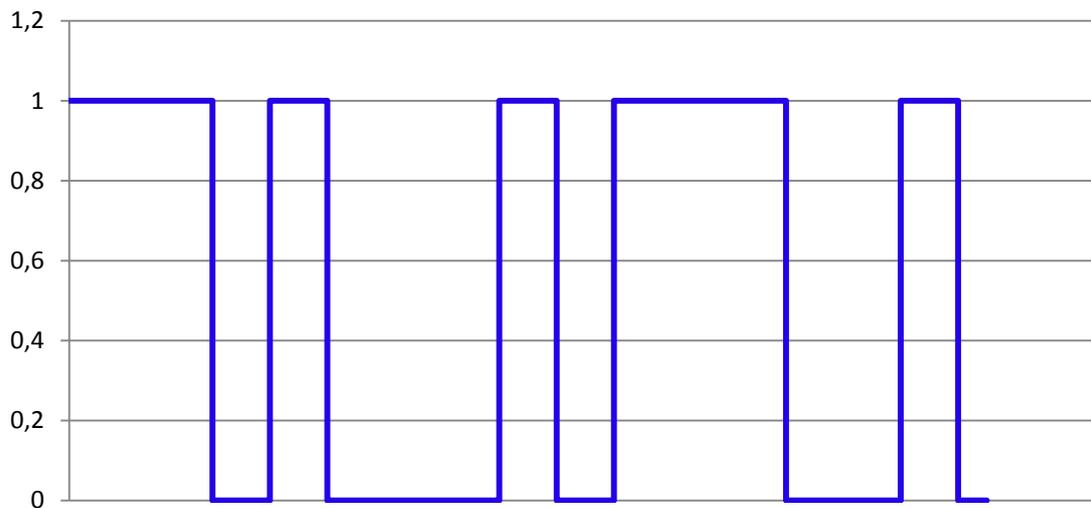


Figura 12. Grafica de la imagen binaria.

También es expresado como:

$$\{d(n)\} \quad \text{(Ecuación 14)}$$

#### 2.2.2. Convertidor Serial-Paralelo (S/P).

En éste proceso la imagen ingresada es separada en base a la cantidad de  $N$  portadoras utilizadas por el sistema, en el caso de estudio corresponde con 4 bit por símbolo para el 16 QAM, esto se combina en un formato paralelo por cada dato suministrado y asignado a una portadora.

Las especificaciones para las OFDM referentes a las portadoras es que pueden tener  $N$  cantidad de éstas entre los valores 16 y 1024, éste rango depende del uso de la modulación. Esto se realiza con el posicionamiento del interruptor automático del convertidor con el paso de cada bit de información; el cambio de posición se cumple hasta completar toda la información usando cíclicamente las  $N$  portadoras seleccionadas.

La salida de éste proceso es el siguiente conjunto:

Tabla 1.

*Reordenamiento de la imagen binaria por el conversor S/P.*

Portadora	Símbolo
1	1101
2	0001
3	0111
4	0010

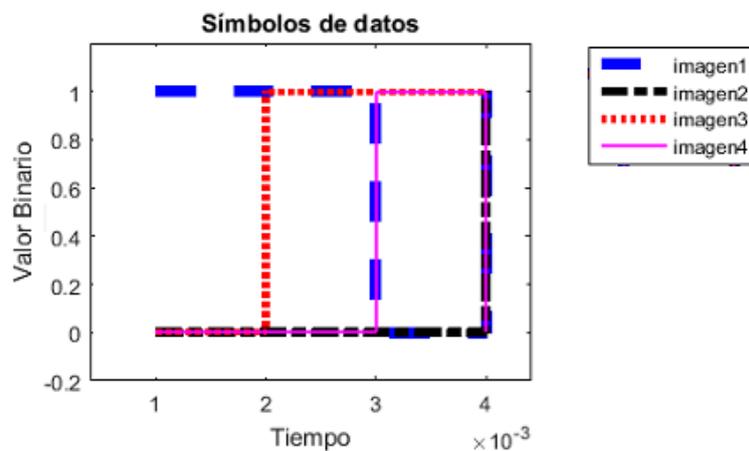


Figura 13. Símbolos de las imágenes de datos.

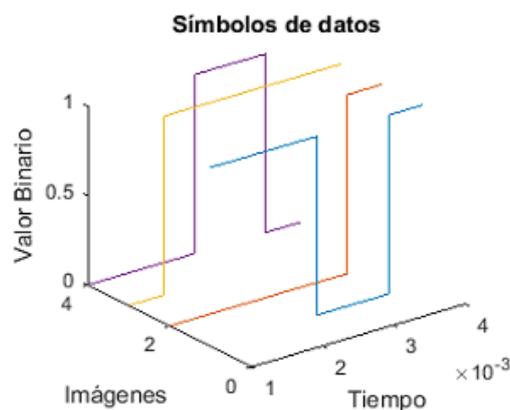


Figura 14. Representación 3D de los símbolos

### 3.2.2 Codificador.

El codificador también llamado mapeador recibe la imagen para aplicarle el formato 16 QAM, para ello debe utilizar una fase inicial como referencia y un símbolo extra que es añadido al inicio del proceso. Se obtiene un conjunto de símbolos mapeados a un ángulo de desfase. Los símbolos 16 QAM se expresan:

$$d(n) = a(n) + jb(n) \quad (15)$$

Se puede verificar que  $a(n)$  es la componente en fase y  $b(n)$  es la componente en cuadratura.

Éste proceso es parte integral de esta modulación. Se implantan las portadoras:

$$\cos(\omega_n \cdot t) \quad y \quad \text{Sen}(\omega_n \cdot t) \quad \text{Ecuación(16)}$$

Y son expresadas como:

$$\tilde{x}_I(t) \quad y \quad \tilde{x}_Q(t) \quad \text{Ecuación(17)}$$

Que son sumadas para retornar a formar un símbolo de salida.

$$\tilde{x}_n(t) \quad \text{Ecuación(18)}$$

Con ésta modulación se posee 16 símbolos diferentes también llamados quadrabit, el resultado de éste proceso se conoce como una combinación de fase y amplitud.

La cantidad de símbolos de modulación utilizados son representados en el cuadrante de la figura 15, a esto se le conoce como la constelación, la modulación 16 QAM usa 16 símbolos con diferentes fases y amplitudes, el objetivo de esto es que se generen nuevos símbolos que combinados con éstas portadoras con la aplicación de un código Gray en donde la diferencia para todas las imágenes será de un bit, no existirán puntos repetidos en la constelación, por ejemplo:

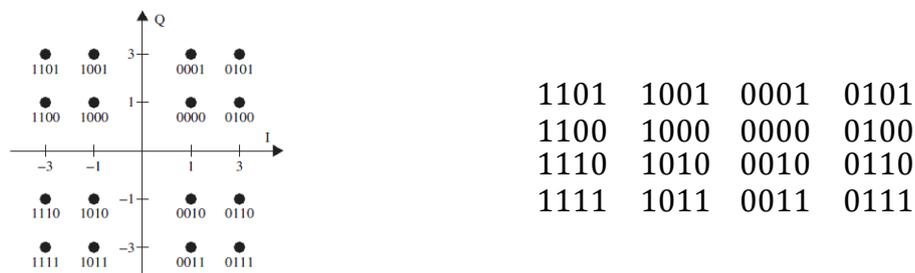


Figura 15. Mapeado base de la modulación 16QAM.

Tomado de: **(Kaise, 2008, pág. 209).**

El símbolo de entrada a éste modulador es reordenado en grupos de cuatro bits, siguiendo el ejemplo anterior, los cuatro primeros son los siguientes:

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ Q & Q' & I & I' \end{bmatrix}$$

Las posiciones de referencia I, I', Q y Q' son la división del bit en los cuatro canales del 16 QAM, donde se posee una preselección basada en la polaridad del número binario, para las posiciones I y Q los binarios 1 son positivos y los binarios 0 los negativos; en las posiciones I' y Q' se asigna el valor numérico

de 0,821 V para los binarios 1 y 0,22 V para los binarios 0. Consecutivamente los términos I son multiplicados por la portadora  $Sen(\omega \cdot t)$ , y los términos Q por la portadora  $Cos(\omega \cdot t)$ . Del ejemplo se tiene:

Tabla 2.

*Modulación 16QAM por cada 4 bits.*

Posición	Binario	Polaridad	Valor	Portadora
Q	1	+		$+0,821V \cdot Cos(\omega \cdot t)$
Q'	1		0,821 V	
I	0	-		$-0,821V \cdot Sen(\omega \cdot t)$
I'	1		0,821 V	

La imagen de salida es sumada y resulta:

$$-0,821V \cdot Sen(\omega \cdot t) + 0,821V \cdot Cos(\omega \cdot t) = 1,161V \cdot Sen(\omega \cdot t + 135^\circ)$$

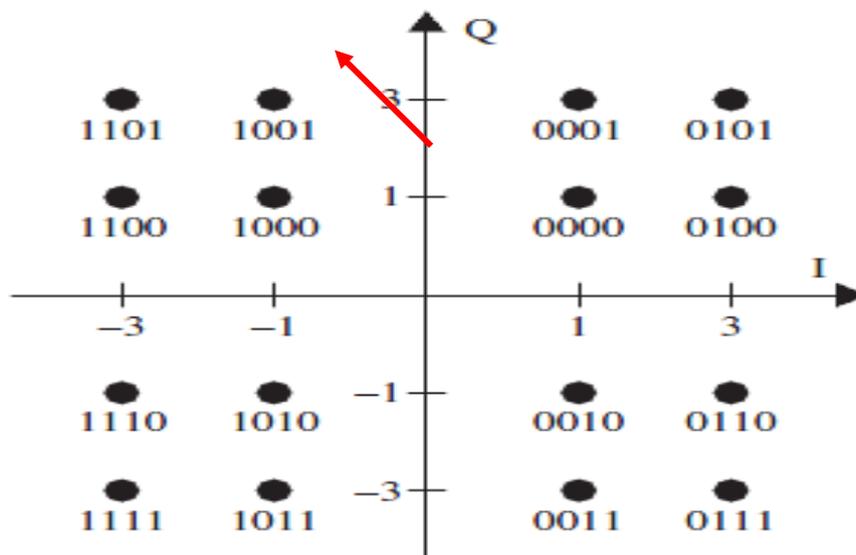


Figura 16. Mapa de constelaciones

Tomado de: **(Kaise, 2008, pág. 209)**

Si se toma la constelación de la figura anterior y ubicamos el punto [1 1 0 1] y se traza una línea desde el origen, se puede obtener un ángulo de  $135^\circ$  y la magnitud de la línea está dada por  $\sqrt{(0,821)^2 + (0,821)^2} = 1,161$ . Del mismo modo se procede con los demás datos del símbolo de entrada. El MATLAB tiene muchas configuraciones de las entradas de datos, para la realización de la modulación el programa permite la conversión de cada símbolo de cuatro bits como un conjunto binario a decimal, con esto se usa la siguiente gráfica y tabla de referencia:

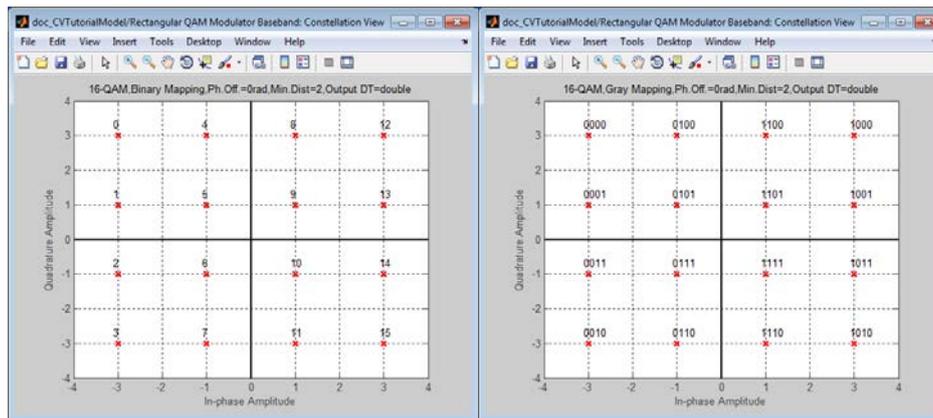


Figura 17. Visualización de la Constelación según MATLAB. Tomado de: (MathWorks)

Tabla 3.

Tabla para codificación según modulación 16QAM.

Entrada binaria				Salida modulada	Código Gray	Decimal	Complejo	
Q	Q'	I	I'					
0	0	0	0	0,311 V	-135°	0000	0	-3 +3i
0	0	0	1	0,850 V	-165°	0001	1	-3 +1i
0	0	1	0	0,311 V	-45°	0011	2	-3 -3i
0	0	1	1	0,850 V	-15°	0010	3	-3 -1i
0	1	0	0	0,850 V	-105°	0110	4	-1 +3i
0	1	0	1	1,161 V	-135°	0111	5	-1 +1i
0	1	1	0	0,850 V	-75°	0101	6	-1 -3i
0	1	1	1	1,161 V	-45°	0100	7	-1 -1i
1	0	0	0	0,311 V	135°	1100	8	3 +3i
1	0	0	1	0,850 V	165°	1101	9	3 +1i
1	0	1	0	0,311 V	45°	1111	10	3 -3i
1	0	1	1	0,850 V	15°	1110	11	3 -1i
1	1	0	0	0,850 V	105°	1010	12	1 +3i
1	1	0	1	1,161 V	135°	1011	13	1 +1i
1	1	1	0	0,850 V	75°	1001	14	1 -3i
1	1	1	1	1,161 V	45°	1000	15	1 -1i

Tomado de: (Tomasi, 2003, pág. 502)

Aplicando la tabla anterior y siguiendo el ejemplo, se tienen cuatro imágenes de salida con números complejos:

Tabla 4.

Realización del mapeo de los símbolos.

Portadora	Símbolo	Símbolo
-----------	---------	---------

	de entrada	de salida
1	1101	$1 + 1i$
2	0001	$-3 + 1i$
3	0111	$-1 - 1i$
4	0010	$-3 - 3i$

La gráfica de la constelación resulta del siguiente modo:

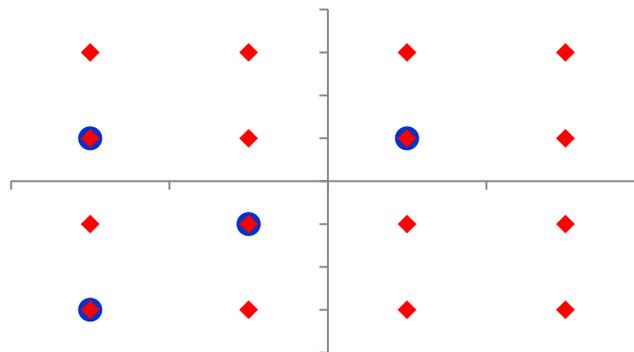


Figura 18. Constelación de salida del ejemplo numérico.

### 3.2.3 Inserción de los pilotos.

Los pilotos son utilizados para conservar una separación entre las imágenes de entrada a este proceso, su identificación es la de un tono y su simbología es comúnmente una flecha. La inserción de los pilotos es realizada con la imagen de salida del mapeador que puede ser representado por el vector del ejemplo de estudio:

Tabla 5.

Imagen de salida de la modulación 16 QAM

Símbolo de entrada	Valor
1	$+1 - 1j$
2	$-3 - 1j$
3	$-1 + 1j$
4	$-3 + 3j$

Estos valores son recibidos por un conversor de Serial a Paralelo (S/P) que realiza la división de la imagen en N cantidad de portadoras que serán utilizadas para insertar un piloto que en el ejemplo numérico será representado

por el valor vectorial cero [ 0] debido a que se manejan cuatro valores. La división en paralelo e inserción del piloto ejemplificados son:

Tabla 6.

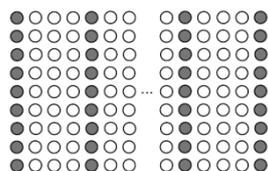
*Inserción de pilotos.*

Símbolo de entrada	Valor	Conversión S/P	Inclusión del piloto
			0
1	$1 + 1j$	$1 + 1j$	0
2	$-3 + 1j$	$-3 + 1j$	$0 + 0j$
3	$-1 - 1j$	$-1 - 1j$	0
4	$-3 - 3j$	$-3 - 3j$	0

El resultado de este procedimiento es una matriz que resulta de la concatenación de cada imagen y una portadora. Para el caso del estudio se tiene:

$$[ 0, 1 + 1j, 0, -3 + 1j, 0 + 0j, -1 - 1j, 0, -3 - 3j, 0 ]$$

Donde una forma visual de representar los pilotos como esferas grises y los datos como esferas blancas de este procedimiento es el siguiente:



*Figura 19.* Representación de la inserción de pilotos.

Tomado de: (Pei-Yun, 2007, pág. 126)

La gráfica de la inserción de piloto es desarrollada realizando una división de los valores numéricos reales de los imaginarios que han sido obtenidos. La siguiente tabla de valores es tomada en cuenta:

Tabla 7.

*Representación numérica de la inserción de pilotos*

Num.	1	2	3	4	5	6	7	8	9
Real	0	1	0	-3	0	-1	0	-3	0
Imaginario	0	1	0	1	0	-1	0	-3	0

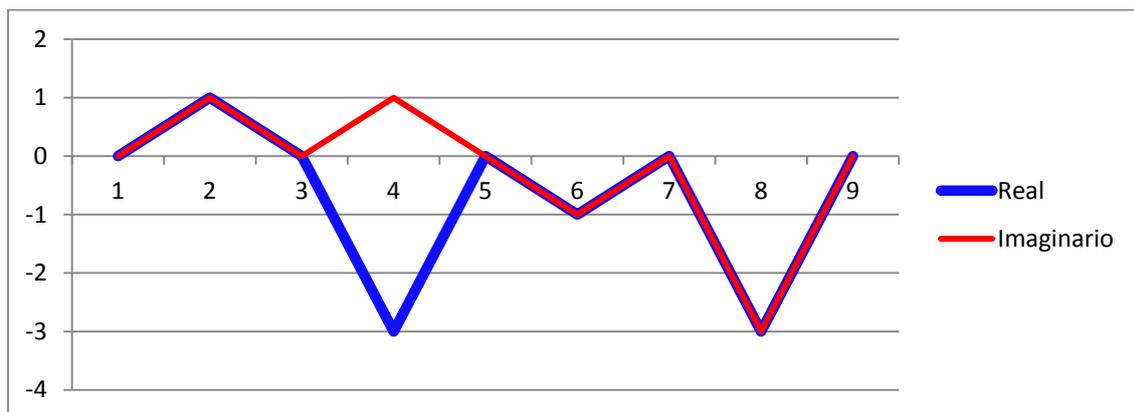


Figura 20. Gráfica de la inserción de pilotos.

### 3.2.4 IFFT.

La aplicación de las transformadas rápidas de Fourier son las funciones que permitieron el establecimiento del OFDM, a pesar del alto consumo de energía dichas transformadas permiten manipular las señales entre el dominio del tiempo y el de la frecuencia. En éste proceso se le aplica la transformada rápida de Fourier inversa (IFFT) para cambiar la longitud de cada imagen recibida a una misma longitud. Para éste proceso se usa la siguiente fórmula:

$$x(n) = \frac{1}{N} \sum_{i=0}^{N-1} X(i) \cdot e^{j\frac{2\pi \cdot n \cdot i}{N}} \quad (\text{Ecuación 19})$$

El resultado obtenido es la modulación de las diversas imágenes que entran al IFFT permitiendo generar una imagen con múltiples solapamientos en el dominio del tiempo.

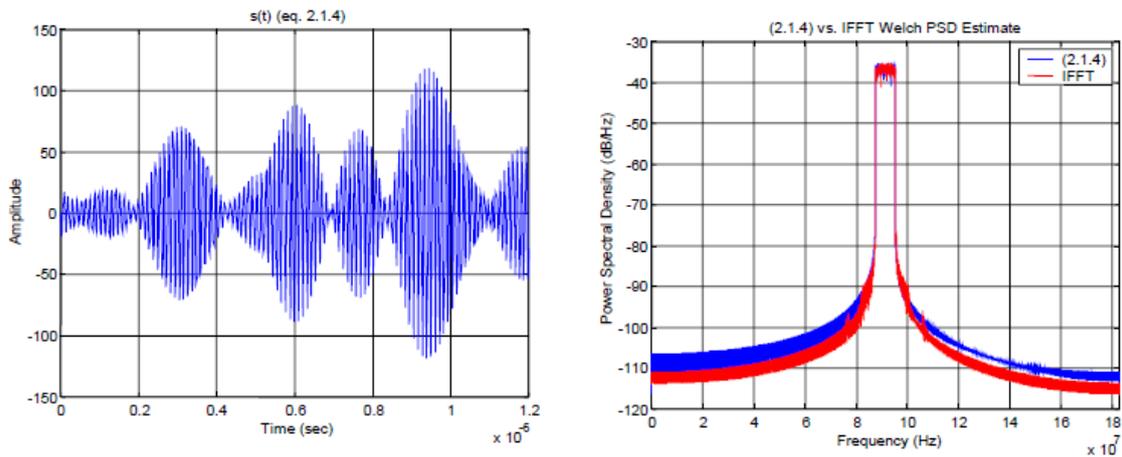


Figura 21. Tiempo y frecuencia de las portadoras IFFT.

Tomado de: (Acosta, 2000, pág. 8)

El MATLAB posee la función de generar la inversa de la transformada rápida de Fourier y posee la opción de generar múltiples valores de salida lo que es conocido como la longitud del IFFT, para el caso de estudio se seleccionó una longitud de 128 datos los cuales son separados en cuatro partes p ara obtener los valores numéricos de las cuatro portadoras del análisis:

Datos de entrada:

$$[ 0, 1 + 1j, 0, -3 + 1j, 0 + 0j, -1 - 1j, 0, -3 - 3j, 0 ]$$

Imagen de salida (IFFT):

Tabla 8.

Tabla de resultado de aplicación de IFFT

Imagen 1	Imagen 2	Imagen 3	Imagen 4
-0,0468750000000000 - 0,0156250000000000i	-0,0156250000000000 + 0,0468750000000000i	0,0468750000000000 + 0,0156250000000000i	0,0156250000000000 - 0,0468750000000000i
-0,0367620775869971 - 0,0269645700913457i	-0,0243841278637641 + 0,0402414011586608i	0,0367620775869971 + 0,0269645700913457i	0,0243841278637641 - 0,0402414011586608i
-0,0241430791658444 - 0,0343456331467416i	-0,0302813185502948 + 0,0317468769599113i	0,0241430791658444 + 0,0343456331467416i	0,0302813185502948 - 0,0317468769599113i
-0,0104343086748280 - 0,0372716651994509i	-0,0329497418278598 + 0,0225132753154761i	0,0104343086748280 + 0,0372716651994509i	0,0329497418278598 - 0,0225132753154761i
0,00288046986643839 - 0,0357346260696651i	-0,0324351847743090 + 0,0137069416596561i	-0,00288046986643839 + 0,0357346260696651i	0,0324351847743090 - 0,0137069416596561i
0,0144141711518017 - 0,0302069244624358i	-0,0291846981268509 + 0,00639372392760585i	-0,0144141711518017 + 0,0302069244624358i	0,0291846981268509 - 0,00639372392760585i
0,0230280229882013 - 0,0215795977539588i	-0,0239862094099731 + 0,00140671521211717i	-0,0230280229882013 + 0,0215795977539588i	0,0239862094099731 - 0,00140671521211717i
0,0279535515174761 -	-0,0178663629480437 -	-0,0279535515174761 +	0,0178663629480437 +

0,0110540522751402i	0,000757282826680094i	0,0110540522751402i	0,000757282826680094i
0,0288712353909777 +	-0,0119588572614091 +	-0,0288712353909777 +	0,0119588572614091 +
0,000000000000000i	0,000000000000000i	0,000000000000000i	0,000000000000000i
0,0259371768309350 +	-0,00735911245009858 +	-0,0259371768309350 -	0,00735911245009858 -
0,0102039525431126i	0,00336502685135003i	0,0102039525431126i	0,00336502685135003i
0,0197551491626649 +	-0,00498279039922235 +	-0,0197551491626649 -	0,00498279039922235 -
0,0183286966378627i	0,00864997997872171i	0,0183286966378627i	0,00864997997872171i
0,0112976757820507 +	-0,00544530538211215 +	-0,0112976757820507 -	0,00544530538211215 -
0,0234338438973668i	0,0148768594110864i	0,0234338438973668i	0,0148768594110864i
0,00178564476170802 +	-0,00897704242944240 +	-0,00178564476170802 -	0,00897704242944240 -
0,0249719388785146i	0,0208983393273904i	0,0249719388785146i	0,0208983393273904i
-0,00745931028136121 +	-0,0153848259074494 +	0,00745931028136121 -	0,0153848259074494 -
0,0228459548192016i	0,0255411229198919i	0,0228459548192016i	0,0255411229198919i
-0,0151723736499906 +	-0,0240647384480954 +	0,0151723736499906 -	0,0240647384480954 -
0,0174138357786711i	0,0277547615824412i	0,0174138357786711i	0,0277547615824412i
-0,0202947245417932 +	-0,0340653182363309 +	0,0202947245417932 -	0,0340653182363309 -
0,00944001667609834i	0,0267481313592149i	0,00944001667609834i	0,0267481313592149i
-0,0220970869120796 +	-0,0441941738241592 +	0,0220970869120796 +	0,0441941738241592 -
0,000000000000000i	0,0220970869120796i	0,000000000000000i	0,0220970869120796i
-0,0202634992658333 -	-0,0531558553002570 +	0,0202634992658333 +	0,0531558553002570 -
0,00965052038782779i	0,0138100924484116i	0,00965052038782779i	0,0138100924484116i
-0,0149258700973675 -	-0,0597050233153696 +	0,0149258700973675 +	0,0597050233153696 -
0,0182264490886032i	0,00234348216491694i	0,0182264490886032i	0,00234348216491694i
-0,00664567880142941 -	-0,0627970147746779 -	0,00664567880142941 +	0,0627970147746779 +
0,0245662340480268i	0,0114361338371013i	0,0245662340480268i	0,0114361338371013i
0,00365462812220650 -	-0,0617180514541196 -	-0,00365462812220650 +	0,0617180514541196 +
0,0277690701461626i	0,0263386122113050i	0,0277690701461626i	0,0263386122113050i

Y su gráfica:

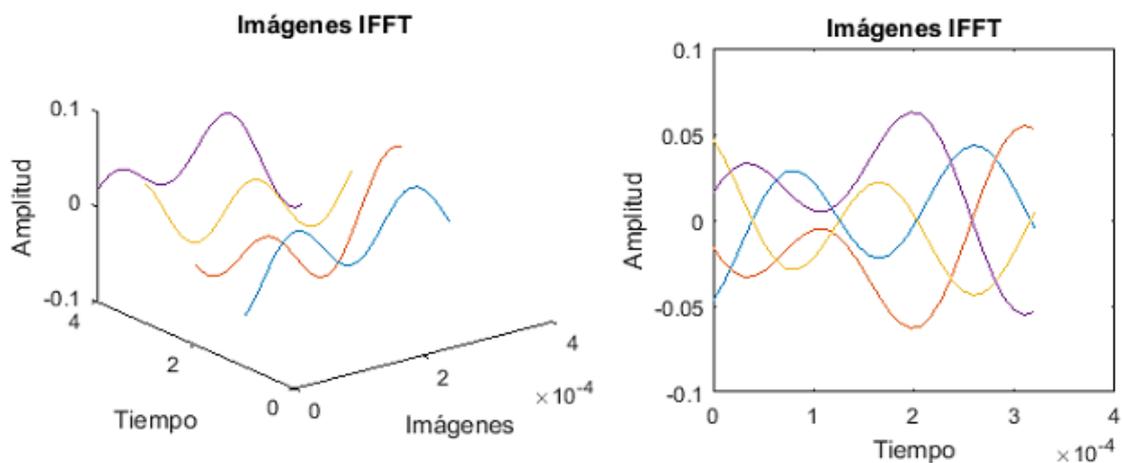


Figura 22. Aplicación de la IFFT del ejemplo citado utilizando MATLAB.

### 3.2.5 Inserción de intervalo de guarda.

El intervalo de guarda también llamado prefijo cíclico es una referencia que es añadida al inicio de cada símbolo generado que será transmitido. El intervalo de guarda es considerado como el tiempo serial de la onda y es llamado banda de base.

Éste prefijo es referenciado como  $T_G$  y es equivalente a la cantidad de términos de muestras, es decir:

$$T_{Sym} = T_{Sub} + T_G \quad (\text{Ecuación20})$$

De aquí se indica que la primera señal que llega en el tiempo  $t_0$  es:

$$\frac{1}{T_{Sub}} \int_0^{T_{Sub}} e^{j2\pi f_k(t-t_0)} e^{-j2\pi f_i(t-t_0-T_s)} dt = 0, \quad K \neq i \quad (\text{Ecuación21})$$

Y el resto de las imágenes:

$$\frac{1}{T_{Sub}} \int_0^{T_{Sub}} e^{j2\pi f_k(t-t_0)} e^{-j2\pi f_i(t-t_0)} dt = 0, \quad K \neq i \quad (\text{Ecuación22})$$

Esto quiere decir que el último símbolo  $N_G$  que es recibido es copiado y ubicado al inicio de cada símbolo asegurando la calidad de la ortogonalidad de las imágenes.

La salida de este proceso es escrita como:

$$\begin{aligned} \{\tilde{x}[n]\} &= \{\tilde{x}[-\mu], \dots, \tilde{x}[-1], \tilde{x}[0], \dots, \tilde{x}[N-1]\} \\ &= \{x[N-\mu], \dots, x[N-1], x[0], \dots, x[N-1]\} \end{aligned} \quad (\text{Ecuación23})$$

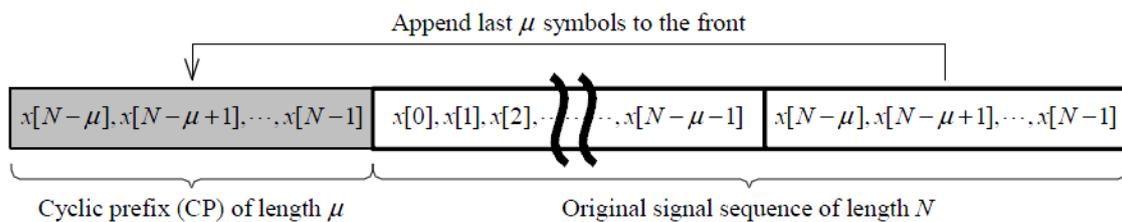


Figura 23. Aplicación del prefijo cíclico.

Tomado de: (Nguyen, 2015, pág. 8)

De nuestro ejemplo numérico, al repetir la última imagen como prefijo cíclico se obtiene:

Tabla 9.

*Aplicación del prefijo cíclico*

Imagen 1	Imagen 2	Imagen 3	Imagen 4
<b>-0,00474430789553133 +</b> <b>0,0507105664734335i</b>	<b>0,0532910087010150 +</b> <b>0,00126498432386767i</b>	<b>0,00474430789553133 -</b> <b>0,0507105664734335i</b>	<b>-0,0532910087010150 -</b> <b>0,00126498432386767i</b>
-0,0468750000000000 - 0,0156250000000000i	-0,0156250000000000 + 0,0468750000000000i	0,0468750000000000 + 0,0156250000000000i	0,0156250000000000 - 0,0468750000000000i
-0,0367620775869971 - 0,0269645700913457i	-0,0243841278637641 + 0,0402414011586608i	0,0367620775869971 + 0,0269645700913457i	0,0243841278637641 - 0,0402414011586608i
-0,0241430791658444 - 0,0343456331467416i	-0,0302813185502948 + 0,0317468769599113i	0,0241430791658444 + 0,0343456331467416i	0,0302813185502948 - 0,0317468769599113i
-0,0104343086748280 - 0,0372716651994509i	-0,0329497418278598 + 0,0225132753154761i	0,0104343086748280 + 0,0372716651994509i	0,0329497418278598 - 0,0225132753154761i
0,00288046986643839 - 0,0357346260696651i	-0,0324351847743090 + 0,0137069416596561i	-0,00288046986643839 + 0,0357346260696651i	0,0324351847743090 - 0,0137069416596561i
0,0144141711518017 - 0,0302069244624358i	-0,0291846981268509 + 0,00639372392760585i	-0,0144141711518017 + 0,0302069244624358i	0,0291846981268509 - 0,00639372392760585i
0,0230280229882013 - 0,0215795977539588i	-0,0239862094099731 + 0,00140671521211717i	-0,0230280229882013 + 0,0215795977539588i	0,0239862094099731 - 0,00140671521211717i
0,0279535515174761 - 0,0110540522751402i	-0,0178663629480437 - 0,000757282826680094i	-0,0279535515174761 + 0,0110540522751402i	0,0178663629480437 + 0,000757282826680094i
0,0288712353909777 + 0,000000000000000i	-0,0119588572614091 + 0,000000000000000i	-0,0288712353909777 + 0,000000000000000i	0,0119588572614091 + 0,000000000000000i
0,0259371768309350 + 0,0102039525431126i	-0,00735911245009858 + 0,00336502685135003i	-0,0259371768309350 - 0,0102039525431126i	0,00735911245009858 - 0,00336502685135003i
0,0197551491626649 + 0,0183286966378627i	-0,00498279039922235 + 0,00864997997872171i	-0,0197551491626649 - 0,0183286966378627i	0,00498279039922235 - 0,00864997997872171i
0,0112976757820507 + 0,0234338438973668i	-0,00544530538211215 + 0,0148768594110864i	-0,0112976757820507 - 0,0234338438973668i	0,00544530538211215 - 0,0148768594110864i

Imagen 1	Imagen 2	Imagen 3	Imagen 4
0,00178564476170802 + 0,0249719388785146i	-0,00897704242944240 + 0,0208983393273904i	-0,00178564476170802 - 0,0249719388785146i	0,00897704242944240 - 0,0208983393273904i
-0,00745931028136121 + 0,0228459548192016i	-0,0153848259074494 + 0,0255411229198919i	0,00745931028136121 - 0,0228459548192016i	0,0153848259074494 - 0,0255411229198919i
-0,0151723736499906 + 0,0174138357786711i	-0,0240647384480954 + 0,0277547615824412i	0,0151723736499906 - 0,0174138357786711i	0,0240647384480954 - 0,0277547615824412i
-0,0202947245417932 + 0,00944001667609834i	-0,0340653182363309 + 0,0267481313592149i	0,0202947245417932 - 0,00944001667609834i	0,0340653182363309 - 0,0267481313592149i
-0,0220970869120796 + 0,000000000000000i	-0,0441941738241592 + 0,0220970869120796i	0,0220970869120796 + 0,000000000000000i	0,0441941738241592 - 0,0220970869120796i
-0,0202634992658333 - 0,00965052038782779i	-0,0531558553002570 + 0,0138100924484116i	0,0202634992658333 + 0,00965052038782779i	0,0531558553002570 - 0,0138100924484116i
-0,0149258700973675 - 0,0182264490886032i	-0,0597050233153696 + 0,00234348216491694i	0,0149258700973675 + 0,0182264490886032i	0,0597050233153696 - 0,00234348216491694i
-0,00664567880142941 - 0,0245662340480268i	-0,0627970147746779 - 0,0114361338371013i	0,00664567880142941 + 0,0245662340480268i	0,0627970147746779 + 0,0114361338371013i

0,00365462812220650 - 0,0277690701461626i	-0,0617180514541196 - 0,0263386122113050i	-0,00365462812220650 + 0,0277690701461626i	0,0617180514541196 + 0,0263386122113050i
0,0148021337725934 - 0,0273004174470313i	-0,0561795667265103 - 0,0409766689657305i	-0,0148021337725934 + 0,0273004174470313i	0,0561795667265103 + 0,0409766689657305i
0,0255125783650832 - 0,0230536966739850i	-0,0463651837110701 - 0,0539177075064698i	-0,0255125783650832 + 0,0230536966739850i	0,0463651837110701 + 0,0539177075064698i
0,0345415804329565 - 0,0153612335902539i	-0,0329242985834651 - 0,0638437841152415i	-0,0345415804329565 + 0,0153612335902539i	0,0329242985834651 + 0,0638437841152415i
0,0408300926523868 - 0,00495352086815960i	-0,0169123781295687 - 0,0697013280433645i	-0,0408300926523868 + 0,00495352086815960i	0,0169123781295687 + 0,0697013280433645i
0,0436270126452268 + 0,00712805464157338i	0,000315743968669912 - 0,0708232813360228i	-0,0436270126452268 - 0,00712805464157338i	-0,000315743968669912 + 0,0708232813360228i
0,0425746515855823 + 0,0196544228725014i	0,0172478112164872 - 0,0670093897859008i	-0,0425746515855823 - 0,0196544228725014i	-0,0172478112164872 + 0,0670093897859008i
0,0377472160787060 + 0,0313532034552241i	0,0323754297908090 - 0,0585551111581135i	-0,0377472160787060 - 0,0313532034552241i	-0,0323754297908090 + 0,0585551111581135i
0,0296380535066610 + 0,0410570403769009i	0,0443564816722570 - 0,0462254650327555i	-0,0296380535066610 - 0,0410570403769009i	-0,0443564816722570 + 0,0462254650327555i
0,0190974867643278 + 0,0478383105682711i	0,0521602339398622 - 0,0311764534049759i	-0,0190974867643278 - 0,0478383105682711i	-0,0521602339398622 + 0,0311764534049759i
0,00722890469592220 + 0,0511143078777244i	0,0551786224741712 - 0,0148327024899891i	-0,00722890469592220 - 0,0511143078777244i	-0,0551786224741712 + 0,0148327024899891i
-0,00474430789553133 + 0,0507105664734335i	0,0532910087010150 + 0,00126498432386767i	0,00474430789553133 - 0,0507105664734335i	-0,0532910087010150 - 0,00126498432386767i

Utilizando el MATLAB se obtienen dos gráficas con las siguientes formas:

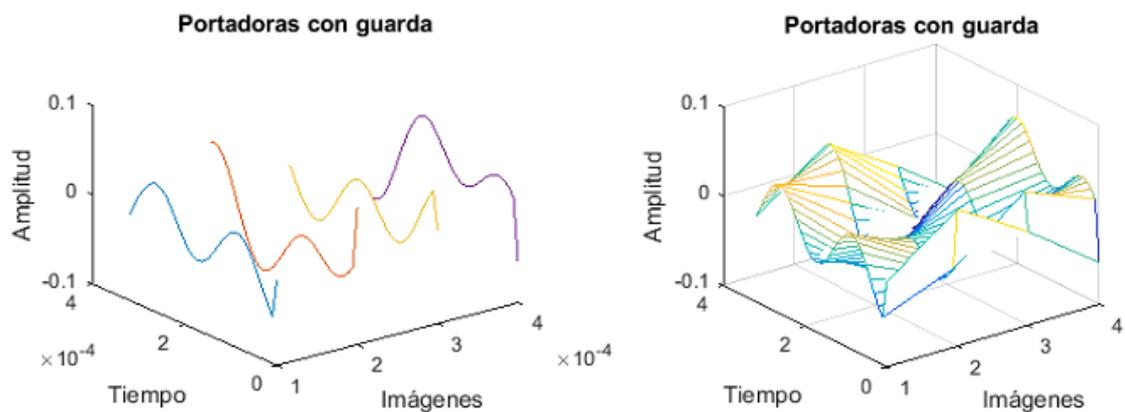


Figura 24. Incorporación del prefijo cíclico del ejemplo citado.

### 3.2.6 Conversor Paralelo-Serial.

Realiza el proceso de rearmado de los múltiples canales para obtener una sola imagen totalizada. Para establecer la señal resultante se toma cada imagen portadora con el desplazamiento del tiempo respectivo y se concatena.

Para el caso numérico las señales fueron concatenadas y se establece una representación gráfica superficial para permitir conservar la idea del proceso.

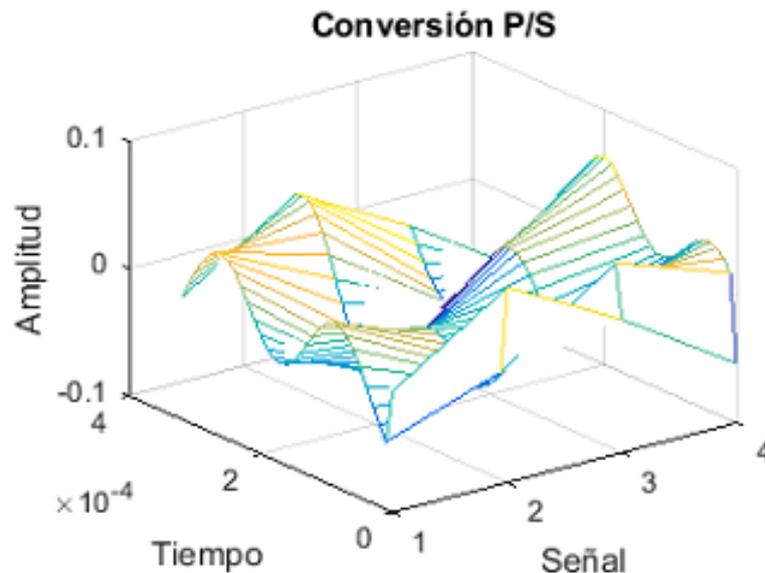


Figura 25. Gráfica de la conversión Serial-Paralelo

### 3.2.7 D/A.

En éste proceso la banda base resultante es mezclada y codificada para que contenga la información en el dominio analógica y pueda ser enviado a través de la antena, en pocas palabras la señal es interpretada como un pulso rectangular. En muchos casos se considera éste proceso como un filtrado aplicado a la señal resultado la siguiente señal:

La salida de éste proceso es:

$$S(t) = \tilde{x}(t) = \tilde{x}_I(t) + j\tilde{x}_Q(t) \quad (\text{Ecuación 24})$$

Un desarrollo gráfico es el siguiente:

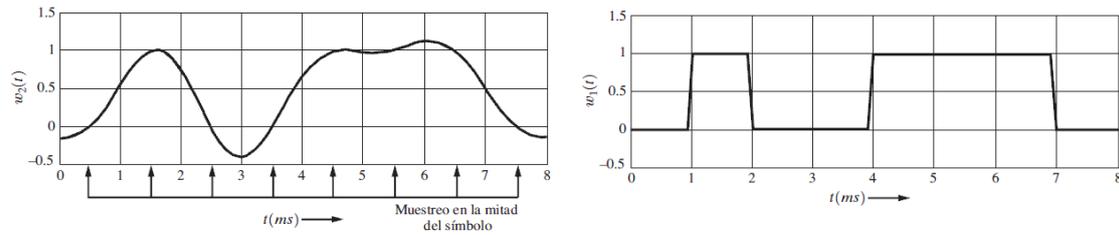


Figura 26. Codificación de la señal.

Tomado de: (Couch, 2007, pág. 156)

Para el proceso matemático de MATLAB no se desarrolla un cálculo en éste proceso, generalmente está contemplado en el desarrollo de la generación de la transmisión e incluso toda la información es manejada como un conjunto de valores complejos, es decir, valores reales e imaginarios.

### 3.2.8 Transmisor.

La preparación para la transmisión es la unificación de la señal como  $x(t)$  que es la manera como es reconocida. Luego de la modulación completa de la imagen se transmite hacia el medio por el uso del sistema de transmisión, generante formado por un conjunto de antenas.

Para el ejemplo numérico se utiliza las funciones gráficas utilizadas para las representaciones de señales de transmisión tanto de manera bidimensional y tridimensional:

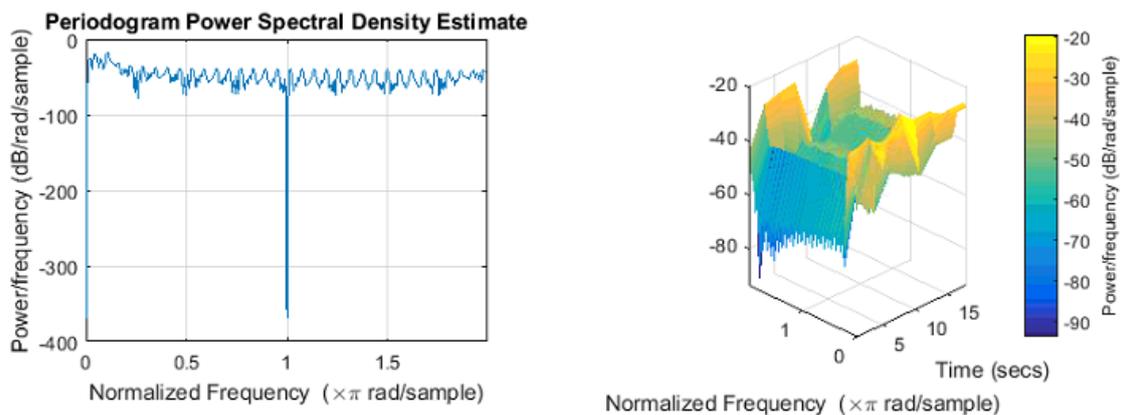


Figura 27. Representación de la señal OFDM transmitida en MATLAB

### 3.2.9 Canal.

En ésta etapa interviene el elemento de la comunicación que portará señal, constituido por cables, antenas, el espacio de trasmisión y todas las variantes que afectan dicha señal como el ruido, el retraso, entre otros.

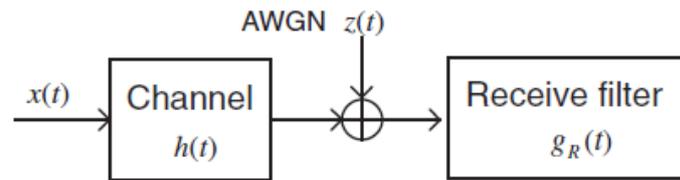


Figura 28. Inclusión de ruido en la señal.

Tomado de: (Cho, 2010, pág. 112)

El símbolo que sale del canal es:

$$Y_{n,K} = H_{n,K} \cdot X_{n,K} + N_{n,K} \quad (\text{Ecuación 25})$$

El valor  $N_{n,K}$  también es conocido como  $w(t)$  o ruido Gaussiano, y  $H_{n,K}$  es el impulso de respuesta del canal.

El MATLAB permite insertar un ruido generado por la transmisión de una señal a través de un cable AWGN y se utiliza un valor constante de 0,2 de radio de ruido por bit transmitido. Esto permite representar el concepto del ruido Gaussiano en el canal. La representación gráfica es:

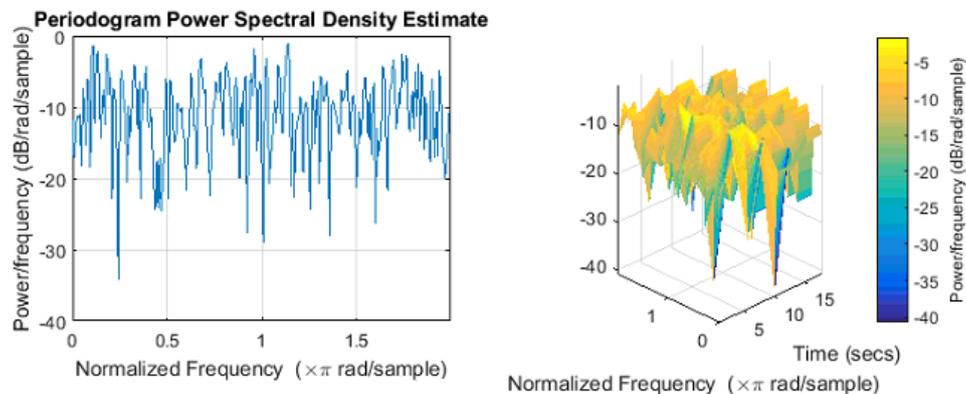


Figura 29. Representación de inserción de ruido

Los parámetros que son utilizados en la modulación QPSK y 16 QAM son:

Tabla 10.

Parámetros QPSK y 16 QAM.

Parámetro	Valor
Longitud del símbolo	1280
Tamaño del IFFT	128
Número de portadoras	128
Símbolos OFDM por marco	10

## 4. CAPITULO IV. DIAGRAMA DE FLUJO

### 4.1. Diagrama OFDM generalizado.

Todo sistema de transmisión de datos posee una estructura general que simplifica el entendimiento de sus procesos, en el caso del OFDM se puede establecer un diagrama general representado de la siguiente forma:

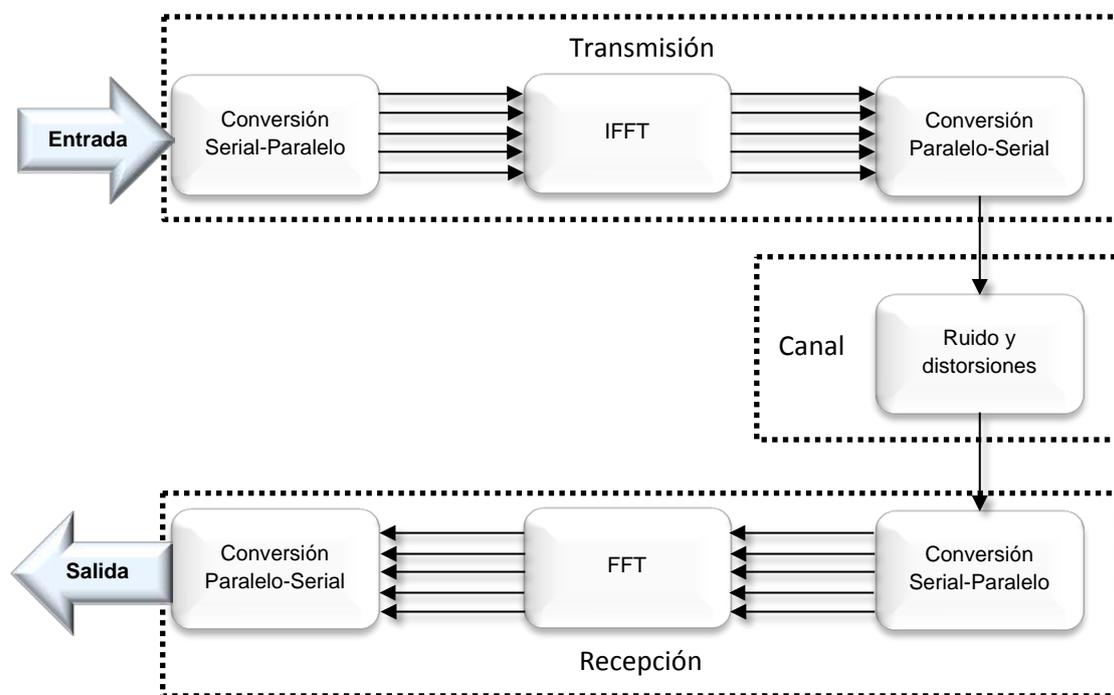


Figura 30. Diagrama de modulación OFDM.

La información que se desea modular es un conjunto de datos que se pueden encontrar codificados o no, ésta será introducida en el proceso de Transmisión que en términos generales es separada en múltiples imágenes que poseen una longitud de n-bit que son tomados por el mapeador 16 QAM, la transformada rápida de Fourier inversa le asigna una subportadora a cada imagen, cada una de ellas a diferente frecuencia, esto quiere decir que toda la información se ha dividido en pequeños canales de transmisión y se ha formado la modulación por OFDM.

Todos los subcanales son tratados con el conversor de paralelo a serial, en este paso se encuentran otros procesos que serán explicados con más detalles. La información se encuentra preparada para su transmisión por el canal; durante el paso de la señal transmitida por el canal, ésta recogerá un conjunto de variaciones como la distorsión, el ruido y el desvanecimiento multitrayectoria.

En el proceso de recepción se cumple una fase inversa de la transmisión, implícitamente al recibir la señal se debe filtrar para eliminar las variaciones del canal por medio de un filtrado, para que luego se use el conversor de serial a paralelo y dividir la imagen recibida a múltiples imágenes para su tratado. La transformada rápida de Fourier removerá las subportadoras de cada imagen y se recompone la señal por medio del conversor de paralelo a serial, aquí se implica la remoción del mapeo; el resultado de éste proceso es la información original que fue introducida al sistema.

#### **4.2. Proceso de Transmisión.**

En los capítulos anteriores se ha indicado la selección de la modulación 16 QAM combinado con la modulación OFDM, más adelante en el capítulo 4 se realiza la comparación entre algunos programas muy utilizados para desarrollar simuladores entre los cuales el MATLAB es la selección de preferencia. Además de lo indicado, se presenta en éste proyecto el estudio solamente de la sección de transmisión, y éste proceso está conformado por:

#### **4.3. Generación de la imagen de bit.**

Para establecer la modulación OFDM se debe establecer un ingreso de datos binarios utilizando los elementos de interfaz de MATLAB, para ello se establece el siguiente diagrama de flujo de éste proceso.

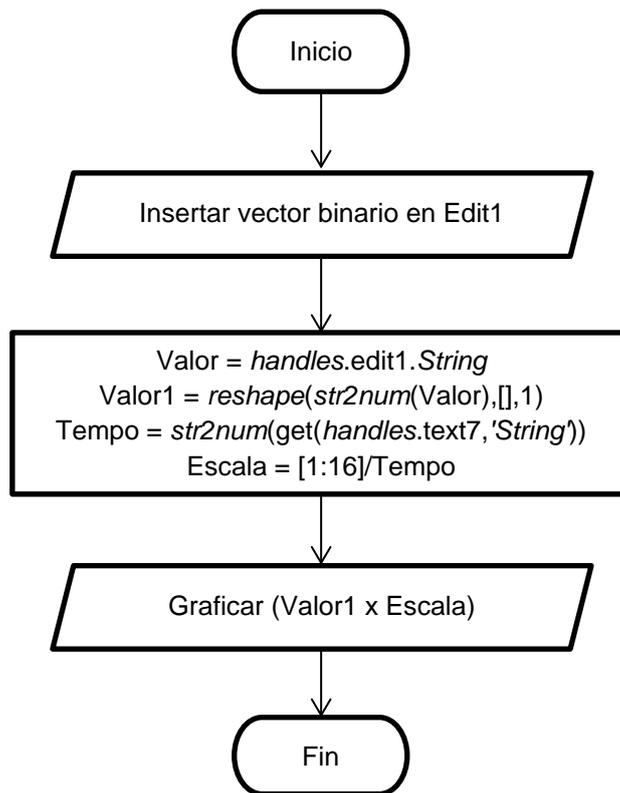


Figura 31. Diagrama de flujo de la generación de la imagen de bit.

La imagen que debe ser ingresada en el texto de **Edit1** debe ser un vector binario compuesto por 16 bits que facilitará el estudio realizado, este dato es recogido por la programación de MATLAB para ser almacenado como la variable **Valor**. Seguidamente la variable Valor es establecida como un vector unitario utilizando la función *reshape* de MATLAB. La velocidad de transmisión es tomada del texto de **Text7** y almacenada como la variable **Tempo**, el cual se convierte en el valor para dividir a un vector de 16 unidades para formar el vector de tiempo de bit denotado como **Escala**. Se representa la gráfica del vector **Valor** vs vector **Escala**. La función *str2num* permite la conversión de un texto a número.

#### 4.4. Conversor S/P.

A partir del ingreso de la información binaria al sistema, el conversor recibe la señal y realiza la separación de la imagen en N cantidad de distribuciones, recordando que N es la cantidad de portadoras que serán utilizadas, en el capítulo 2 se explica cómo se desarrolla este proceso mediante un ejemplo numérico. El diagrama de flujo correspondiente a éste bloque es:

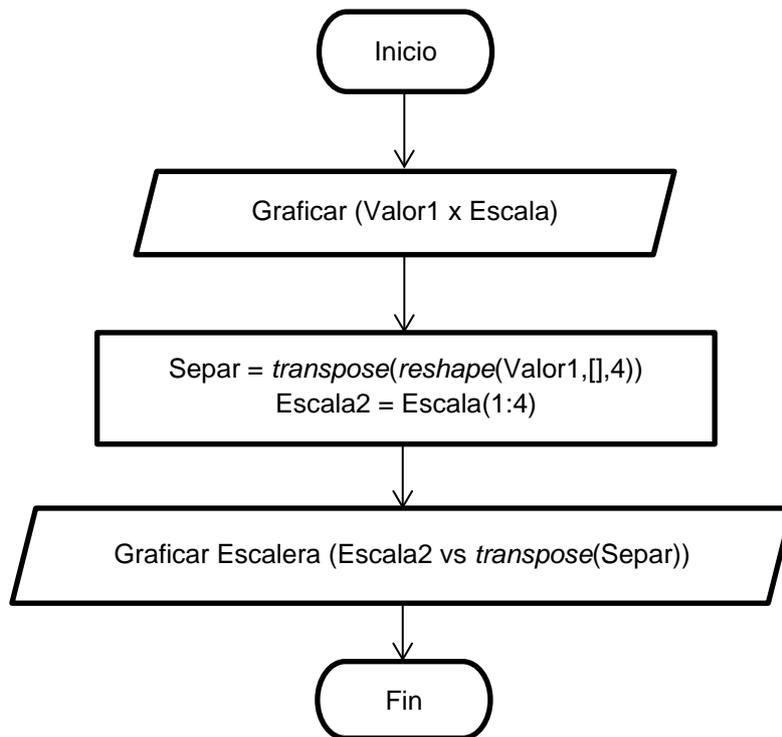


Figura 32. Diagrama de flujo de la generación de la conversión S/P.

Como se puede observar **Graficar** indica que se realiza la representación gráfica del ingreso del bloque al inicio de diagrama de flujo. Luego se aplica la función `reshape` de MATLAB para realizar la división del vector de 16 bits para obtener una matriz de cuatro filas y cuatro columnas llamado **Separ** que corresponde con la conversión de serial a paralelo previo a la aplicación del 16 QAM. La variable **Escala2** toma los cuatro primeros términos del vector **Escala** correspondiente al tiempo de la transmisión. La gráfica Escalera presentada es la equivalente a la función gráfica `stairs` de MATLAB que toma los valores de un vector o matriz y realiza su representación de forma escalera. Su diagrama de bloque es la siguiente:

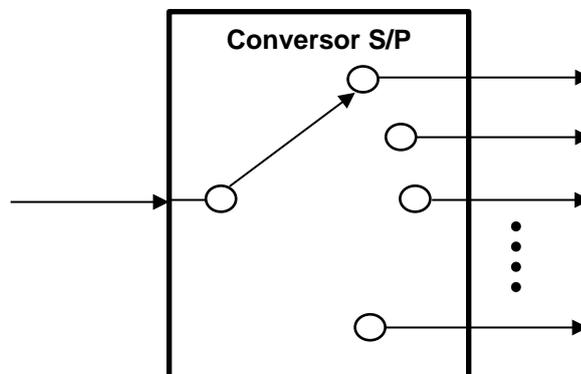


Figura 33. Detalle de la conversión serial a paralelo.

#### 4.5. Modulación 16 QAM.

El mapeador 16 QAM es quien modula la amplitud de una portadora en cuadratura. La modulación se encuentra explicada con sus detalles en el capítulo dos incluyendo una representación numérica.

El diagrama de flujo de la modulación 16 QAM, se encuentra presentada en la siguiente figura.

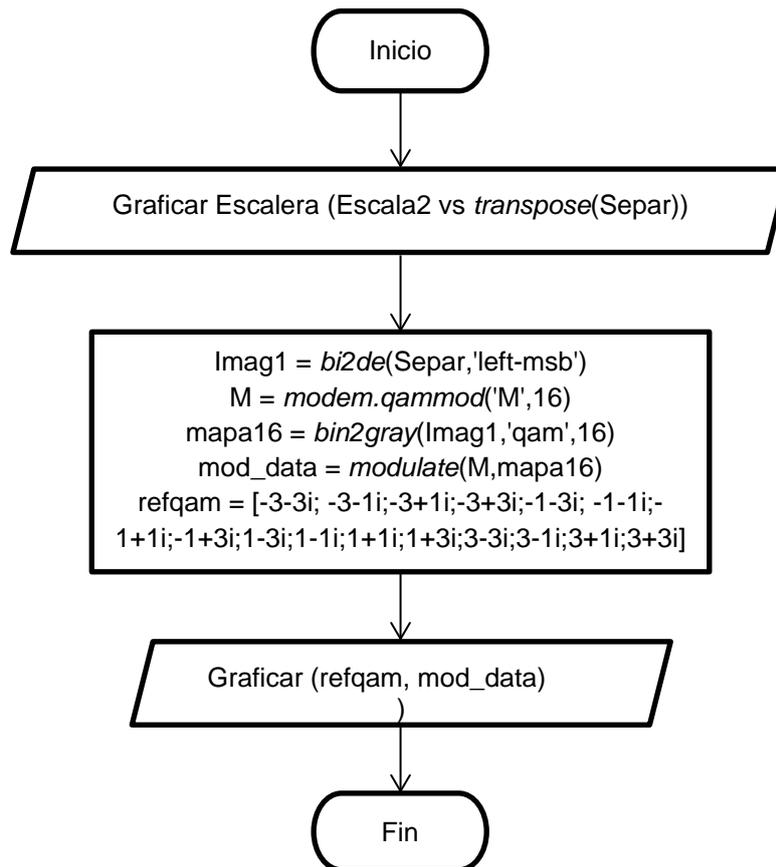


Figura 34. Diagrama de flujo del 16 QAM.

Las funciones de MATLAB desarrolladas en éste bloque son totalmente específicas para obtener ésta modulación. La función *bi2de* permite realizar la conversión de la matriz binaria **Separ** a decimal para registrarse como **Imag1** que es un vector. La función *modem.qammod* combinada internamente con *modulate* de MATLAB son las funciones puras para la modulación QAM y al usar la numeración 16 desarrolla exactamente la 16 QAM de la variable **Imag1**. La función *bin2gray* realiza la conversión de un valor entero a código Gray **Imag1** necesario para dicha modulación. La constante **refqam** posee los valores de referencia de la modulación utilizada. Seguidamente se realiza la representación gráfica de la modulación. La estructura interna de un canal del mapeador se puede representar de la siguiente manera:

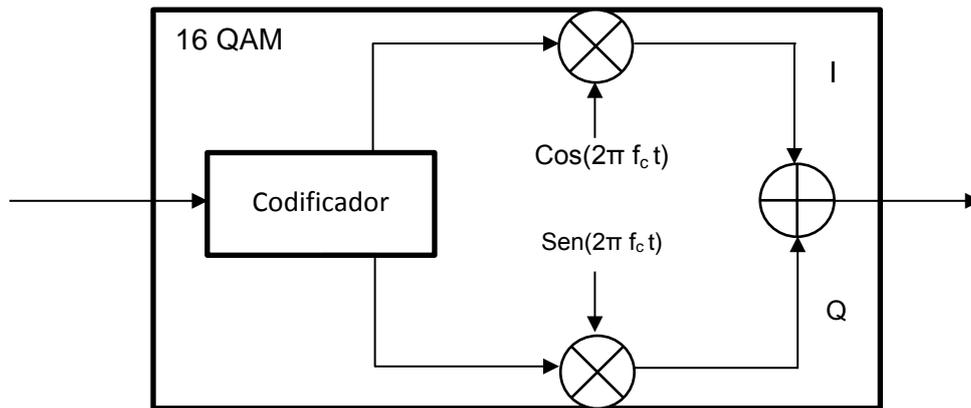


Figura 35. Detalle de la modulación 16 QAM

#### 4.6. Piloto.

La inserción de los pilotos se realiza a través de una separación de las señales como fue representado en el capítulo anterior. Su diagrama de flujo es:

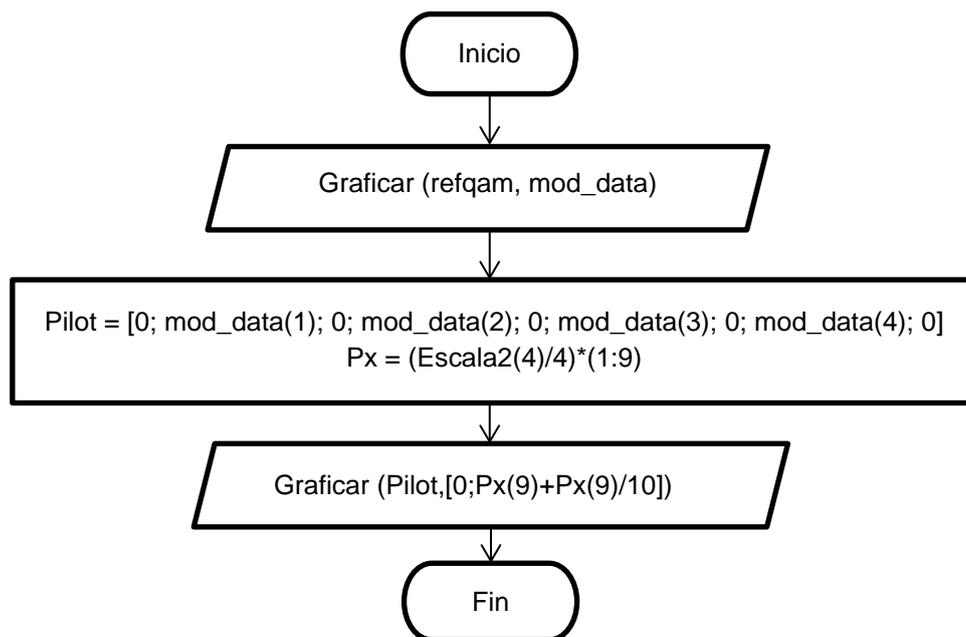


Figura 36. Diagrama de flujo del 16 QAM

Después de graficar los datos de entrada, se aplica una programación simple la cual permite seleccionar los elementos de un vector de manera independiente, mediante la sintaxis *mod\_data(1)* se toma el valor del primer elemento del vector **mod\_data**. Al encerrar un conjunto de valores numéricos dentro de los corchetes y separados por punto y coma se obtiene un nuevo vector denotado **Pilot** y representa el vector resultante de la inserción de los pilotos para realizar su gráfica. La representación en diagrama de bloque del proceso es:

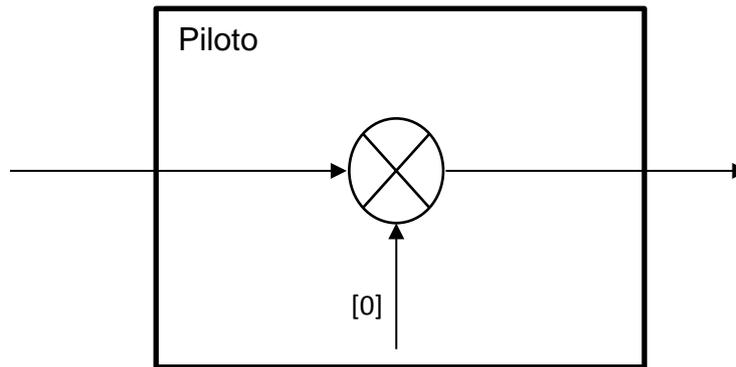


Figura 37. Detalle de la aplicación de los pilotos

#### 4.7. IFFT.

La transformada rápida de Fourier inversa es la modulación base del OFDM cuyo proceso es explicado en el capítulo 2 con un ejemplo numérico. El diagrama de bloque se puede representar de éste modo:

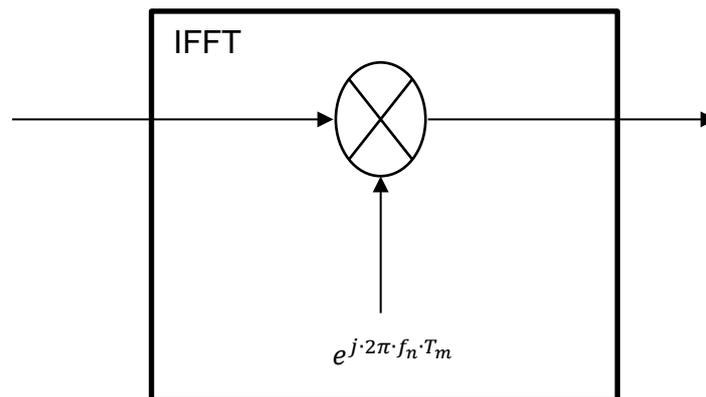


Figura 38. Aplicación de la transformada rápida de Fourier inversa

El diagrama de flujo de éste bloque es:

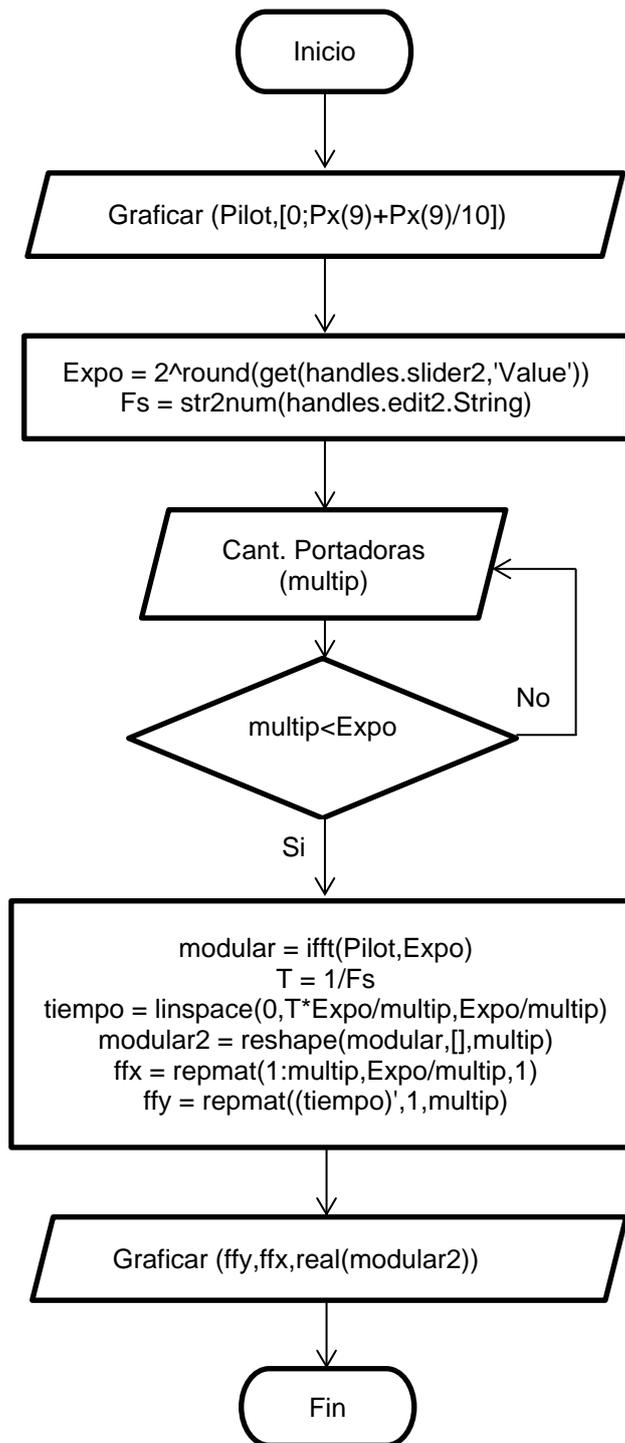


Figura 39. Diagrama de flujo de la IFFT

Inicialmente se realiza la gráfica de los datos de entrada. Se lee el valor de la posición de la barra deslizante **Slide3** como un valor entero mediante la fórmula *round* de MATLAB y se utiliza como exponente de una base 2 para calcular la longitud de la IFFT con el nombre de **Expo**. Se lee el texto de **Edit2** y se convierte en número con la función *str2num* para almacenarlo como **Fs**. La cantidad de portadoras puede ser seleccionada de **Popupmenu1**, pero en el diagrama de flujo debe estar presente la condición de que dicho valor debe ser

menor que la longitud de la IFFT guardada como **Expo**, si no se cumple la condición no se puede proceder. La función *ifft* de MMATLAB de genera la aplicación de la IFFT tomando el vector **Pilot** y el resultado del proceso está programado para utilizar la longitud de magnitud **Expo** almacenado con el nombre **modular**. Se calcula una fracción de tiempo **T** para obtener un vector **tiempo**. El cálculo de la variable **modular2** es realizado con la división del vector resultante de la IFFT por la cantidad de portadoras seleccionadas. Las matrices **ffx** y **ffy** son arreglos con dimensiones idénticas a la matriz **modular2**, pero con valore de posición y tiempo respectivamente para la generación de la gráfica del proceso.

#### 4.8. Prefijo cíclico.

Para añadir el prefijo cíclico la teoría indica que se duplica la última imagen recibida, éste proceso se explica en el capítulo anterior y el proceso puede ser representado así:

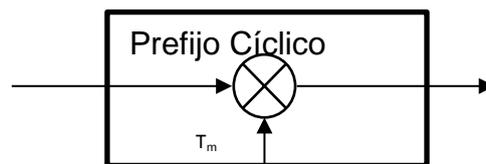


Figura 40. Detalle de la aplicación del prefijo cíclico

El diagrama de la aplicación de la guarda es el siguiente:

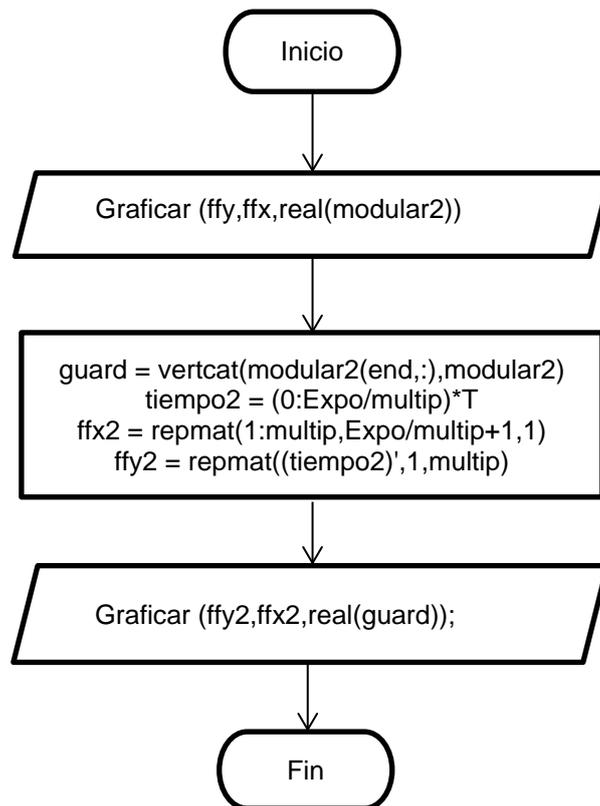


Figura 41. Diagrama de flujo de la guarda

Este proceso solo requiere de la aplicación de la fórmula *vertcat* de MATLAB que permite seleccionar la última fila de la matriz **modular2** y ubicarla como primera fila como una nueva matriz nombrada **guard**. Debido al incremento de la matriz se recalcula el tiempo y se generan las matrices **ffx2** y **ffx2** para establecer las posiciones y el tiempo de las gráficas tridimensionales. El resultado gráfico de éste proceso son dos generaciones tridimensionales realizadas con funciones diferentes de MATLAB las cuales son *plot3* y *mesh*.

#### 4.9. Conversor P/S.

Éste conversor realiza un proceso de unión de las imágenes entrantes, es el proceso contrario de la conversión de serial a paralelo, su explicación está en el capítulo 2. El diagrama de flujo para desarrollar el presente bloque es:

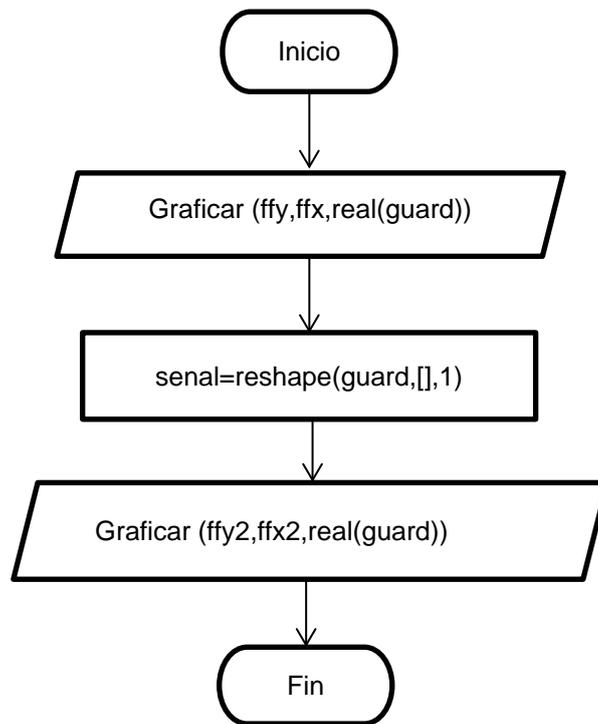


Figura 42. Diagrama de flujo de la conversión P/S.

En éste proceso sólo se realizan las gráficas de los datos de entrada del proceso, se aplica la función *reshape* de MATLAB para realizar la concatenación de la matriz de múltiples filas como un vector nombrado **senal**, y posteriormente se realiza la gráfica en representación de la unión de las portadoras. Su diagrama de bloque se representa como:

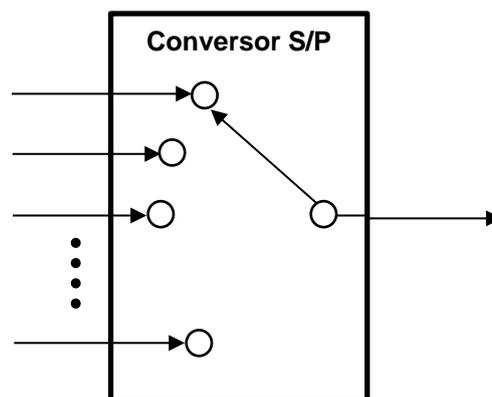


Figura 43. Detalle de la conversión paralelo a serial

#### 4.10. Transmisión.

La transmisión es el proceso final del estudio y como ha sido indicado en el capítulo anterior, se realiza la preparación de la señal para su envío y su diagrama de flujo es:

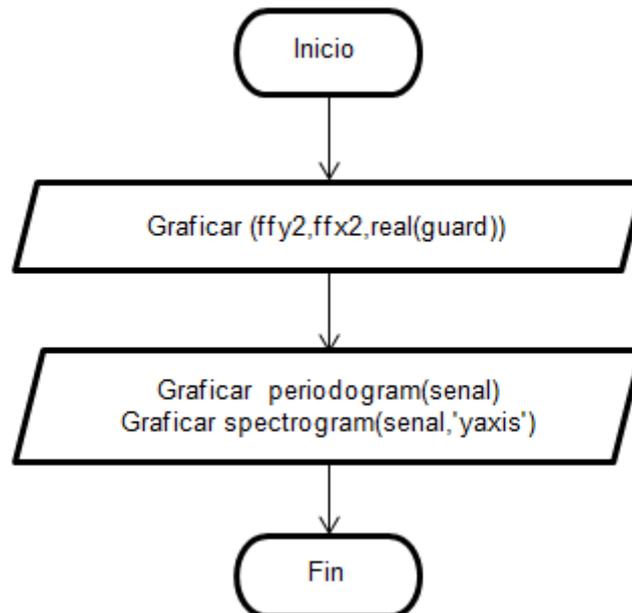


Figura 44. Diagrama de flujo de la transmisión.

Luego de realizar las representaciones gráficas de los datos de entrada se utiliza el mismo vector **señal** y se representa aplicando dos generadores gráficos avanzados del MATLAB, el primero es específicamente para el estudio de señales *periodogram* y el segundo es el *spectrogram*. Estas dos funciones permiten que el MATLAB posea aplicaciones en el campo de las telecomunicaciones. El diagrama de bloque de este proceso es el siguiente:



Figura 45. Representación la transmisión.

#### 4.11. Canal.

A pesar de no formar parte del estudio, se ha decidido incluir el canal como un elemento de cableado AWGN por el cual se inserta un ruido, su explicación se encuentra en el capítulo 2 y el diagrama de flujo correspondiente con éste proceso es:

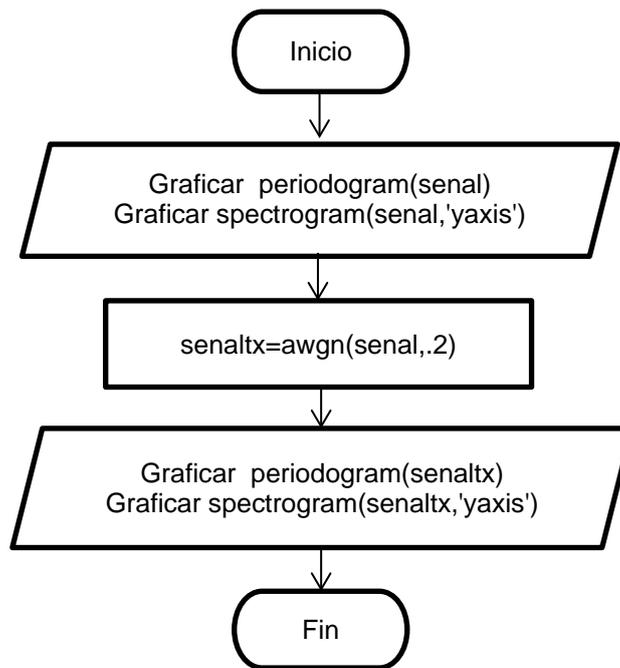


Figura 46. Diagrama de flujo del canal.

Para el proceso de transmisión se utilizan las mismas representaciones gráficas del proceso anterior pero se establece el cálculo de la interferencia por ruido del canal cuando se utiliza un cable AWGN, la función de MATLAB que permite introducir un ruido es *awgn*, se aplicó un valor de 0,2 y se registra con el nombre de **senaltx**. Su diagrama de bloque es:

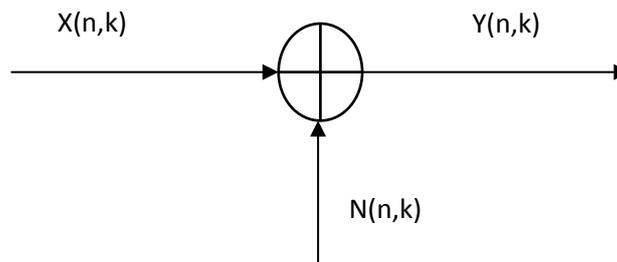


Figura 47. Representación de la inserción del ruido en el canal

#### 4.12. Sistema de transmisión.

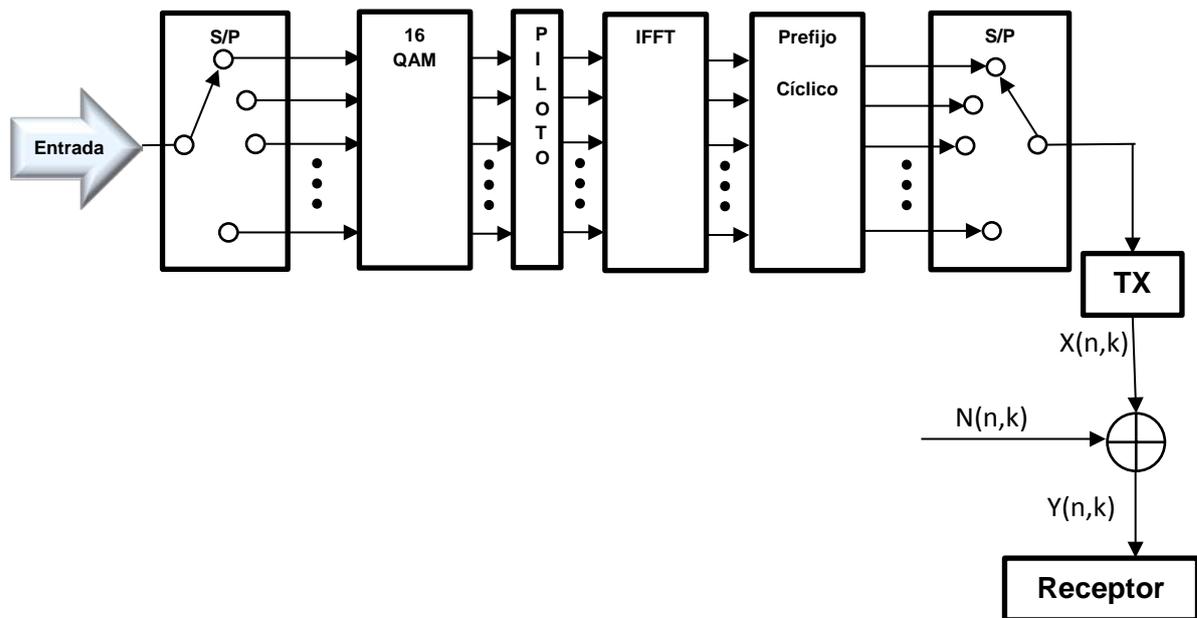


Figura 48. Diagrama de bloque de la modulación 16 QAM y OFDM.

## 5. CAPÍTULO V. SIMULADOR OFDM

Luego de iniciar el MATLAB se crea un nuevo archivo y se guarda con el nombre "Control". Éste documento es el encargado de poseer la codificación completa del simulador, todos los cálculos matemáticos y controles de componentes.

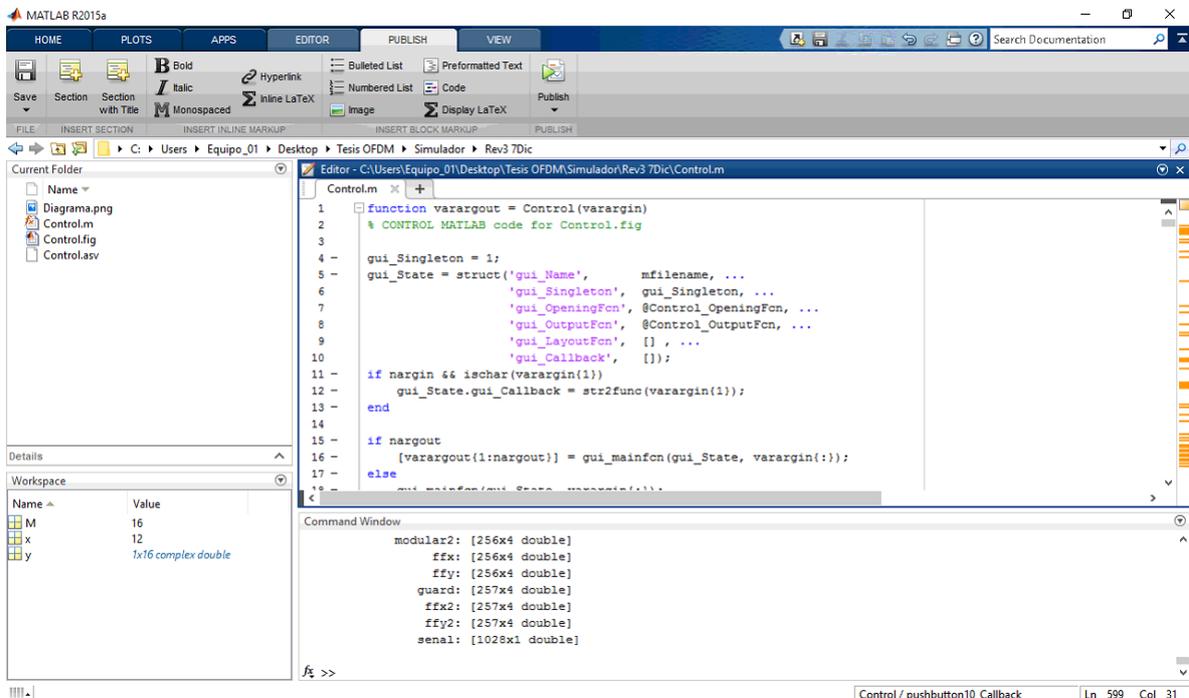


Figura 49. Entorno de programación del MATLAB



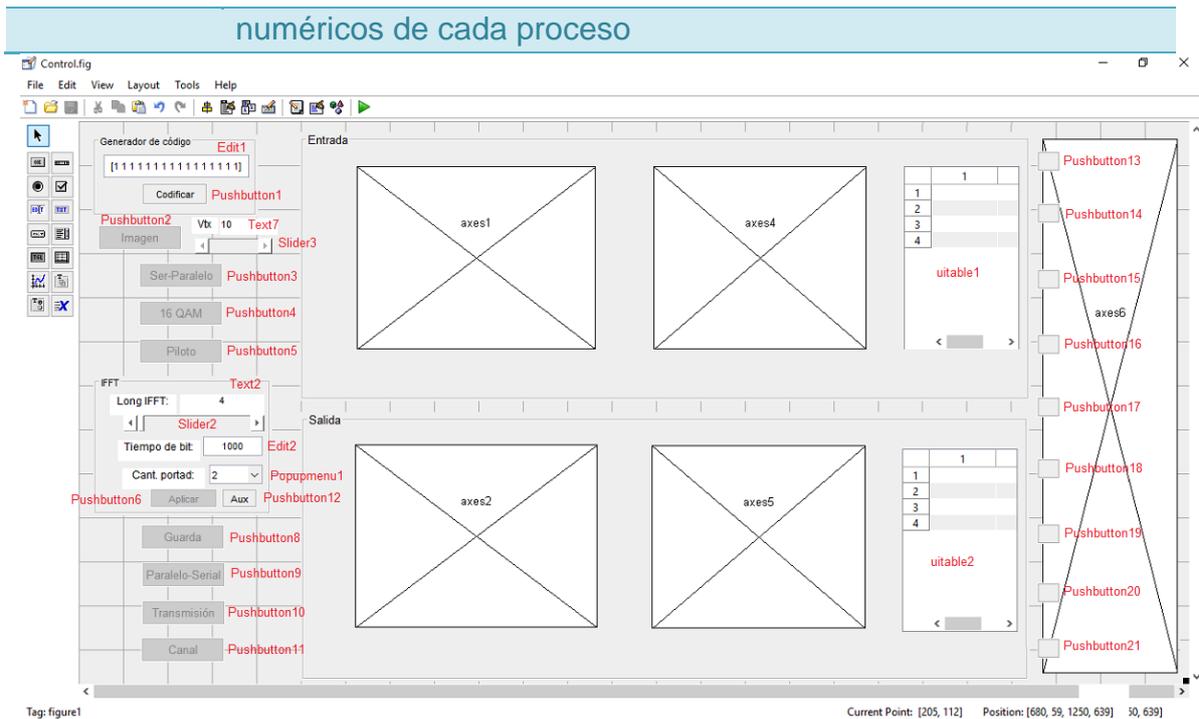


Figura 51. Indicación del nombre de cada elemento

### 5.1. Código de inicio de carga de la ventana.

Automáticamente MATLAB genera un código para establecer los parámetros de inicio de la ventana y de enlace con los elementos. Para el simulador se estableció la demostración del diagrama del proceso de estudio al lado derecho de la ventana. La codificación está en el Anexo II sección 1:

### 5.2 Generación del código de imagen.

Estos elementos se encargan del ingreso de un texto binario que será utilizado por el simulador y de control inicial de los botones. Está formado por los elementos de la figura:

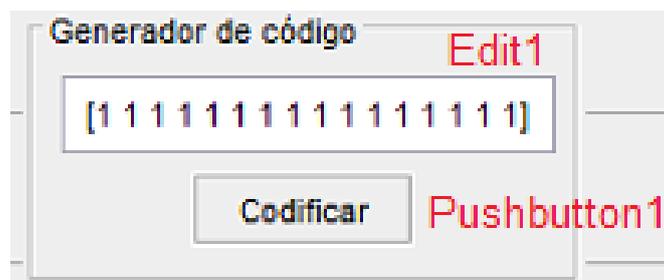


Figura 52. Bloque generador de código

La codificación del **botón1** está en el Anexo II sección 2:

En éste código se debe tomar en cuenta que éste botón es el único que lee los bits de entrada para ser registrado como la variable **Valor** y puede inhabilitar a los demás botones para controlar proceso. Éste botón realiza un llamado al siguiente botón con el objetivo de mostrar un resultado gráfico de los bits ingresados.

### 5.3 Generación de la imagen.

Este bloque establece la primera representación gráfica del simulador que son los ingresos de los bits. Lo constituyen los siguientes elementos de interacción:



Figura 53. Bloque de Imagen

La codificación del **Slider3** está en el Anexo II sección 3:

Como se puede apreciar en el anexo el Slider3 sólo es utilizado para expresar un valor numérico en la etiqueta **Text7**. La codificación del **boton2** está en el Anexo II sección 4:

La función principal de éste proceso es la conversión del texto **Valor** a vector binario y la captura del valor numérico indicado en Text7 para generar la velocidad del ingreso del bit, ambos valores son graficados y exportados para ser usados en el siguiente bloque.

### 5.4. Conversión de serial a paralelo.

Establece la división del código de entrada de 16 bits a cuatro elementos de 4 bits cada uno. Esto permitirá preparar los datos de ingreso para la modulación 16 QAM. El elemento de interfaz en el **botón3**.



Figura 54. Bloque conversión serial-paralelo

Su codificación esta adjunta en el Anexo II sección 5.

A partir de éste proceso se inicia los desplazamientos de las gráficas y datos hacia la ubicación superior central de la ventana para facilitar la visualización en la salida y entrada del proceso. Se prepara la representación de la división del vector entrante en cuadro imágenes y se presenta una gráfica 3D para mejorar su visualización.

### 5.5. Modulación 16 QAM.

MATLAB posee un conjunto de funciones para desarrollar las modulaciones, en éste bloque se representa la modulación 16 QAM, para ello se establecen los siguientes elementos de interacción:

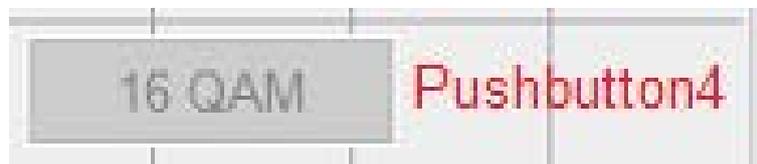


Figura 55. Bloque modulación 16 QAM

La codificación de éste botón esta adjunta en el Anexo II sección 6.

La secuencia de todos los bloques será siempre el establecimiento de la entrada, procesamiento y salida de datos. En éste bloque se realiza la función de modulación QAM establecida para 16 QAM debido a su facilidad de explicación académica como ha sido explicado anteriormente.

### 5.6. Inserción de pilotos.

Para éste bloque se toma el vector resultante de la modulación 16 QAM que siempre estará contemplada por un vector de cuatro (04) elementos, debe separarlos e introducir valores nulos al inicio, entre los valores y al final. El elemento de interfaz es el siguiente:



Figura 56. Bloque de la inserción del piloto

La codificación del **botón5** esta adjunta en el Anexo II sección 7:

El vector de datos resultante de la inserción de piloto contiene nueve (09) valores numéricos y debido a eso su gráfica son líneas de punto a punto. Ése valor del vector es exportado para la generación de la modulación IFFT.

### 5.7. Modulación IFFT.

Con el ingreso del vector Pilot se realiza su gráfica y representación de los valores numéricos en la tabla de entrada. Este bloque posee diversos elementos de interface ya que se pueden gestionar diferentes variables, esta es su estructura:



Figura 57. Bloque de modulación IFFT

La primera variable es la longitud de la IFFT, MATLAB permite establecer la longitud del vector obtenida de la modulación, para controlar el valor se utilizó el **Slider2**, cuando se manipula envía un valor numérico a la etiqueta **Text2**, su código esta adjunta en el Anexo II sección 8.

La siguiente variable es el tiempo de control de la salida del IFFT, sólo permite al usuario establecer un valor numérico de entrada.

Para establecer la cantidad de portadoras que se usa en la IFFT se aplica el **Popupmenu1**, los valores numéricos que presentan son controlados por el **Slider2**, su código es sólo el de creación y es realizado automáticamente por MATLAB.

El botón6 es el elemento que aplica la función IFFT de MATLAB, recoge los valores presentados en el **Text2** para la modulación, el valor de **Edit2** para establecer la variación del tiempo expresada en las gráficas y el valor de **Popupmenu1** para dividir el vector del IFFT resultante y establecer las

portadoras de salida. La codificación del **botón6** esta adjunta en el anexo II sección 9.

Con este control se puede obtener una variación de salidas del IFFT para un mismo valor de entrada se señal después de la inserción del piloto, esto permite poder analizar los cambios de las variables asociadas.

Junto a éste bloque se aplicó el **botón12** que permite representar una gráfica auxiliar en tridimensional del proceso resultante de las portadoras de IFFT y su código está en el Anexo II sección 10:

### 5.8. Inserción de la guarda.

Este bloque permite duplicar la última fila del vector resultante de la IFFT, este proceso puede ser fácilmente evaluado en las tablas de datos y por las gráficas presentadas. Está formado por el siguiente elemento:



Figura 58. Bloque de inserción de guarda

Y la codificación del **boton8** esta adjunta en el anexo II sección 11.

La salida del bloque permite representar la guarda en un espacio tridimensional simple. La matriz resultante posee una fila más que la de entrada.

### 5.9. Conversión paralela a serial.

La aplicación de la conversión realizada por MATLAB se realiza a través de una concatenación de las portadoras, su elemento de control es el siguiente:



Figura 59. Bloque de conversión paralelo-serial.

La codificación del **Botón9** esta adjunta en el anexo II sección 11.

La gráfica asociada con éste bloque es simplemente una capa que una las portadoras que ingresan para facilitar el entendimiento del proceso con la visualización de su gráfica tridimensional.

### 5.10. Transmisión.

El proceso de transmisión de la señal resultante es facilitado por MATLAB simplemente por la generación de funciones gráficas para el estudio de señales, el elemento que permite controlar éste proceso es:



*Figura 60.* Bloque de la transmisión

Y la codificación del **botón10** esta adjunta en el Anexo II sección 12.

La representación en tres dimensiones de la señal transmitida es importante para ayudar a idealizar el proceso.

### 5.11. Canal.

Este bloque se ha establecido como un adicional para presentar el cambio en la señal por la presencia del ruido en la señal, el elemento de control es:



*Figura 61.* Bloque del canal

Y la codificación del **botón11** esta adjunta en el anexo II sección 13.

La representación final del simulador permite la comparación de la señal por la interferencia sufrida durante el paso por un canal, se utilizó la función AWGN de MATLAB que permite introducir un ruido controlado para luego graficarlo.

### 5.12. Elementos visuales.

Para facilitar la visualización del bloque actual del proceso se han introducido un conjunto de nueve (09) botones, éstos no poseen codificación alguna por lo tanto no tienen acciones, sólo se manipula un color de fondo para representar la ubicación actual. La acción de cambio de los colores se realiza desde el inicio de los botones de cada bloque. Los elementos que lo forman son:

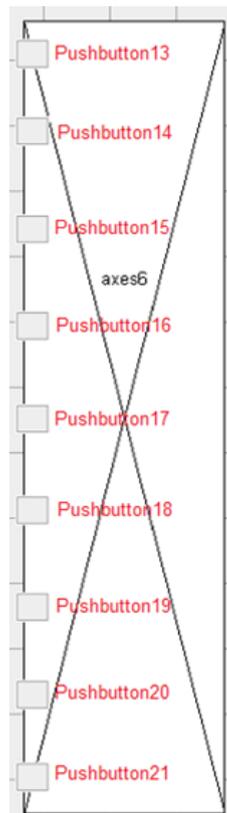


Figura 62. Sección de los elementos visuales

El elemento **axes6** permite introducir la imagen de bloque del proceso ya preestablecido. La ubicación es a la derecha de la ventana y no posee otra función.

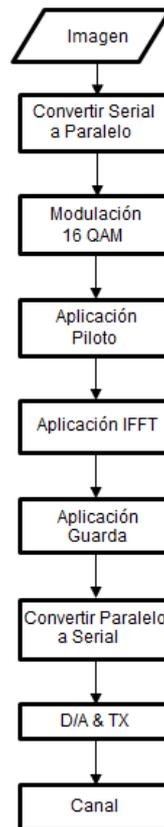


Figura 63. Diagrama de bloque del simulador

Luego de armar el simulador se inicia para obtener la ventana resultante:

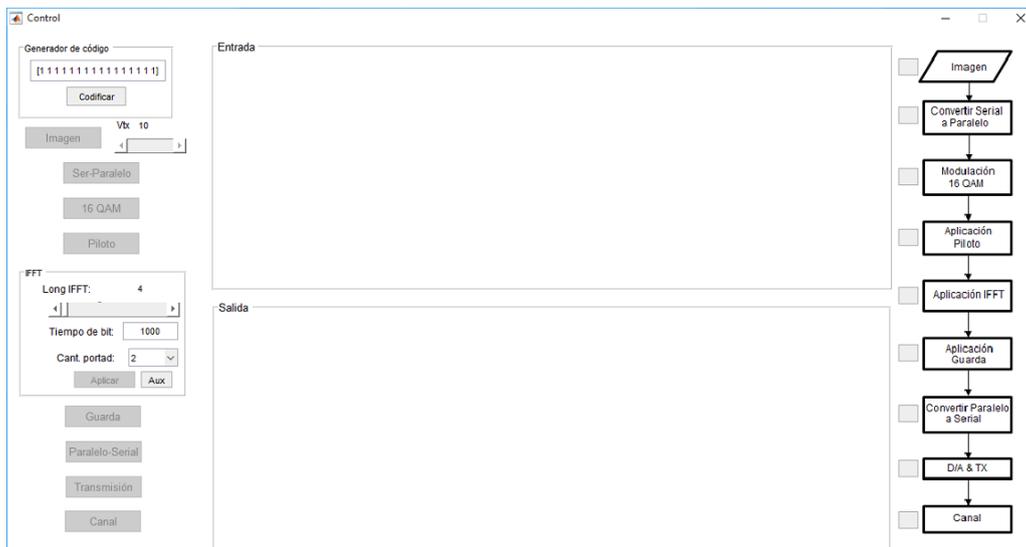


Figura 64. Estado de inicio del simulador

Hay que tomar en cuenta que el ingreso de bit está limitado a 16 bits binarios, por lo tanto, se ha establecido un ingreso automático de valor unitario donde se deben reemplazar por ceros (0) en las posiciones correspondiente, esto es por

el motivo de que la modulación ha sido desarrollada como una combinación del 16 QAM y del OFDM como ha sido indicado en éste proyecto.

## 6. CAPÍTULO VI. PRUEBAS DE FUNCIONAMIENTO

Una vez se ha iniciado el simulador se procede a realizar un conjunto de pruebas con tres valores de ingresos binarios diferentes. A partir de esos valores se realizarán los procesos de modulación y se demostrarán las gráficas resultantes para realizar comparaciones.

El proceso de pruebas estará estructurado siguiendo el diagrama de bloques del simulador y se irá indicando los cambios de las variables posibles de cada etapa.

### 6.1. Datos de prueba.

Para controlar las pruebas se seleccionaron los siguientes arreglos de bits:

$$A = [1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1]$$

$$B = [0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0]$$

$$C = [1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1]$$

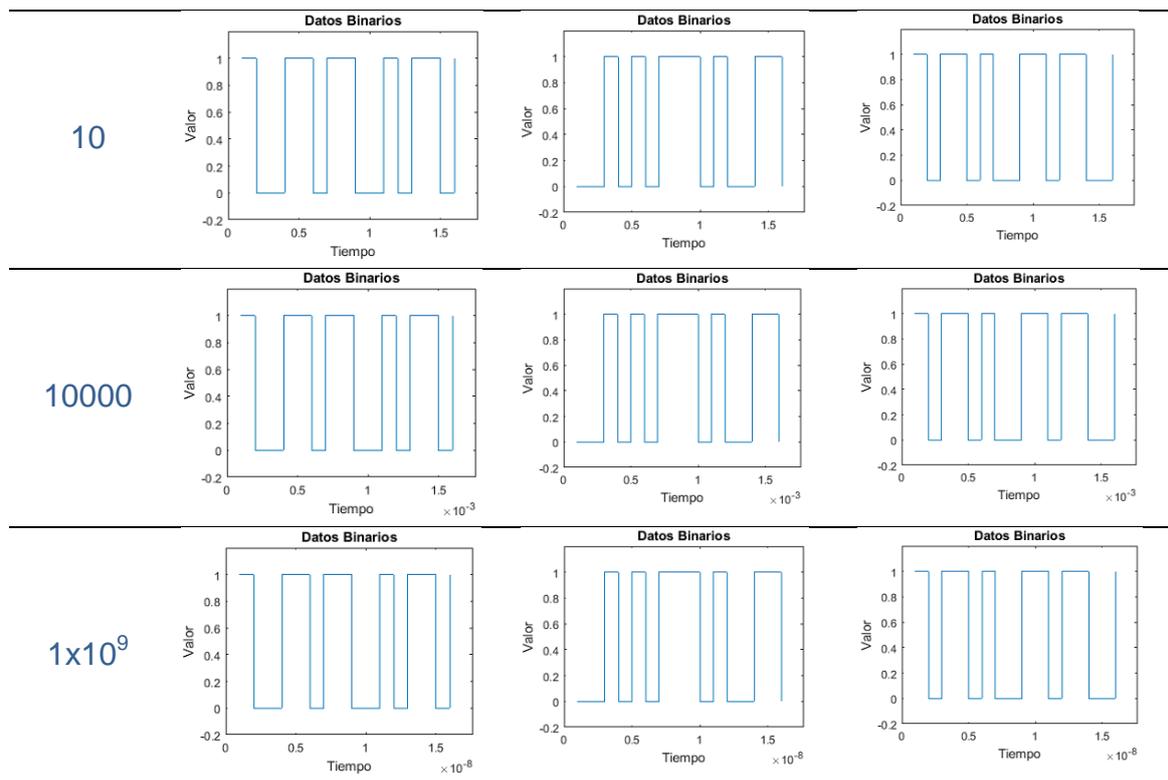
### 6.2. Generación de imágenes.

Una vez que se han ingresado cada uno de los arreglos de bits A, B y C, se procede a generar sus imágenes. Al pulsar el botón con el texto **Codificar** se genera automáticamente la imagen con una velocidad de transmisión de 10 Bits/s. Se modifica la velocidad de transmisión y procede de manera igual con los otros arreglos, resultando la siguiente tabla:

Tabla 12.

*Gráficas de ingreso de datos*

Vtx (Bits/s)	Arreglo A	Arreglo B	Arreglo C
-----------------	-----------	-----------	-----------



Se puede observar que el incremento de la velocidad de transmisión genera un cambio en el valor de la escala reduciéndola proporcionalmente en base a la velocidad. Las gráficas de las imágenes corresponden con los valores binarios como son introducidos.

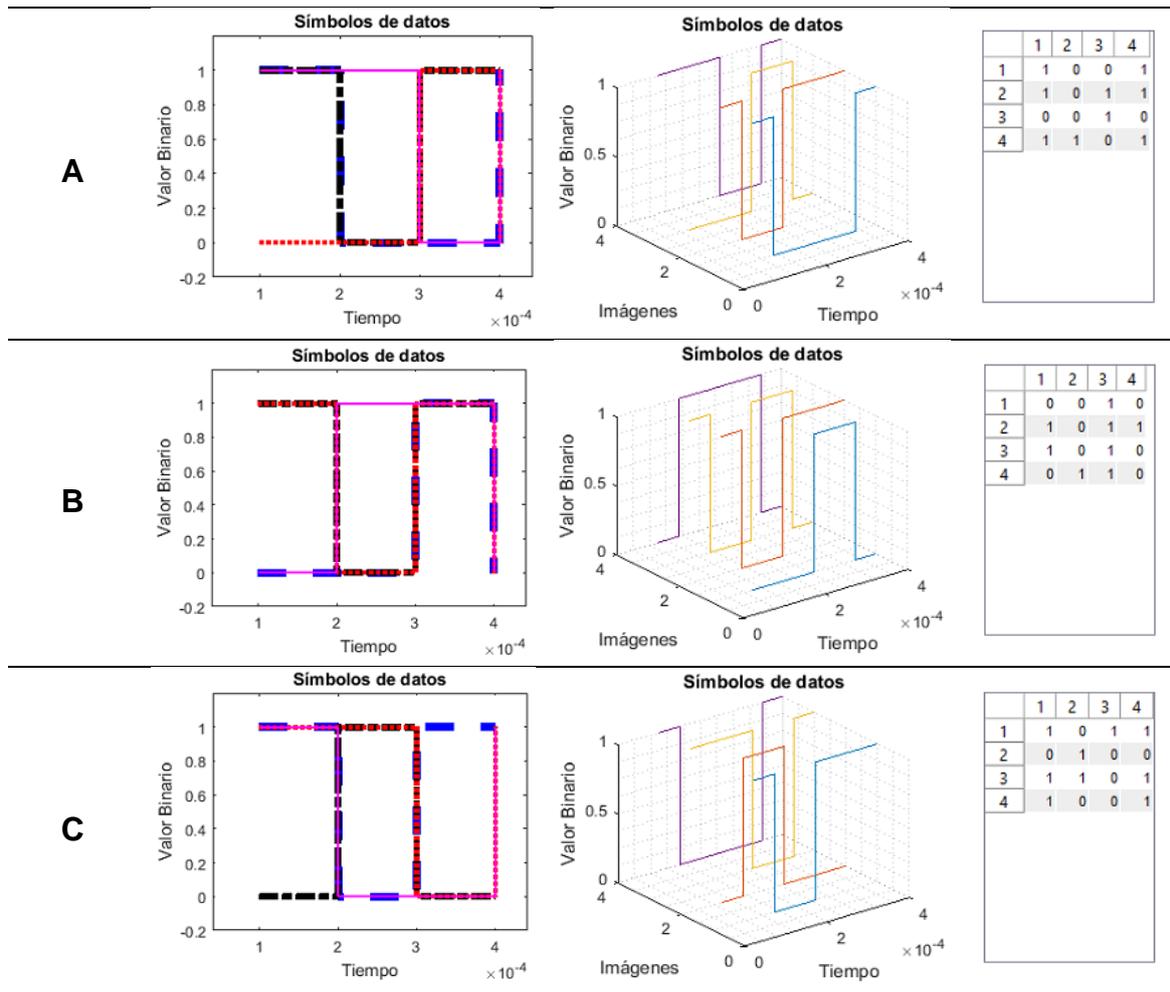
### 6.3. Conversión de serial a paralelo.

Las pruebas de este bloque serán desarrolladas con los tres arreglos indicados y con una velocidad de transmisión de 10.000 Bits/s. Los resultados son los siguientes:

Tabla 13.

*Resultados de la conversión S/P*

Arreglo	Gráfica 1	Gráfica 2	Tabla
---------	-----------	-----------	-------



La división de los datos en cuatro imágenes es causada por la modulación 16 QAM ya que requiere cuatro bits para aplicar su función. La gráfica tridimensional permite apreciar de un modo diferente la diferencia entre cada imagen y su concordancia con las tablas de datos.

#### 6.4. Modulación 16 QAM.

Cada conjunto de datos que entra en la modulación es recibido en un formato de cuatro bits, como se explicó anteriormente se realiza un mapeo en la constelación donde se muestra la referencia para visualizar el resultado, los valores numéricos resultantes se indican para verificar los puntos.

Tabla 14.

Constelaciones 16 QAM de las pruebas.

Arreglo	Constelación	Tabla										
A	<p>Mapa 16 QAM</p>	<table border="1"> <thead> <tr> <th></th> <th>1</th> </tr> </thead> <tbody> <tr> <td>1</td> <td><math>3.0000 + 1.0000i</math></td> </tr> <tr> <td>2</td> <td><math>3.0000 - 1.0000i</math></td> </tr> <tr> <td>3</td> <td><math>-3.0000 - 3.0000i</math></td> </tr> <tr> <td>4</td> <td><math>1.0000 + 1.0000i</math></td> </tr> </tbody> </table>		1	1	$3.0000 + 1.0000i$	2	$3.0000 - 1.0000i$	3	$-3.0000 - 3.0000i$	4	$1.0000 + 1.0000i$
		1										
1	$3.0000 + 1.0000i$											
2	$3.0000 - 1.0000i$											
3	$-3.0000 - 3.0000i$											
4	$1.0000 + 1.0000i$											
B	<p>Mapa 16 QAM</p>	<table border="1"> <thead> <tr> <th></th> <th>1</th> </tr> </thead> <tbody> <tr> <td>1</td> <td><math>-3.0000 - 3.0000i</math></td> </tr> <tr> <td>2</td> <td><math>3.0000 - 1.0000i</math></td> </tr> <tr> <td>3</td> <td><math>3.0000 - 3.0000i</math></td> </tr> <tr> <td>4</td> <td><math>-1.0000 - 3.0000i</math></td> </tr> </tbody> </table>		1	1	$-3.0000 - 3.0000i$	2	$3.0000 - 1.0000i$	3	$3.0000 - 3.0000i$	4	$-1.0000 - 3.0000i$
		1										
1	$-3.0000 - 3.0000i$											
2	$3.0000 - 1.0000i$											
3	$3.0000 - 3.0000i$											
4	$-1.0000 - 3.0000i$											
C	<p>Mapa 16 QAM</p>	<table border="1"> <thead> <tr> <th></th> <th>1</th> </tr> </thead> <tbody> <tr> <td>1</td> <td><math>3.0000 - 1.0000i</math></td> </tr> <tr> <td>2</td> <td><math>-1.0000 + 3.0000i</math></td> </tr> <tr> <td>3</td> <td><math>1.0000 + 1.0000i</math></td> </tr> <tr> <td>4</td> <td><math>3.0000 + 1.0000i</math></td> </tr> </tbody> </table>		1	1	$3.0000 - 1.0000i$	2	$-1.0000 + 3.0000i$	3	$1.0000 + 1.0000i$	4	$3.0000 + 1.0000i$
		1										
1	$3.0000 - 1.0000i$											
2	$-1.0000 + 3.0000i$											
3	$1.0000 + 1.0000i$											
4	$3.0000 + 1.0000i$											

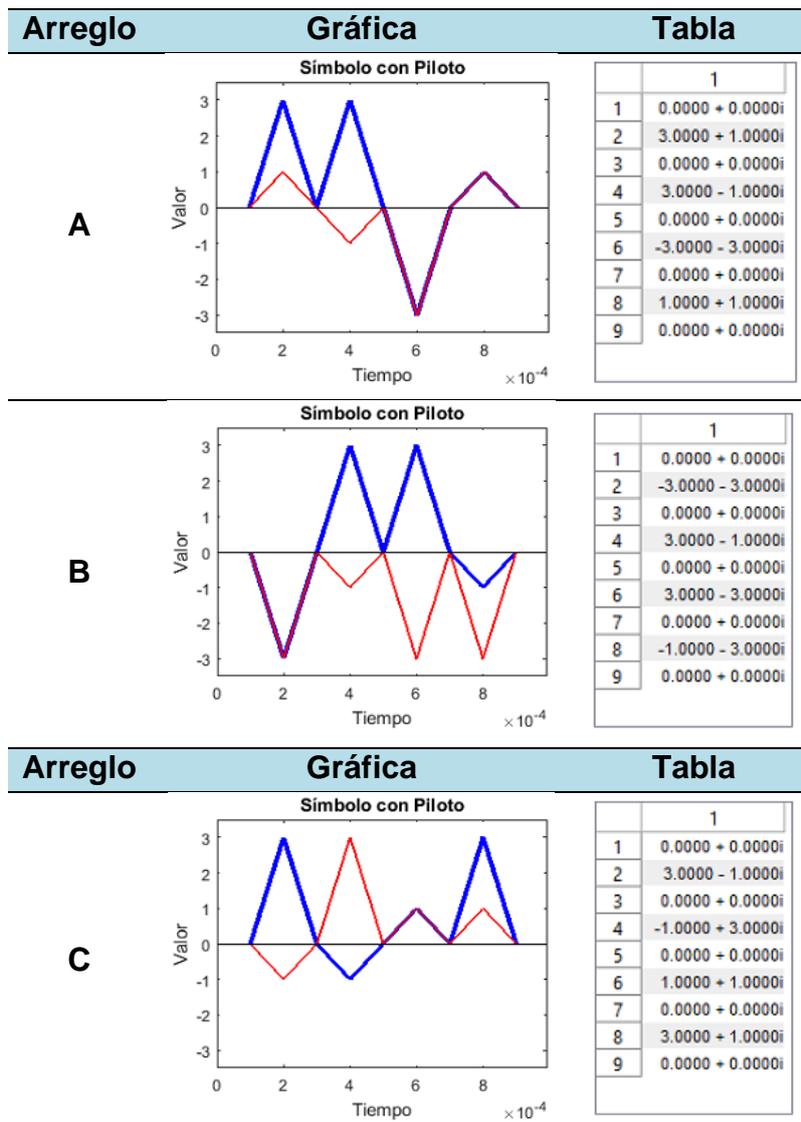
Se puede observar que los resultados obtenidos corresponden con los valores de la tabla presentada en el capítulo 2. La variación de los valores permite obtener diferentes gráficas lo que convierte el simulador en una herramienta de apoyo académico.

### 6.5. Inserción de pilotos.

Para insertar los pilotos se ingresan los vectores de cuatro valores resultantes de las modulaciones de 16 QAM de cada arreglo de bits. Continuando con el uso del simulador se obtienen los siguientes resultados:

Tabla 15.

Gráficas de inserción de pilotos.



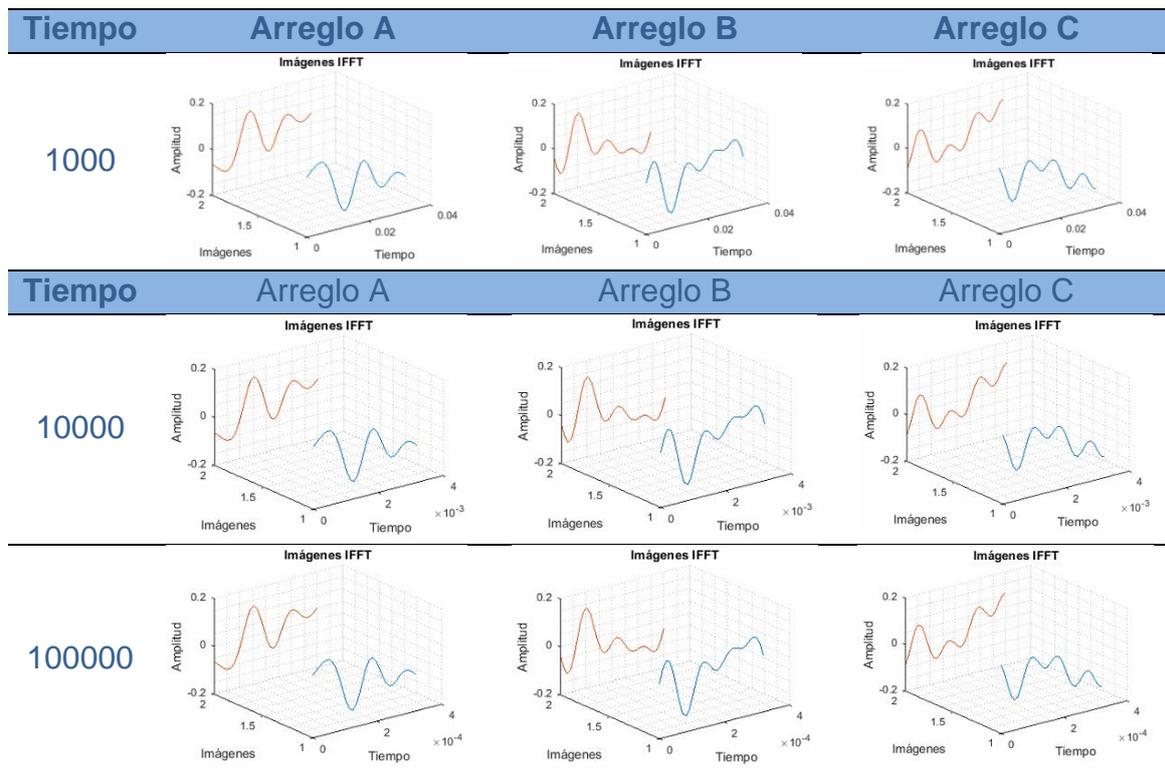
Se puede verificar que los vectores resultantes de la inserción de los pilotos poseen una longitud de nueve (09) valores. Las gráficas presentadas demuestran los puntos de datos nulos del proceso y la variación de cada uno de ellos en correspondencia con los valores de la tabla.

## 6.6. Aplicación de IFFT.

Esta modulación posee un conjunto de variables que pueden ser modificadas. Para afianzar el estudio se establecerá una variación del tiempo de bits de la modulación y se realizarán salidas de dos portadoras con longitud de 64 para apreciar el cambio del tiempo en sus gráficas.

Tabla 16.

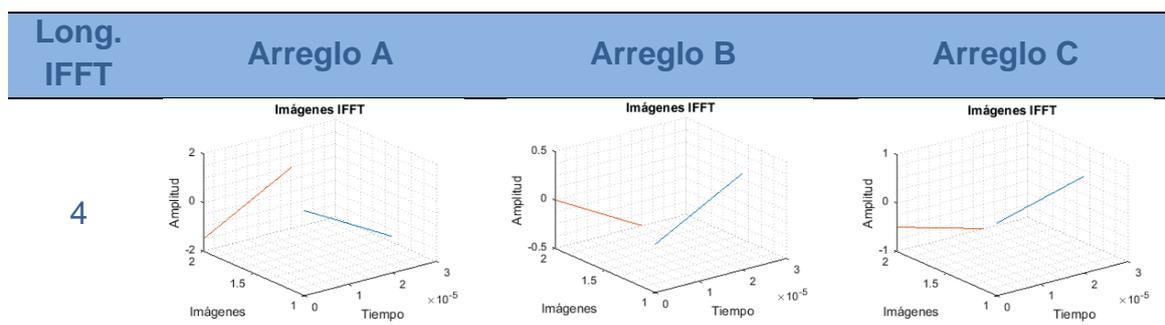
Tabla de aplicación de IFFT con variación de tiempo.

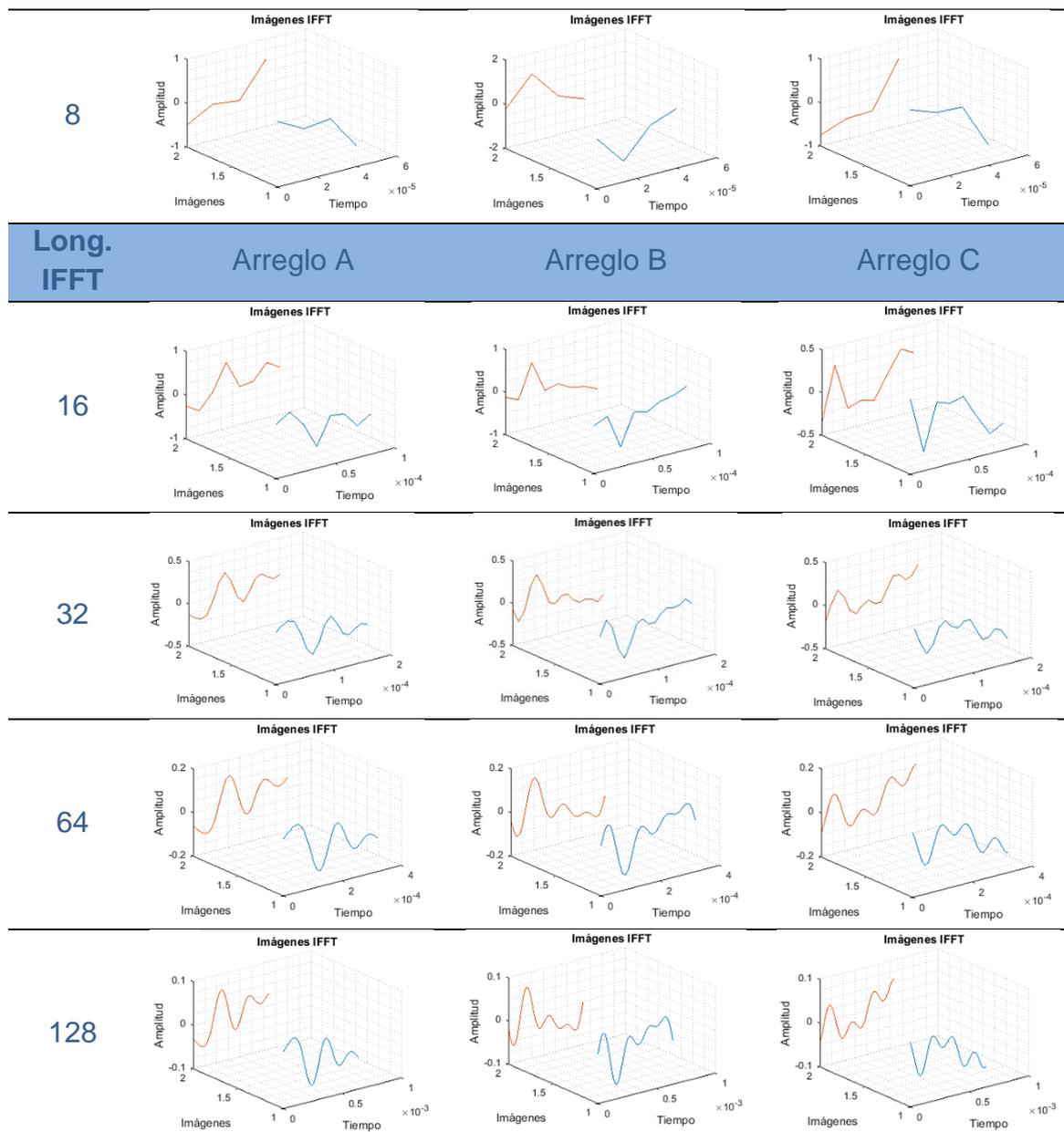


Esta prueba demuestra que el eje del tiempo varía para ajustarse al valor unitario de las divisiones de los ejes, por lo tanto, a mayor velocidad menor será el valor de cada referencia del eje. La longitud de la IFFT será variada para cada arreglo conservando constante el tiempo de bits en 100.000 Bits/s y con dos portadoras, los resultados son los siguientes:

Tabla 17.

Gráficas de aplicación IFFT con variación de la longitud





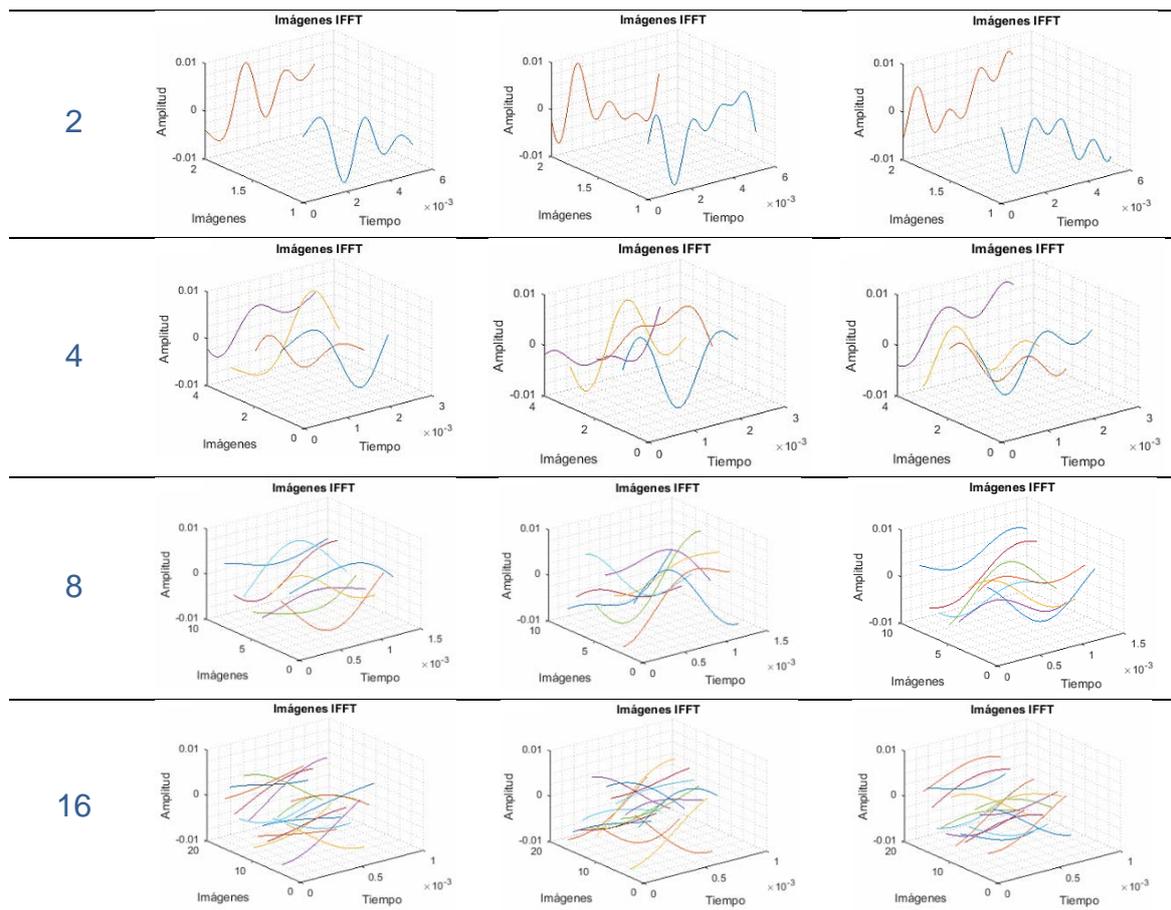
La aplicación de la IFFT demuestra utilizando dos portadoras permite demostrar que la longitud de la modulación es dividida en función a la cantidad de portadoras, esto produce que la gráfica sea cada vez más suavizada cuando se incrementa dicha longitud. La realización de la modulación denota la diferencia entre cada arreglo por la diferencia de los datos.

Estableciendo ahora la variación de las portadoras con el tiempo de bits de 100.000 Bits/s y una longitud de 1024, las gráficas resultan:

Tabla 18.

*Aplicación de IFFT con variación de las portadoras.*

Porta.	Arreglo A	Arreglo B	Arreglo C
--------	-----------	-----------	-----------

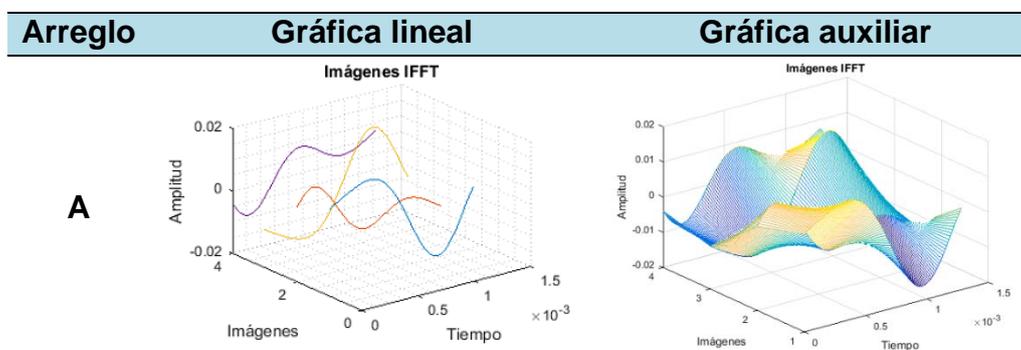


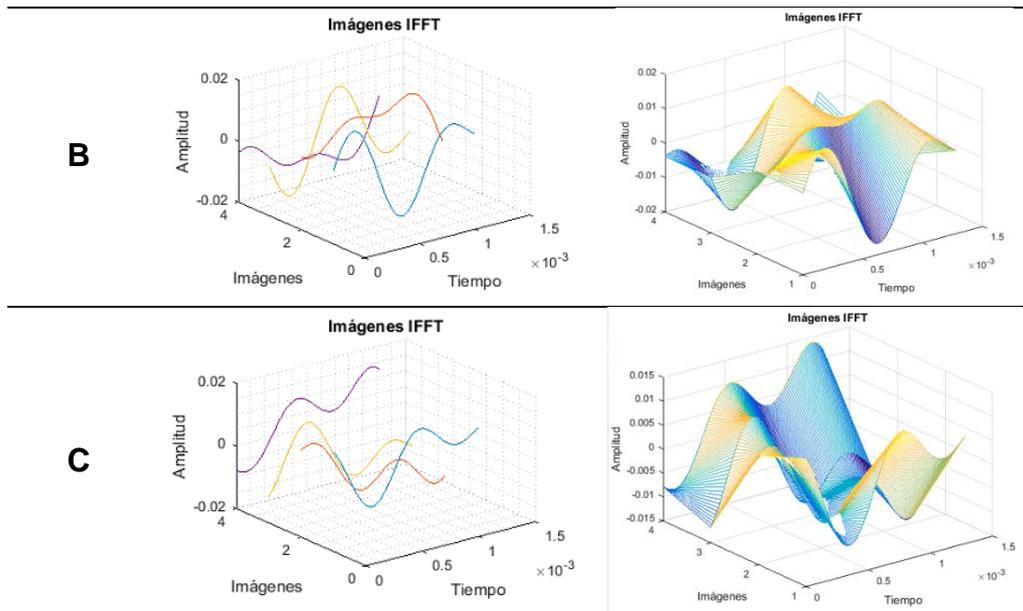
De éstos resultados se puede visualizar que las líneas de las gráficas disminuyen sus curvaturas a medida que la cantidad de portadoras incrementan.

Para continuar el estudio se mantendrá la selección de longitud de IFFT de 512, 100.000 Bits/s y cuatro portadoras, las gráficas incluyendo las auxiliares son las siguientes:

Tabla 19.

*Gráficas IFFT seleccionadas.*





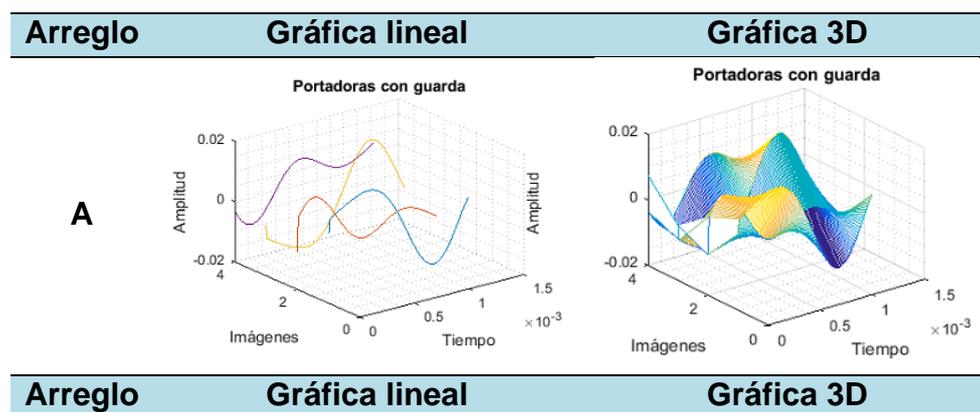
Estas configuraciones de salidas son seleccionadas por su presentación gráfica y para conservar una cantidad de cuatro portadoras para los siguientes bloques.

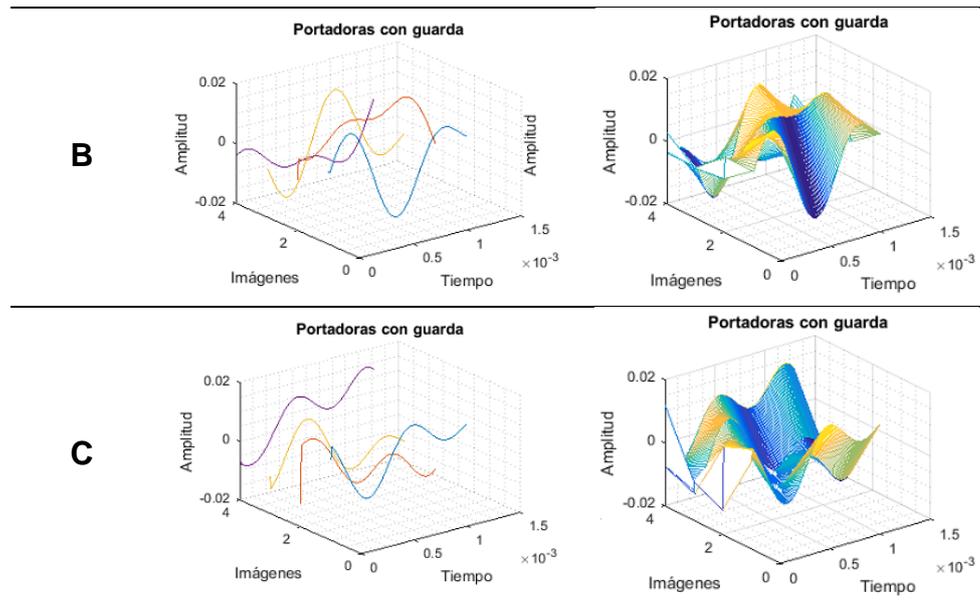
### 6.7. Aplicación de guarda.

La aplicación de la guarda es una duplicación del último elemento de cada portadora. Este proceso para los tres arreglos seleccionados presenta los siguientes resultados gráficos:

Tabla 20.

*Gráficas de la guarda.*





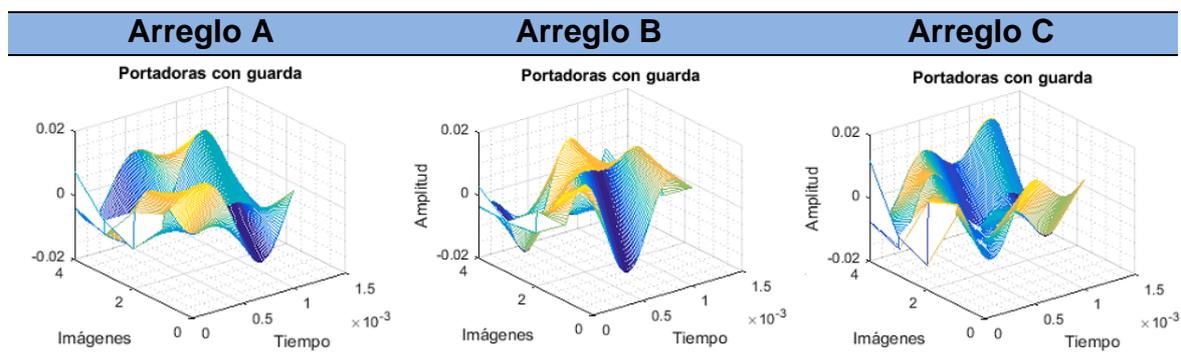
Las gráficas líneas obtenidas de éste proceso permiten visualizar que se incorpora el prefijo a cada portadora en comparación a las gráficas que entran al bloque. Las gráficas tridimensionales son las representaciones en una capa de unión de las portadoras.

### 6.8. Conversión paralelo a serial.

Este bloque realiza la unión de las portadoras, por lo tanto, su representación gráfica se conserva en un patrón tridimensional con las variaciones resultantes como representación de la concatenación de valores.

Tabla 21,

*Gráficas de conversión P/S.*



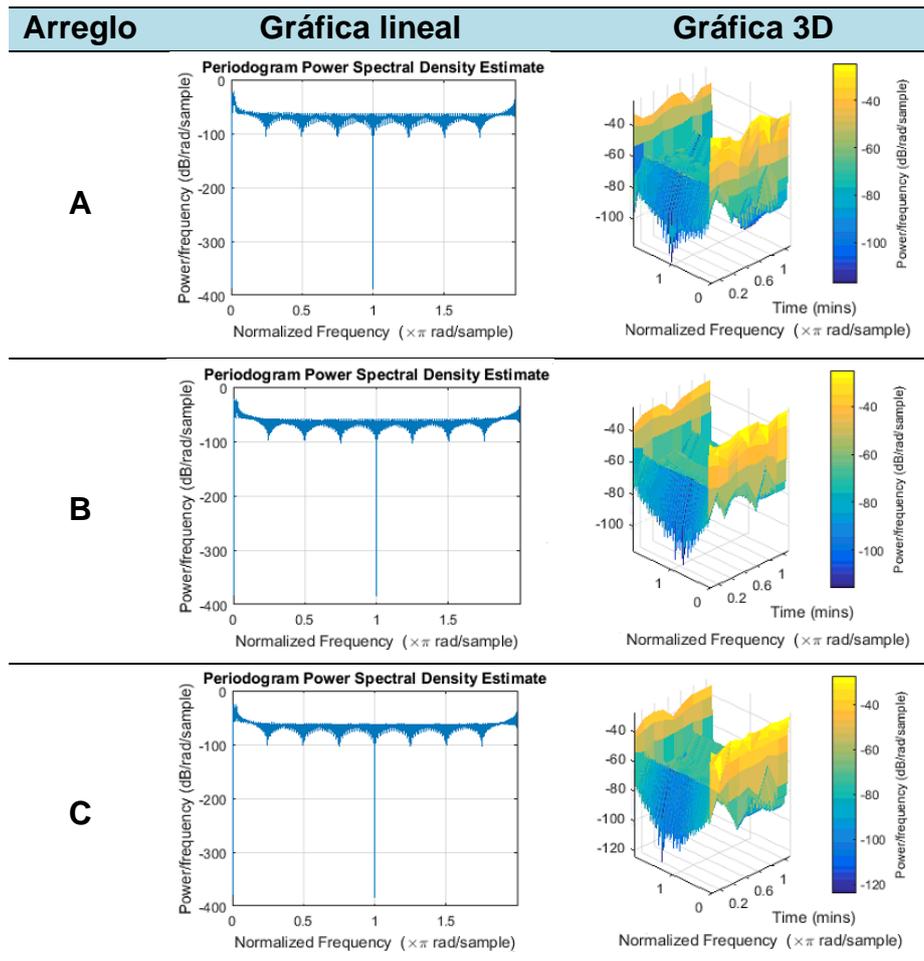
Las gráficas de éste proceso son idénticas al anterior debido a que se expresa la idea académica de la unión de las portadoras, aunque en el simulador se indiquen las tablas de resultados en la salida del bloque los vectores resultantes de la concatenación de las portadoras.

## 6.9. Transmisión.

En este proceso se realiza un análisis del vector obtenido de la unión de las portadoras, en el MATLAB se realizan gráficas especiales para esto y se incluye una representación tridimensional. Los resultados para los tres datos de estudios son:

Tabla 22.

*Gráficas de la transmisión.*



Las gráficas obtenidas en éste bloque demuestran una variación detectable en la representación del periodograma y de la visualización tridimensional. Estas representaciones son importantes para dar a entender la idea y el concepto de la transmisión.

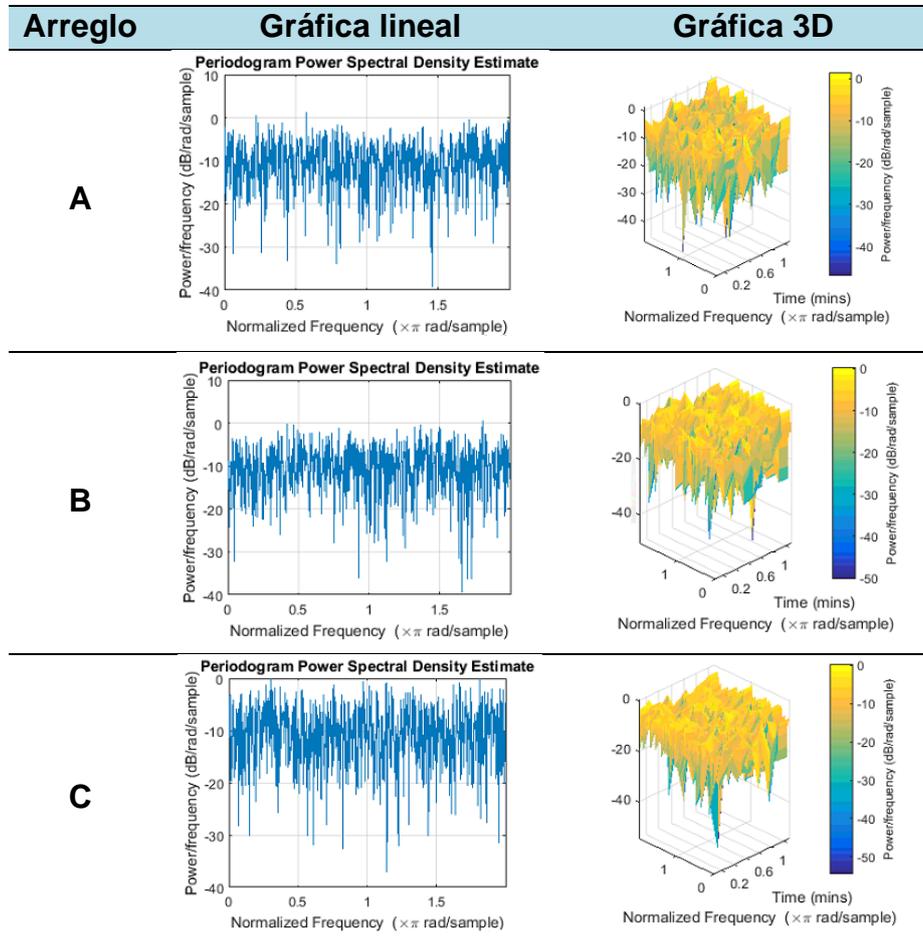
## 6.10. Canal.

A pesar de que éste bloque no pertenece a la transmisión, se incluye la representación gráfica en el mismo orden de la transmisión para realizar una

comparación de sus resultados y demostrar los cambios. Para los casos de prueba se tienen los siguientes resultados:

Tabla 23.

*Gráficas del paso por el canal.*



El simulador desarrollado en MATLAB ha permitido establecer diversas variables para diseñar la OFDM de un vector binario incluyendo la generación de diversas gráficas para visualizar el estado de la modulación.

Entre los resultados es interesante contemplar la variación de la constelación obtenida por los diversos vectores que ingresan a la modulación y cómo los cambios de longitud, tiempo y portadoras aplicadas durante la IFFT pueden permitir la manipulación de una modulación con múltiples combinaciones.

Las gráficas tridimensionales permiten visualizar las ubicaciones de las portadoras en un entorno de tiempo diferente e incluso se puede observar que las portadoras generan una sola transmisión por la unión de ellas. Las gráficas de estimación de densidad de la señal que permite generar el MATLAB

complementa el estudio desarrollado a pesar de tener una limitación de 15 bits de entrada de datos, donde se puede observar la diferencia entre las gráficas.

## CONCLUSIONES Y RECOMENDACIONES

### 7.1 Conclusiones

El desarrollo de un simulador OFDM combinado con una modulación 16 QAM, facilita el entendimiento académico del proceso de realización de la modificación de datos, estableciéndose como una herramienta práctica de fácil manejo. A nivel de implementación en el laboratorio, generalmente dichas prácticas guiadas permiten a los estudiantes a percibir de una forma tangible la transmisión por aplicación de las modulaciones como refuerzo a las teorías recibidas en clases

Se determinó la cantidad de variables requeridas en el proceso de aplicación de OFDM durante la realización del simulador para representar una interfaz completa con el usuario.

Mediante un simulador OFDM los estudiantes pueden asimilar de mejor forma los conceptos de esta modulación, adicionalmente pueden interactuar en el proceso y obtener diversos resultados. Cuando un estudiante se desenvuelve como un investigador, éste se esfuerza en resolver y comprender los pasos asociados con su meta de estudio, por lo tanto, los lectores y usuarios del simulador podrán prestar un mayor interés a las nuevas tendencias de las modulaciones digitales y se podrá incentivar su estudio y análisis

El desarrollo del diagrama de flujo permite que el uso de la aplicación sea visualizado como un proceso compuesto por funciones internas lo que permite al usuario tener una mejor comprensión del proceso de modulación OFDM.

Las pruebas y utilización del simulador permiten establecer una confianza de los resultados estableciéndose una base académica para futuros usos y desarrollos de otros proyectos.

Las aplicaciones de las funciones de MATLAB permiten gestionar bloque por bloque la realización de todo el proceso de modulación y realizar las representaciones gráficas para visualizar los resultados, esto es importante para conocer OFDM.

El desarrollo de un simulador de OFDM facilita el aprendizaje del proceso de la modulación de datos, estableciéndose como una herramienta práctica de fácil manejo.

En referencia al uso en el laboratorio, las prácticas guiadas permiten a los estudiantes, percibir de una forma visual la transmisión por aplicación de las modulaciones como refuerzo a las teorías recibidas en clases.

En las pruebas que se realizaron mediante el simulador OFDM desarrollado en MATLAB ha permitido establecer entre sus resultados diversos resultados, la generación de diversas gráficas para visualizar el estado de la modulación es una de ellas, la variación de las constelaciones obtenida por los diversos vectores que ingresan a la modulación y cómo los cambios de longitud, tiempo y portadoras aplicadas durante la IFFT pueden permitir la manipulación de una modulación con múltiples combinaciones, como una ayuda pedagógica, las gráficas tridimensionales permiten visualizar las portadoras en un entorno mucho más didáctico..

## **7.2. Recomendaciones**

Incentivar a los estudiantes de Ingeniería de Telecomunicaciones al desarrollo de propuestas de aplicaciones de nuevas tecnologías de intercambios de informaciones por modulaciones y al uso de programaciones de cálculos avanzados como el MATLAB.

Estimular el uso del simulador en las prácticas de laboratorio y en clases académicas para reforzar los conocimientos adquiridos.

Impulsar la competencia educativa para gestionar cambios al simulador y desarrollos de prácticas de laboratorios con el mismo.

Proponer la creación de simuladores para la aplicación en prácticas de laboratorio con las diversas modulaciones OFDM.

Velar por afianzar la importancia académica del hábito en propuestas y proyectos durante su formación académica para incrementar su capacidad investigativa de la carrera.

## REFERENCIAS

- Acosta, G. (2000). *OFDM Simulation using MATLAB, Trabajo de laboratorio*. Recuperado el 04 de noviembre de 2017, de [http://sarl.ece.gatech.edu/wp-content/uploads/sites/488/2016/09/Tutorial\\_web.pdf](http://sarl.ece.gatech.edu/wp-content/uploads/sites/488/2016/09/Tutorial_web.pdf)
- Addlink Software Científico. (s.f.). Recuperado el 31 de octubre de 2017, de <https://www.addlink.es/productos/software/maple>
- Ashqui, S. (2016). *Implementación de un Sistema OFDM de Transmisión Y Recepción Inalámbrica de Imagen a través de modulación QPSK y8PSK*. Recuperado el 22 de octubre de 2017, de <http://www.dsplace.espol.edu.ec/xmlui/bitstream/handle/123456789/37219/D-103526.pdf?sequence=-1&isAllowed=y>
- Briceño, J. (2012). *Principios de las Comunicaciones (3era ed.)*. Recuperado el 25 de octubre de 2017, de [http://www.serbi.ula.ve/serbiula/libros-electronicos/Libros/principios/pdf/libro\\_completo.pdf](http://www.serbi.ula.ve/serbiula/libros-electronicos/Libros/principios/pdf/libro_completo.pdf)
- Cacheme. (2012). Tomado el 4 de noviembre de 2017, de <http://cacheme.org/alternativas-a-matlab-scilab/#!prettyPhoto>
- Cho, Y. K. (2010). *MIMO-OFDM Wirelees Communications with MATLAB*. Recuperado el 20 de octubre de 2017, de <http://www.ee.iitm.ac.in/~giri/pdfs/EE6002/book-cho.pdf>
- Couch, L. (2007). *Sistemas de comunicaciones digitales y analógicos (7ma Ed.)*. Recuperado el 31 de octubre de 2017, de <https://hellsingge.files.wordpress.com/2015/02/sistemas-de-comunicacion-digital-y-analogico-leon-w-couch-7ma-edicion.pdf>
- Descripción de MATLAB. (s.f.). Recuperado el 04 de noviembre de 2017, de <http://nereida.deioc.ull.es/~pcgull/ihiu01/cdrom/matlab/contenido/node2.html>
- Estepa, R. (2004). *Evolucion Historica de las Telecomunicaciones*. Recuperado el 21 de octubre de 2017, de <http://trajano.us.es/~rafa/ARSS/apuntes/tema1.pdf>
- Footplot. (s.f.). Recuperado el 20 de octubre de 2017, de [www.footplot.com](http://www.footplot.com)
- GNU OCTAVE. (s.f.). Recuperado el 3 de noviembre de 2017, de <https://www.gnu.org/software/octave/>
- Hemple, C. (2012). *Analysis and Simulation of Wireless OFDM communications*. Recuperado el 02 de noviembre de 2017, de

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.857.7409&rep=rep1&type=pdf>

Jagadeesh, D. (2005-2007). *Simulation of Wireless Communication System using OFDM Principle*. Recuperado el 01 de noviembre de 2017, de <http://ethesis.nitrkl.ac.in/4380/1/a.pdf>

Kaise, F. K. (2008). *Multi-Carrier and Spread Spectrum Systems*. Recuperado el 23 de octubre de 2017, de <http://www.ee.iitm.ac.in/~giri/pdfs/EE6002/book-kaiser.pdf>

Lopez, F. (2005). *Performance analysis of MIMO-OFDM systems using complex Gaussian quadratic forms*. Recuperado el 01 de noviembre de 2017, de <http://www.ic.uma.es/repository/fileDownloader?rfname=818bfc20-6226-4bb2-98fc-bea8ada45ca1.pdf>

Mathworks. (s.f.). Recuperado el 02 de noviembre de 2017, de <https://www.mathworks.com/help/physmod/sps/powersys/ref/powergui.html>

MathWorks. (s.f.). Recuperado de Constelation Visualization: <https://www.mathworks.com/help/comm/ug/constellation-visualization.html>

Navarro, J. (2015). *Simulación de señales distribuidas en RoF para Redes WiMAX a través de la Modulación OFDM y 64-QAM*. Recuperado el 28 de octubre de 2017, de <http://repositorio.ucsg.edu.ec/bitstream/3317/4329/1/T-UCSG-POS-MTEL-48.pdf>

Nguyen, H. (2015). *Introduction to OFDM*. Recuperado el 28 de octubre de 2017, de Universidad de Saskatchewan: <http://www.engr.usask.ca/classes/EE/456/protected/Introduction-to-OFDM-F15.pdf>

*Orthogonal Frequency Division Multiplexing System with Frequency Offset Estimation*. (s.f.). Recuperado el 31 de octubre de 2017, de International Journal of Electronics and Computer Science Engineering.: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.300.624&rep=rep1&type=pdf>

Pei-Yun, C. T.-D. (2007). *OFDM Baseband Receiver Design for Wireless Communication*. Recuperado el 10 de noviembre de 2017, de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.468.6434&rep=rep1&type=pdf>.

- Scilab Enterprise*. (2017). Tomado el 23 de octubre de 2017, de <https://www.scilab.org/en/scilab/about>
- SciStatCalc*. (2013). Tomado el 16 de noviembre de 2017, de FFT calculator: <http://scistatcalc.blogspot.com/2013/12/fft-calculator.html>
- Simulación de Sistemas de Control Continuo con MATLAB y SIMULINK*. (2006). Recuperado el 04 de noviembre de 2017, de <http://isa.uniovi.es/docencia/dscc/matlabysimulink.pdf>
- Tannhausser. (2015). *La Mirada del Replicante*. Recuperado el 25 de octubre de 2017, de <https://lamiradadelreplicante.com/2015/05/31/gnu-octave-4-0-llega-estrenando-interfaz-grafica/>
- Tibor, W. (2014). *Introduction to the use of MATLAB*. Recuperado el 31 de octubre de 2017, de [http://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0041\\_introduction\\_matlab/2011-0041\\_introduction\\_matlab.pdf](http://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0041_introduction_matlab/2011-0041_introduction_matlab.pdf)
- Tomasi, W. (2003). *Sistemas de Comunicaciones Electrónicas*. Recuperado el 3 de noviembre de 2017, de <https://hellsingge.files.wordpress.com/2014/08/sistemas-de-comunicaciones-electronicas-tomasi-4ta-edicic3b3n.pdf>
- Wei Lu, G. (2015). *Optical Signal Processing for High-Order Quadrature-Amplitude Modulation Formats*. Recuperado el 01 de noviembre de 2017, de INTECH: <https://www.intechopen.com/books/applications-of-digital-signal-processing-through-practical-approach/optical-signal-processing-for-high-order-quadrature-amplitude-modulation-formats> doi: 10.5772/61681

**ANEXOS**



B																		
C																		

2. Proceda a introducir la **Señal A** en la casilla de *Generación de Código*.

**Nota:** conserve los corchetes existentes en el formato de la casilla para evitar errores del código.

3. Presione el botón **Codificar** y espere hasta que aparezca la gráfica. Dibuje la gráfica de la imagen en el espacio correspondiente:

Señal A	Señal B	Señal C

4. Modifique la velocidad de transmisión ( $V_{tx}$ ) por medio de la barra deslizante al lado del botón *Imagen*, al valor 1.000 y seguidamente pulse el botón indicado. Repita éste paso para los siguientes valores: 100.000 y 100.000.000. Anote su observación:

---



---



---



---



---



---



---

5. Presione el botón *Ser-Paralelo* y dibuje la gráfica bidimensional conservando la diferenciación de las líneas.

Señal A	Señal B	Señal C

6. Presione el botón *16 QAM* y complete la constelación obtenida en el espacio siguiente:

Señal A				Señal B				Señal C			
+	+	+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+	+	+

7. Anote los valores numéricos resultantes de la modulación 16 QAM en las casillas siguientes:

Número	Señal A	Señal B	Señal C
1			
2			
3			
4			

8. Presione el botón *Piloto* y grafique el resultado.

Señal A	Señal B	Señal C

9. De la sección del IFFT, conserve la longitud de la modulación en el valor de cuatro (04), establezca un tiempo de bit de 100.000 Bits/s, y la cantidad de dos (02) portadoras. Presione el botón *Aplicar*.

9.1. Inicie un incremento gradual de la longitud del IFFT y anote su observación:

---



---



---



---

9.2. Incremente el valor del tiempo de bit agregándole ceros "0" al valor inicial. Anote su observación:

---



---

---

---

---

9.3. Seleccione diferentes valores de la cantidad de portadoras y anote su observación:

---

---

---

---

9.4. Establezca la longitud de 512, tiempo de bit de 100.000 Bits/s, ocho (08) portadoras y presione el botón Aux del bloque IFFT para visualizar una representación tridimensional. Mantenga las variables.

10. Presione el botón *Guarda* y anote su observación:

---

---

---

---

---

---

11. Presione el botón *Transmisión* y luego el botón *Canal*. Anote su observación.

---

---

---

---

---

---

12. Realice de nuevo los procedimientos con las otras dos señales seleccionadas, complete las gráficas respectivamente y verifique sus observaciones.

## ANEXO 2 Código

Sección 1.- Código de Carga de la ventana

```
function Control_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
handles.output = hObject;
guidata(hObject, handles);
axes(handles.axes6);
```

```
MATLABImage=imread('Diagrama.png');  
image(MATLABImage)  
axis off  
axis image
```

```
function varargout = Control_OutputFcn(hObject, eventdata, handles)  
varargout{1} = handles.output;
```

## Sección 2.- Código del **botón 1**

```
function pushbutton1_Callback(hObject, eventdata, handles)  
%Generación de los bits de entrada
```

```
%Manejo de los botones
```

```
set(handles.pushbutton2,'Enable','on');  
set(handles.pushbutton3,'Enable','on');  
set(handles.slider3,'Enable','on');  
set(handles.pushbutton4,'Enable','off');  
set(handles.pushbutton5,'Enable','off');  
set(handles.pushbutton6,'Enable','off');  
set(handles.pushbutton8,'Enable','off');  
set(handles.pushbutton9,'Enable','off');  
set(handles.pushbutton10,'Enable','off');  
set(handles.pushbutton11,'Enable','off');  
set(handles.pushbutton12,'Enable','off');
```

```
%Entrada de datos
```

```
Valor=handles.edit1.String;
```

```
%Salida de los datos
```

```
handles.Valor=Valor;  
guidata(hObject,handles);
```

```
pushbutton2_Callback(handles.pushbutton2, eventdata, handles);
```

## Sección 3.- codificación **Slider3**

```
function slider3_Callback(hObject, eventdata, handles)
```

```
%Esta codificación permite controlar el tiempo de bit
```

```
Expo2=10^round(get(handles.slider3,'Value'));  
set(handles.text7,'String',Expo2);
```

## Sección 4.- codificación **boton2**

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

```
% Este botón sólo manipula la salida binaria
%Por tal motivo no hay entrada de datos
```

```
%Control de tablas y gráficos
```

```
cla(handles.axes1);
set(handles.axes1,'Visible','off');
cla(handles.axes4,'reset');
set(handles.axes4,'visible','off');
cla(handles.axes5,'reset');
set(handles.axes5,'visible','off');
set(handles.uitable1,'Visible','off','Data',cell(size(get(handles.uitable1,'Data'))));
set(handles.axes2,'Visible','on');
set(handles.uitable2,'Visible','on');
```

```
set(handles.pushbutton13,'BackgroundColor',[.59 1 1]);
set(handles.pushbutton14,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton15,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton16,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton17,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton18,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton19,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton20,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton21,'BackgroundColor',[.94 .94 .94]);
```

```
%Entrada de datos
```

```
Valor=handles.Valor;
```

```
%Captura de los datos
```

```
Valor1 = reshape(str2num(Valor),[],1);
Tempo=str2num(get(handles.text7,'String'));
Escala=[1:16]/Tempo;
```

```
%generador de gráficas
```

```
stairs(handles.axes2,Escala,Valor1);
title(handles.axes2,'Datos Binarios')
xlabel(handles.axes2,'Tiempo');
ylabel(handles.axes2,'Valor');
xlim(handles.axes2,[0,Escala(16)+Escala(16)/10]);
ylim(handles.axes2,[-0.2,1.2]);
set(handles.uitable2,'Data',Valor1);
set(handles.uitable2,'ColumnWidth',{30});
```

```
%Salida de los datos
```

```
handles.Escala=Escala;
handles.Valor1=Valor1;
guidata(hObject,handles);
```

## Sección 5.- 4.4. Conversión de serial a paralelo

```
function pushbutton3_Callback(hObject, eventdata, handles)
```

```
% Procedimiento para controlar la modulación 16 QAM
```

```
%Control de los botones, gráficos y tablas
```

```
set(handles.pushbutton4,'Enable','on');
```

```
cla(handles.axes4,'reset');
```

```
set(handles.axes4,'visible','off');
```

```
cla(handles.axes5,'reset');
```

```
set(handles.axes5,'visible','on');
```

```
set(handles.axes1,'Visible','on');
```

```
set(handles.uitable1,'Visible','on');
```

```
set(handles.pushbutton13,'BackgroundColor',[.94 .94 .94]);
```

```
set(handles.pushbutton14,'BackgroundColor',[.59 1 1]);
```

```
set(handles.pushbutton15,'BackgroundColor',[.94 .94 .94]);
```

```
set(handles.pushbutton16,'BackgroundColor',[.94 .94 .94]);
```

```
set(handles.pushbutton17,'BackgroundColor',[.94 .94 .94]);
```

```
set(handles.pushbutton18,'BackgroundColor',[.94 .94 .94]);
```

```
set(handles.pushbutton19,'BackgroundColor',[.94 .94 .94]);
```

```
set(handles.pushbutton20,'BackgroundColor',[.94 .94 .94]);
```

```
set(handles.pushbutton21,'BackgroundColor',[.94 .94 .94]);
```

```
%Entrada de datos
```

```
Escala=handles.Escala;
```

```
Valor1=handles.Valor1;
```

```
Separ=transpose(reshape(Valor1,[],4));
```

```
%Generador de gráficos en la entrada
```

```
stairs(handles.axes1,Escala,Valor1);
```

```
title(handles.axes1,'Datos Binarios')
```

```
xlabel(handles.axes1,'Tiempo');
```

```
ylabel(handles.axes1,'Valor');
```

```
xlim(handles.axes1,[0,Escala(16)+Escala(16)/10]);
```

```
ylim(handles.axes1,[-0.2,1.2]);
```

```
set(handles.uitable1,'Data',Valor1);
```

```
set(handles.uitable1,'ColumnWidth',{30});
```

```
%datos de salida
```

```
Escala2=Escala(1:4);
```

```
set(handles.uitable2,'visible','on','Data',Separ);
```

```
set(handles.uitable2,'ColumnWidth',{30,30,30,30});
```

```
Garfi=stairs(handles.axes2,Escala2,transpose(Separ));
```

```
xlim(handles.axes2,[Escala2(4)/10,Escala2(4)+Escala2(4)/10]);
```

```
ylim(handles.axes2,[-0.2,1.2]);
```

```
legend(handles.axes2,'imagen1','imagen2','imagen3','imagen4');
```

```

set(Garfi(1),'LineWidth',4.5,'LineStyle','--','Color','b');
set(Garfi(2),'LineWidth',3.5,'LineStyle','-','Color','k');
set(Garfi(3),'LineWidth',2.5,'LineStyle',':','Color','r');
set(Garfi(4),'LineWidth',1.5,'LineStyle','-','Color','m');
title(handles.axes2,'Símbolos de datos')
xlabel(handles.axes2,'Tiempo');
ylabel(handles.axes2,'Valor Binario');

```

### %Preparación remota de los datos para la gráfica 3D

```

Datos3=kron(Separ,ones(3,1));
Datos3([1 12],:)=[];
vec1=Escala2(1)*[1 1.5 1.5 2 2.5 2.5 3 3.5 3.5 4]';
Ex=kron(vec1,ones(1,4));
vec2=[1 2 3 4];
Ey=kron(vec2,ones(10,1));
plot3(handles.axes5,Ex,Ey,Datos3);
grid(handles.axes5,'minor')
title(handles.axes5,'Símbolos de datos')
xlabel(handles.axes5,'Tiempo');
ylabel(handles.axes5,'Imágenes');
zlabel(handles.axes5,'Valor Binario');

```

### %envío de datos

```

handles.Escala2=Escala2;
handles.Separ=Separ;
handles.Ex=Ex;
handles.Ey=Ey;
handles.Datos3=Datos3;
guidata(hObject,handles);

```

Sección 6.- Modulación 16 QAM.

```
function pushbutton4_Callback(hObject, ~, handles)
```

### %Preparación de símbolos

### %Control de tablas, botones y datos

```

set(handles.pushbutton5,'Enable','on');
cla(handles.axes4,'reset');
set(handles.axes4,'visible','on');
cla(handles.axes5,'reset');
set(handles.axes5,'visible','off');

```

```

set(handles.pushbutton13,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton14,'BackgroundColor',[.94 .94 .94]);

```

```

set(handles.pushbutton15,'BackgroundColor',[.59 1 1]);
set(handles.pushbutton16,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton17,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton18,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton19,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton20,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton21,'BackgroundColor',[.94 .94 .94]);

```

### %Entrada de datos

```
Separ=handles.Separ;
```

```
Escala2=handles.Escala2;
```

```
Ex=handles.Ex;
```

```
Ey=handles.Ey;
```

```
Datos3=handles.Datos3;
```

### %Presentación de los datos de entrada

```
set(handles.uitable1,'visible','on','Data',Separ);
```

```
set(handles.uitable1,'ColumnWidth',{30,30,30,30});
```

```
Garfi=stairs(handles.axes1,Escala2,transpose(Separ));
```

```
xlim(handles.axes1,[Escala2(4)/10,Escala2(4)+Escala2(4)/10]);
```

```
ylim(handles.axes1,[-0.2,1.2]);
```

```
legend(handles.axes1,'imagen1','imagen2','imagen3','imagen4');
```

```
set(Garfi(1),'LineWidth',4.5,'LineStyle','--','Color','b');
```

```
set(Garfi(2),'LineWidth',3.5,'LineStyle','-','Color','k');
```

```
set(Garfi(3),'LineWidth',2.5,'LineStyle',':','Color','r');
```

```
set(Garfi(4),'LineWidth',1.5,'LineStyle','-','Color','m');
```

```
title(handles.axes1,'Símbolos de datos')
```

```
xlabel(handles.axes1,'Tiempo');
```

```
ylabel(handles.axes1,'Valor Binario');
```

### %Preparación remota de los datos para la gráfica 3D

```
plot3(handles.axes4,Ex,Ey,Datos3);
```

```
grid(handles.axes4,'minor')
```

```
title(handles.axes4,'Símbolos de datos')
```

```
xlabel(handles.axes4,'Tiempo');
```

```
ylabel(handles.axes4,'Imágenes');
```

```
zlabel(handles.axes4,'Valor Binario');
```

### %Manejo de los datos antes de la modulación

```
Imag1=bi2de(Separ,'left-msb');
```

```
M = modem.qammod('M',16); % función de modulación 16 QAM
```

### %Mapa del código Gray

```
mapa16 = bin2gray(Imag1,'qam',16);
```

### % Modulación / Constelación

```

mod_data =modulate(M,mapa16);
%Presentación de los datos de salida
set(handles.uitable2,'visible','on','Data',mod_data);
set(handles.uitable2,'ColumnWidth',{100});
refqam = [-3-3i; -3-1i;-3+1i;-3+3i;-1-3i; -1-1i;-1+1i;-1+3i;1-3i;1-1i;1+1i;1+3i;3-3i;3-1i;3+1i;3+3i];
plot(handles.axes2,real(refqam),imag(refqam),'r+',real(mod_data),imag(mod_data),'bo',real([-3i; 3i]),imag([-3i; 3i]),'k-',real([-3; 3]),imag([-3; 3]),'k-');
title(handles.axes2,'Mapa 16 QAM')
xlabel(handles.axes2,'Fase');
ylabel(handles.axes2,'Cuadratura');
xlim(handles.axes2,[-3.5,3.5]);
ylim(handles.axes2,[-3.5,3.5]);

%Salida de los datos
handles.refqam=refqam;
handles.mod_data=mod_data;
guidata(hObject,handles);

```

## Sección 7.- Inserción de pilotos

```
function pushbutton5_Callback(hObject, eventdata, handles)
```

```
%datos a la entrada para insertar pilotos
```

```
%Control de tablas, botones y datos
```

```
set(handles.pushbutton6,'Enable','on');
```

```

set(handles.pushbutton13,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton14,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton15,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton16,'BackgroundColor',[.59 1 1]);
set(handles.pushbutton17,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton18,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton19,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton20,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton21,'BackgroundColor',[.94 .94 .94]);

```

```
%Entrada de datos
```

```

mod_data=handles.mod_data;
Escala2=handles.Escala2;
Separ=handles.Separ;
refqam=handles.refqam;
cla(handles.axes4,'reset');
set(handles.axes4,'visible','off');

```

```

cla(handles.axes5,'reset');
set(handles.axes5,'visible','off');
set(handles.uitable1,'visible','on','Data',mod_data);
set(handles.uitable1,'ColumnWidth',{100});
%Presentación de los datos de entrada
refqam = [-3-3i; -3-1i;-3+1i;-3+3i;-1-3i; -1-1i;-1+1i;-1+3i;1-3i;1-1i;1+1i;1+3i;3-3i;3-1i;3+1i;3+3i];
plot(handles.axes1,real(refqam),imag(refqam),'r+',real(mod_data),imag(mod_data),'bo',real([-3i; 3i]),imag([-3i; 3i]),'k-',real([-3; 3]),imag([-3; 3]),'k-');
title(handles.axes1,'Mapa 16 QAM')
xlabel(handles.axes1,'Fase');
ylabel(handles.axes1,'Cuadratura');
xlim(handles.axes1,[-3.5,3.5]);
ylim(handles.axes1,[-3.5,3.5]);

cla(handles.axes2);
%Insertar piloto
Pilot=[0; mod_data(1); 0; mod_data(2); 0; mod_data(3); 0; mod_data(4); 0];
Px=(Escala2(4)/4)*(1:9);
%Presentación de los datos de salida
set(handles.uitable2,'visible','on','Data',Pilot);
set(handles.uitable2,'ColumnWidth',{100});
PGraf=plot(handles.axes2,Px,[real(Pilot),imag(Pilot)],[0;Px(9)+Px(9)/10],[real([0; 0]),imag([0;0])],'k-');
set(PGraf(1),'LineWidth',3,'Color','b');
set(PGraf(2),'LineWidth',1.5,'Color','r');
title(handles.axes2,'Símbolo con Piloto')
xlabel(handles.axes2,'Tiempo');
ylabel(handles.axes2,'Valor');
xlim(handles.axes2,[0,Px(9)+Px(9)/10]);
ylim(handles.axes2,[-3.5,3.5]);

%Salida de los datos
handles.Pilot=Pilot;
guidata(hObject,handles);

```

## Sección 8.- Modulación IFFT

```

function slider2_Callback(hObject, eventdata, handles)
%Esta codificación permite controlar la longitud de la salida del IFFT
Expo=2^round(get(handles.slider2,'Value'));
set(handles.text2,'String',Expo);
if Expo == 4
    set(handles.popupmenu1,'String',{'2'});

```

```

end
if Expo == 8
    set(handles.popupmenu1,'String',{'2','4'});
end
if Expo == 16
    set(handles.popupmenu1,'String',{'2','4','8'});
end
if Expo == 32
    set(handles.popupmenu1,'String',{'2','4','8','16'});
end
if Expo == 64
    set(handles.popupmenu1,'String',{'2','4','8','16','32'});
end
if Expo >= 128
    set(handles.popupmenu1,'String',{'2','4','8','16','32','64'});
end
handles.Expo=Expo;
guidata(hObject,handles);

```

#### Sección 9.- Aplicar IFFT **Botón 6**

```
function pushbutton6_Callback(hObject, eventdata, handles)
```

```
%Aplicación del IFFT
```

```
%Control de tablas, botones y datos
```

```

set(handles.pushbutton8,'Enable','on');
set(handles.pushbutton12,'Enable','on');
cla(handles.axes4,'reset');
set(handles.axes4,'visible','off');
cla(handles.axes5,'reset');
set(handles.axes5,'visible','off');

```

```

set(handles.pushbutton13,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton14,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton15,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton16,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton17,'BackgroundColor',[.59 1 1]);
set(handles.pushbutton18,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton19,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton20,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton21,'BackgroundColor',[.94 .94 .94]);

```

```
%Entrada de datos
```

```

Pilot=handles.Pilot;
Escala2=handles.Escala2;
Expo=2^round(get(handles.slider2,'Value'));

```

### %Presentación de los datos de entrada

```
cla(handles.axes1,'reset');
Px=(Escala2(4)/4)*(1:9);
set(handles.uitable1,'visible','on','Data',Pilot);
set(handles.uitable1,'ColumnWidth',{100});
PGraf=plot(handles.axes1,Px,[real(Pilot),imag(Pilot)],[0;Px(9)+Px(9)/10],[real([0;
0]),imag([0;0]),'k-');
set(PGraf(1),'LineWidth',3,'Color','b');
set(PGraf(2),'LineWidth',1.5,'Color','r');
title(handles.axes1,'Símbolo con Piloto')
xlabel(handles.axes1,'Índice');
ylabel(handles.axes1,'Valor');
xlim(handles.axes1,[0,Px(9)+Px(9)/10]);
ylim(handles.axes1,[-3.5,3.5]);

multip=2^(get(handles.popupmenu1,'Value'));
```

### %IFFT

```
modular=ifft(Pilot,Expo);
Fs=str2num(handles.edit2.String);
T=1/Fs;
tiempo=linspace(0,T*Expo/multip,Expo/multip);
modular2=reshape(modular,[],multip);
ffx=repmat(1:multip,Expo/multip,1);
ffy=repmat((tiempo)',1,multip);
```

### %Presentación de los datos de salida

```
set(handles.uitable2,'Data',modular2);
set(handles.uitable2,'ColumnWidth',{100});
cla(handles.axes2,'reset');
plot3(handles.axes2,ffx,ffx,real(modular2));
grid(handles.axes2,'minor')
title(handles.axes2,'Imágenes IFFT')
xlabel(handles.axes2,'Tiempo');
ylabel(handles.axes2,'Imágenes');
zlabel(handles.axes2,'Amplitud');
```

### %Salida de los datos

```
handles.Fs=Fs;
handles.T=T;
handles.Expo=Expo;
handles.multip=multip;
handles.tiempo=tiempo;
handles.modular2=modular2;
handles.ffx=ffx;
```

```
handles.ffy=ff;
guidata(hObject,handles);
```

## Sección 10.- Grafica Auxiliar Botón 12

```
function pushbutton12_Callback(hObject, eventdata, handles)
%Gráfica auxiliar para la IFFT
modular2=handles.modular2;
ffx=handles.ffx;
ffy=handles.ffy;
figure('Name','IFFT superficie');
mesh(ffy,ffx,real(modular2));
title('Imágenes IFFT')
xlabel('Tiempo');
ylabel('Imágenes');
zlabel('Amplitud');
```

## Sección 11.- Inserción de guarda botón 8

```
function pushbutton8_Callback(hObject, eventdata, handles)
%datos a la entrada para aplicar la guarda
```

### %Control de tablas, botones y datos

```
set(handles.pushbutton9,'Enable','on');
cla(handles.axes4,'reset');
set(handles.axes4,'visible','off');
cla(handles.axes5,'reset');
set(handles.axes5,'visible','on');
set(handles.pushbutton13,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton14,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton15,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton16,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton17,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton18,'BackgroundColor',[.59 1 1]);
set(handles.pushbutton19,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton20,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton21,'BackgroundColor',[.94 .94 .94]);
```

### %Entrada de datos

```
tiempo=handles.tiempo;
T=handles.T;
modular2=handles.modular2;
multip=handles.multip;
Expo=handles.Expo;
```

```
ffx=handles.ffx;  
ffy=handles.ffy;
```

```
%Presentación de los datos de entrada  
set(handles.uitable1,'visible','on','Data',modular2);  
set(handles.uitable1,'ColumnWidth',{100});  
cla(handles.axes1,'reset');  
grid minor  
plot3(handles.axes1,ffx,ffx,real(modular2));  
grid(handles.axes1,'minor')  
title(handles.axes1,'Imágenes IFFT')  
xlabel(handles.axes1,'Tiempo');  
ylabel(handles.axes1,'Imágenes');  
zlabel(handles.axes1,'Amplitud');
```

```
%Presentación de los datos de salida  
guard=vertcat(modular2(end,:),modular2);  
tiempo2=(0:Expo/multip)*T;  
ffx2= repmat(1:multip,Expo/multip+1,1);  
ffx2= repmat((tiempo2)',1,multip);  
set(handles.uitable2,'visible','on','Data',guard);  
set(handles.uitable2,'ColumnWidth',{100});  
plot3(handles.axes2,ffx2,ffx2,real(guard));  
grid(handles.axes2,'minor')  
title(handles.axes2,'Portadoras con guarda')  
xlabel(handles.axes2,'Tiempo');  
ylabel(handles.axes2,'Imágenes');  
zlabel(handles.axes2,'Amplitud');  
mesh(handles.axes5,ffx2,ffx2,real(guard));  
grid(handles.axes5,'minor')  
title(handles.axes5,'Portadoras con guarda')  
xlabel(handles.axes5,'Tiempo');  
ylabel(handles.axes5,'Imágenes');  
zlabel(handles.axes5,'Amplitud');
```

```
%Salida de los datos  
handles.guard=guard;  
handles.ffx2=ffx2;  
handles.ffx2=ffx2;  
guidata(hObject,handles);
```

Sección 11.- Conversión Paralelo/ Serial botón 9.

```
function pushbutton9_Callback(hObject, eventdata, handles)  
%Conversión de Paralelo a Serial
```

### %Control de tablas, botones y datos

```
set(handles.pushbutton10,'Enable','on');
cla(handles.axes4,'reset');
set(handles.axes4,'visible','on');
cla(handles.axes5,'reset');
set(handles.axes5,'visible','off');
```

```
set(handles.pushbutton13,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton14,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton15,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton16,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton17,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton18,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton19,'BackgroundColor',[.59 1 1]);
set(handles.pushbutton20,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton21,'BackgroundColor',[.94 .94 .94]);
```

### %Entrada de datos

```
guard=handles.guard;
ffx2=handles.ffx2;
ffx2=handles.ffx2;
set(handles.uitable1,'visible','on','Data',guard);
set(handles.uitable1,'ColumnWidth',{100});
```

### %Presentación de los datos de entrada

```
plot3(handles.axes1,ffx2,ffx2,guard);
grid(handles.axes1,'minor')
title(handles.axes1,'Portadoras con guarda')
xlabel(handles.axes1,'Tiempo');
ylabel(handles.axes1,'Imágenes');
zlabel(handles.axes1,'Amplitud');
mesh(handles.axes4,ffx2,ffx2,real(guard));
grid(handles.axes4,'minor')
title(handles.axes4,'Portadoras con guarda')
xlabel(handles.axes4,'Tiempo');
ylabel(handles.axes4,'Imágenes');
zlabel(handles.axes4,'Amplitud');
```

### %Presentación de los datos de salida

```
senal=reshape(guard,[],1);
set(handles.uitable2,'visible','on','Data',senal);
set(handles.uitable2,'ColumnWidth',{100});
mesh(handles.axes2,ffx2,ffx2,real(guard));
```

```
grid(handles.axes2,'minor')
title(handles.axes2,'Conversión P/S')
xlabel(handles.axes2,'Tiempo');
ylabel(handles.axes2,'Señal');
zlabel(handles.axes2,'Amplitud');
```

```
%Salida de los datos
```

```
handles.senal=senal
guidata(hObject,handles);
```

Sección 12.- Transmisión Botón 10

```
function pushbutton10_Callback(hObject, eventdata, handles)
```

```
%Transmisión
```

```
%Control de tablas, botones y datos
```

```
set(handles.pushbutton11,'Enable','on');
cla(handles.axes4,'reset');
set(handles.axes4,'visible','off');
```

```
set(handles.pushbutton13,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton14,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton15,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton16,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton17,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton18,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton19,'BackgroundColor',[.94 .94 .94]);
set(handles.pushbutton20,'BackgroundColor',[.59 1 1]);
set(handles.pushbutton21,'BackgroundColor',[.94 .94 .94]);
```

```
%Entrada de datos
```

```
guard=handles.guard;
ffx2=handles.ffx2;
ffy2=handles.ffy2;
senal=handles.senal;
Fs=handles.Fs;
freq2=Fs/2*linspace(-1,1,Fs);
```

```
%Presentación de los datos de entrada
```

```
set(handles.uitable1,'visible','on','Data',senal);
set(handles.uitable1,'ColumnWidth',{100});
mesh(handles.axes1,ffx2,ffy2,real(guard));
grid(handles.axes1,'minor')
title(handles.axes1,'Conversión P/S')
xlabel(handles.axes1,'Tiempo');
```

```
ylabel(handles.axes1,'Señal');  
zlabel(handles.axes1,'Amplitud');
```

```
%Presentación de los datos de salida
```

```
axes(handles.axes2);  
periodogram(senal);  
set(handles.axes5,'Visible','on');  
axes(handles.axes5);  
spectrogram(senal,'yaxis');  
view(-45,20);  
set(handles.uitable2,'visible','on','Data',senal);
```

Sección 13.- Canal botón 11

```
function pushbutton11_Callback(hObject, eventdata, handles)  
% Señal transmitida a través de un cable AWG
```

```
%Entrada de los datos
```

```
senal=handles.senal;  
axes(handles.axes1);  
periodogram(senal);  
set(handles.axes4,'Visible','on');
```

```
set(handles.pushbutton13,'BackgroundColor',[.94 .94 .94]);  
set(handles.pushbutton14,'BackgroundColor',[.94 .94 .94]);  
set(handles.pushbutton15,'BackgroundColor',[.94 .94 .94]);  
set(handles.pushbutton16,'BackgroundColor',[.94 .94 .94]);  
set(handles.pushbutton17,'BackgroundColor',[.94 .94 .94]);  
set(handles.pushbutton18,'BackgroundColor',[.94 .94 .94]);  
set(handles.pushbutton19,'BackgroundColor',[.94 .94 .94]);  
set(handles.pushbutton20,'BackgroundColor',[.94 .94 .94]);  
set(handles.pushbutton21,'BackgroundColor',[.59 1 1]);
```

```
%Presentación de los datos de entrada
```

```
axes(handles.axes4);  
spectrogram(senal,'yaxis');  
view(-45,20);  
set(handles.uitable1,'visible','on','Data',senal);
```

```
%Presentación de los datos de salida
```

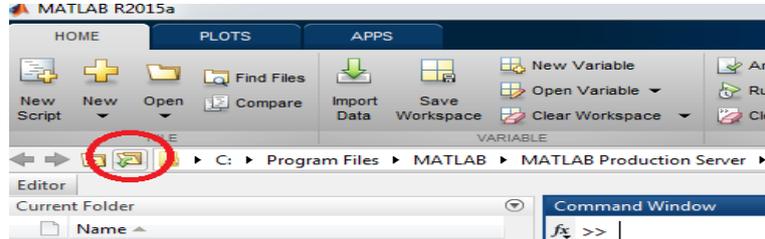
```
senaltx=awgn(senal,.2);  
set(handles.uitable2,'visible','on','Data',senaltx);  
axes(handles.axes2);  
periodogram(senaltx);  
set(handles.axes5,'Visible','on');
```

```
axes(handles.axes5);  
spectrogram(senaltx,'yaxis');  
view(-45,20);
```

### Anexo III Instalación de Archivo de configuración en Matlab

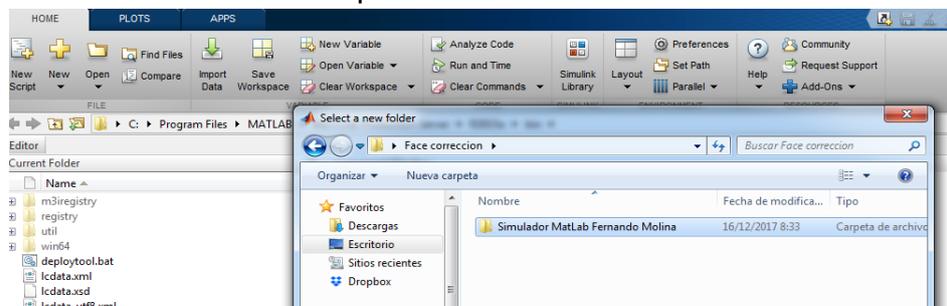
1.- Guardar el archivo en una carpeta

2.- Vamos a la pestaña Browser for folder de la aplicación Matlab

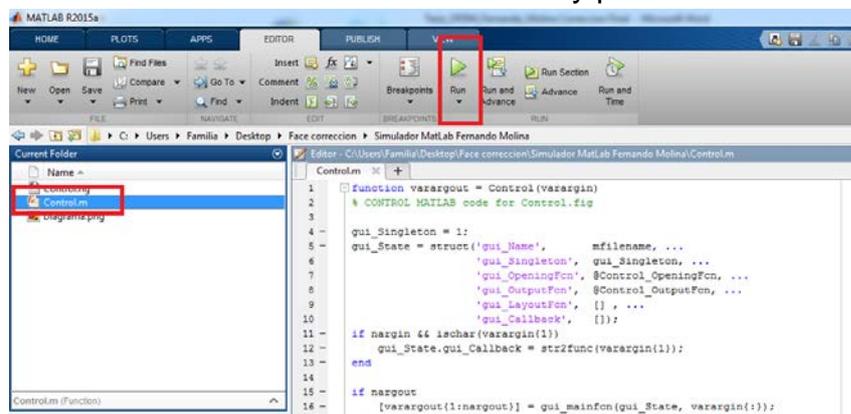


3.- Elegimos la carpeta donde guardamos nuestro archivo de configuraciones

C:\Users\Familia\Desktop\Face correccion\Simulador MatLab Fernando Molina



4.- Seleccionamos el archivo control.m y presionamos RUN



5.- El programa se ejecutara

