

# **Simulación de OFDM en MATLAB con fin didáctico**

**Carlos Dolz Fernández**

Grado en Ingeniería de  
Tecnologías y Servicios de  
Telecomunicación  
Sistemas de Comunicación

**Tutor/a de TF**

Rubén Molina Casasnovas

**Profesor/a responsable de  
la asignatura**

Carlos Monzo Sánchez

Junio 2023



Esta obra está sujeta a una licencia de Reconocimiento-  
NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## Ficha del Trabajo Final

<b>Título del trabajo:</b>	Simulación de OFDM en MATLAB con fin didáctico
<b>Nombre del autor/a:</b>	Carlos Dolz Fernández
<b>Nombre del Tutor/a de TF:</b>	Rubén Molina Casasnovas
<b>Nombre del/de la PRA:</b>	Carlos Monzo Sánchez
<b>Fecha de entrega:</b>	06/2023
<b>Titulación o programa:</b>	Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
<b>Área del Trabajo Final:</b>	Sistemas de Comunicación
<b>Idioma del trabajo:</b>	Castellano
<b>Palabras clave</b>	OFDM, MATLAB, simulación
<b>Resumen del Trabajo</b>	
<p>En el estudio de las diferentes modulaciones disponibles para transmitir una señal el alumnado de Telecomunicaciones se encuentra con abundante información en forma de ecuaciones de carácter complejo. Según se vuelve más sofisticada la modulación, tanto más difícil se hace la comprensión.</p> <p>En el caso concreto de la modulación OFDM (multiplexación por división en frecuencias ortogonales), se trata de un sistema que agrupa modulaciones lineales para mejorar la efectividad en condiciones muy severas de canal de comunicaciones, sin necesidad de utilizar un ecualizador complicado en el receptor. Por ello se ha pretendido mostrar una simulación parametrizable que ayude a su asimilación.</p> <p>Se ha utilizado para el trabajo la herramienta MATLAB, que ofrece un entorno de programación avanzada, incluyendo funciones propias que encapsulan el funcionamiento de sistemas usando esta modulación, y ofrece una salida visual en forma gráfica muy amigable para el usuario final. Se ha implementado la simulación de la respuesta en frecuencia a una entrada simulada de un sistema con modulación OFDM, con los valores de los datos de entrada generados aleatoriamente, y parametrizando valores como el número de portadoras, de forma que la respuesta obtenida sea clarificadora en el proceso de estudio de esta modulación.</p>	

De esta manera, se obtuvo un software que es capaz de mostrar gráficamente la respuesta de un sistema que utiliza la modulación OFDM mediante la programación en MATLAB. Se espera así facilitar su estudio ofreciendo una herramienta donde visualmente se pueda observar el resultado de la modulación.

### **Abstract**

In the study of the different modulations available to transmit a signal, Telecommunications students find abundant information in the form of complex equations. As modulation becomes more sophisticated, the more difficult understanding becomes.

The specific case of OFDM modulation (orthogonal frequencies division multiplexing), is a system that groups linear modulations to improve effectiveness in very severe communication channel conditions, without the need to use a complicated equalizer in the receiver. Therefore, it has been intended to show a parameterizable simulation that helps its assimilation.

The MATLAB tool used for the work offers an advanced programming environment, including its own functions that encapsulate the operation of systems using this modulation, and offers a visual output in graphic form very friendly to the end user. The simulation of the frequency response to a simulated input of a system with OFDM modulation has been implemented, with the values of the input data generated randomly, and parameterizing values such as the number of carriers, so that the response obtained is clarifying in the process of studying this modulation.

In this way, a software was obtained that is capable of graphically displaying the response of a system that uses OFDM modulation through programming in MATLAB. It is hoped to facilitate its study by offering a tool where the modulation result can be visually observed.

# Index

1.	Introducción	1
1.1.	Contexto y justificación del Trabajo	1
1.2.	Objetivos del Trabajo	1
1.3.	Impacto en sostenibilidad, ético-social y de diversidad	2
1.4.	Enfoque y método seguido	2
1.5.	Planificación del trabajo	2
1.6.	Breve sumario de productos obtenidos	4
1.7.	Breve descripción de otros capítulos de la memoria	4
2.	Materiales y métodos	5
2.1.	Estudio de la modulación OFDM	5
2.2.	Simulaciones OFDM en MATLAB	9
2.3.	Descripción de la solución propuesta	11
3.	Resultados	25
3.1.	Descripción de los resultados	25
3.2.	Resultados según la parametrización de valores	27
4.	Conclusiones	34
5.	Trabajos futuros	35
6.	Glosario	36
7.	Bibliografía	38
8.	Anexos	41

# Lista de Figuras

Figura 1: Diagrama de Gantt	3
Figura 2: El concepto <i>multicarrier</i>	5
Figura 3: Formas espectrales superpuestas ortogonales para OFDM	6
Figura 4: Diagrama de bloques de la implementación OFDM	7
Figura 5: Introducción del tiempo de guarda	8
Figura 6: Espectro de densidad de potencia de una señal OFDM con tiempo de guarda en escala lineal (a) y en escala logarítmica (b)	9
Figura 7: Panel "Parámetros OFDM"	14
Figura 8: Selección de puntos FFT	15
Figura 9: Panel de gestión de subportadoras	15
Figura 10: Supresión de portadoras laterales	16
Figura 11: Selección de eliminar portadora central	16
Figura 12: Número de portadoras de datos	16
Figura 13: Botón <i>Mapping</i> portadoras	16
Figura 14: Selección de prefijo cíclico	17
Figura 15: Número de símbolos OFDM	17
Figura 16: <i>Oversampling</i>	17
Figura 17: <i>Windowing</i>	17
Figura 18: Botón Aplicar	18
Figura 19: Panel de señal temporal	18
Figura 20: Panel de espectro frecuencial	18
Figura 21: Vista de diseño en MATLAB <i>App Designer</i>	19
Figura 22: Aplicación de simulación de señal OFDM	25
Figura 23: Ventana <i>Mapping</i> portadoras	26
Figura 24: Señal generada con distintos números de puntos FFT	27
Figura 25: Aplicación de supresión de subportadoras	28
Figura 26: Aplicación de supresión de subportadora central	29
Figura 27: Inclusión de prefijo cíclico	30
Figura 28: Señal OFDM con prefijo cíclico	30
Figura 29. Señal con diferentes valores de <i>oversampling</i>	31
Figura 30: Aplicación de <i>windowing</i>	32

Figura 31: Señal OFDM con *windowing*



# 1. Introducción

En este primer capítulo de la memoria nos situaremos en el contexto y la justificación del trabajo, se fijarán los objetivos, su impacto ético social, la planificación de éste para llevarlo a cabo y finalizará con una breve descripción del resto de capítulos.

## 1.1. Contexto y justificación del Trabajo

El estudio de modulaciones de señales para su transmisión es el eje central de cualquier disciplina de telecomunicaciones, ya que abarca todo el proceso desde el punto de vista del emisor, canal y receptor. Es un aspecto que se ha ido perfeccionando a lo largo del tiempo, desde unas modulaciones analógicas sencillas iniciales hasta las más complejas digitales utilizadas en la actualidad, que optimizan el proceso en cuanto a cantidad, calidad y velocidad de transmisión de datos.

Esta complejidad influye en su estudio, ya que aparte de la agrupación de modulaciones sencillas que se realiza, influyen aspectos más difíciles de asimilar como un amplio número de señales portadoras con distintos tiempos de transmisión y de guarda entre ellas, etc.

Mediante esta simulación se ha pretendido facilitar la comprensión de algunos aspectos de esta modulación, para que de forma gráfica se puedan observar las consecuencias de introducir diferentes parámetros en el sistema encargado de realizar la transmisión de datos.

## 1.2. Objetivos del Trabajo

Los objetivos que se quieren alcanzar con este trabajo son:

- Estudio de la modulación OFDM, para comprender todos los aspectos involucrados en su realización.
- Diseño e implementación de un programa en MATLAB que simule la modulación OFDM de una señal aleatoria, con diferentes parámetros de entrada (número de portadoras utilizadas, inclusión de prefijo cíclico, ...).
- Análisis de los resultados obtenidos para los diferentes casos propuestos, verificando la exactitud del modelo simulado, así como los resultados que ofrece.

### 1.3. Impacto en sostenibilidad, ético-social y de diversidad

Este proyecto tiene un aspecto principalmente técnico, por lo que, en cuanto a la sostenibilidad, no impacta con ninguna de los ODS (Objetivos de Desarrollo Sostenible 2030, ONU [1]) con los que la UOC está públicamente comprometida.

Respecto a la dimensión comportamiento ético y de responsabilidad social (RS), por las características de este trabajo, no tiene ningún impacto ni positivo ni negativo en aspectos ético-sociales.

Con relación a la diversidad, al ser un trabajo técnico, no tiene una gran implicación con los ODS de esta dimensión. Aun así, se han considerado todas las aportaciones documentales con independencia del género de la autora o autor. También se ha optado por hacer un uso flexible del lenguaje inclusivo allí donde ha sido posible.

### 1.4. Enfoque y método seguido

Para la implementación de la simulación propuesta, primero se ha estudiado la teoría de la modulación OFDM, buscando la solución que más se pueda beneficiar de los resultados que se obtienen.

Para el desarrollo del trabajo, se ha utilizado programación en MATLAB, considerando que esta aplicación ofrece funciones ya construidas que implementan partes del sistema de modulación en el transmisor y demodulación en el receptor. De esta forma, se ha adecuado su uso para conseguir los datos simulados requeridos.

### 1.5. Planificación del trabajo

Para realizar la planificación del trabajo se ha dividido este en 5 fases, coincidiendo su final con las entregas parciales a realizar en el aula virtual:

- Definición. Del 01/03/2023 al 12/03/2023 (9 jornadas).
- Estado del arte. Del 13/03/2023 al 26/03/2023 (11 jornadas).
- Diseño e implementación. Del 27/03/2023 al 19/05/2023 (40 jornadas).
- Memoria. Del 20/05/2023 al 12/06/2023 (16 jornadas).
- Defensa. Del 13/06/2023 al 22/06/2023 (8 jornadas).

El esquema de la planificación temporal es:

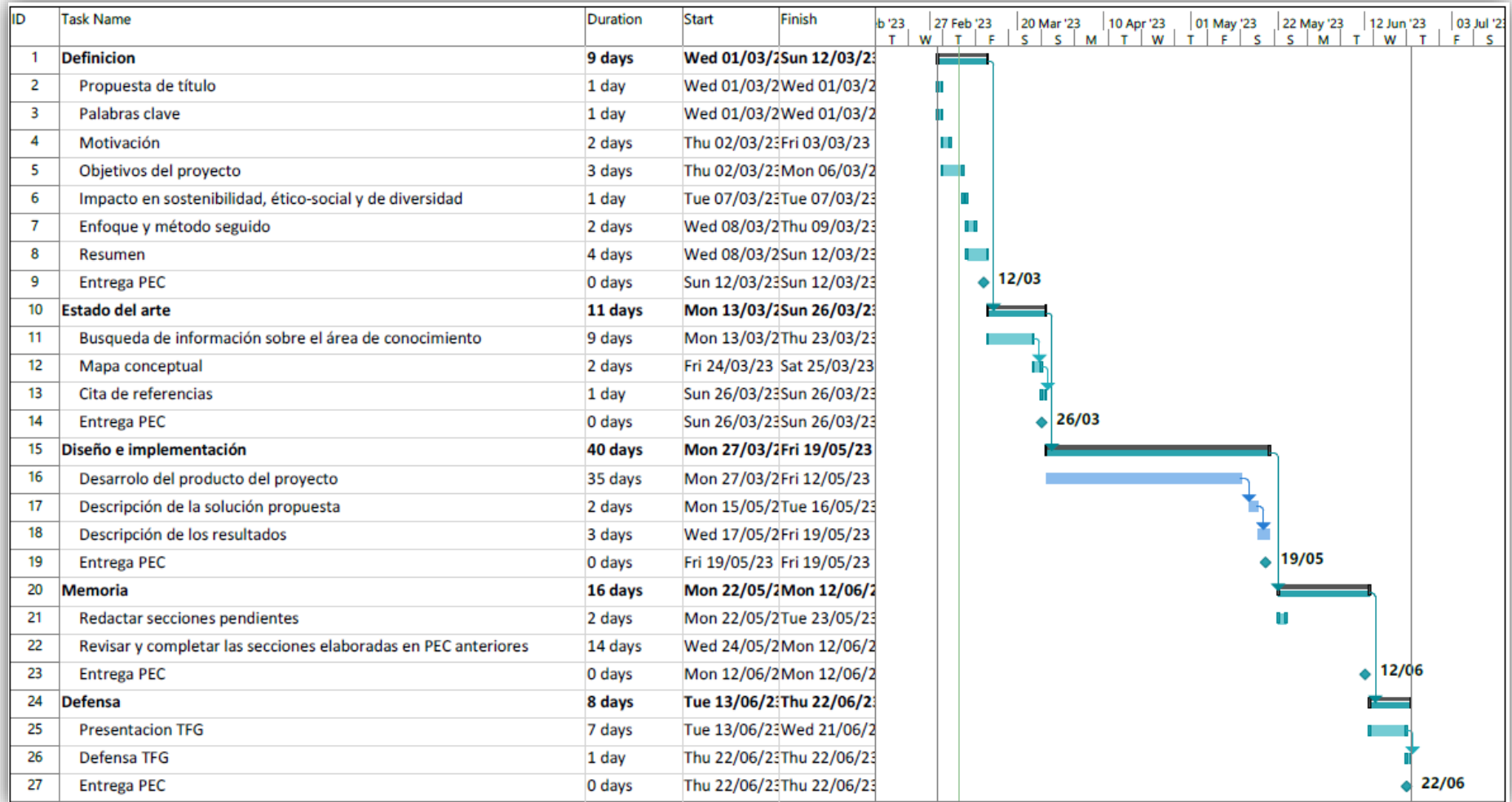


Figura 1: Diagrama de Gantt

## 1.6. Breve resumen de productos obtenidos

Al final del trabajo se obtendrán los siguientes productos:

- La propia memoria del trabajo, que incluye reflexiones y metodología a utilizar en el diseño de simulaciones sobre tecnologías similares utilizando MATLAB.
- La aplicación de simulación de la señal modulada con OFDM, con la parametrización disponible y los resultados gráficos.
- El código fuente de la aplicación desarrollada en MATLAB, que permite por lado la construcción visual de la aplicación y por otro, la modulación OFDM de una señal.

## 1.7. Breve descripción de otros capítulos de la memoria

En el capítulo materiales y métodos se analiza primero las características teóricas de la modulación OFDM, para a continuación estudiar el estado del arte en cuanto a publicaciones similares, y finalmente detallar en la descripción de la solución propuesta la metodología, funciones y componentes utilizados para realizar la aplicación que gestiona la simulación.

En el siguiente capítulo se presentan los resultados obtenidos mediante el uso de la aplicación, tanto en la configuración de parámetros para realizar la simulación de una señal modulada como en los gráficos obtenidos resultado de su aplicación, analizando el porqué de estos.

A continuación, se detallan las conclusiones, reflexión y análisis crítico de los resultados obtenidos, posible impacto en sostenibilidad, ético-social y de diversidad del proyecto y líneas de trabajo futuro que han surgido tras la ejecución del trabajo.

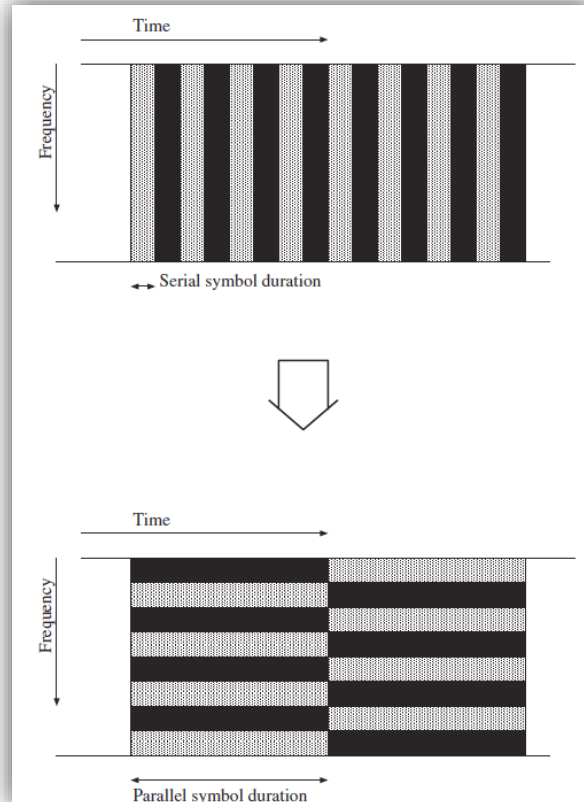
Después aparecen el glosario y la bibliografía utilizados en la confección del trabajo. Por último, en el anexo se podrá encontrar el código MATLAB de la aplicación desplegada.

## 2. Materiales y métodos

El trabajo propuesto consiste en la realización de una simulación de una modulación OFDM en un entorno de programación como MATLAB con fines didácticos. Por ello, primero veremos un resumen de las características y comportamiento teórico de OFDM y, a continuación, cómo se pueden trasladar los aspectos con características gráficas al mundo de la simulación a través de MATLAB.

### 2.1. Estudio de la modulación OFDM

La multiplexación por división en frecuencias ortogonales, en inglés *orthogonal frequency division multiplexing* (OFDM) se consigue sumando  $N$  modulaciones lineales (típicamente de tipo QAM) usando  $N$  portadoras diferentes, siendo  $N$  un número entero relativamente alto (64, 1024, ...). De esta manera, el ancho de banda disponible para la transmisión se reparte entre cada una de las modulaciones QAM que la conforman repartiéndose entre ellas la secuencia original de símbolos a transmitir [2].

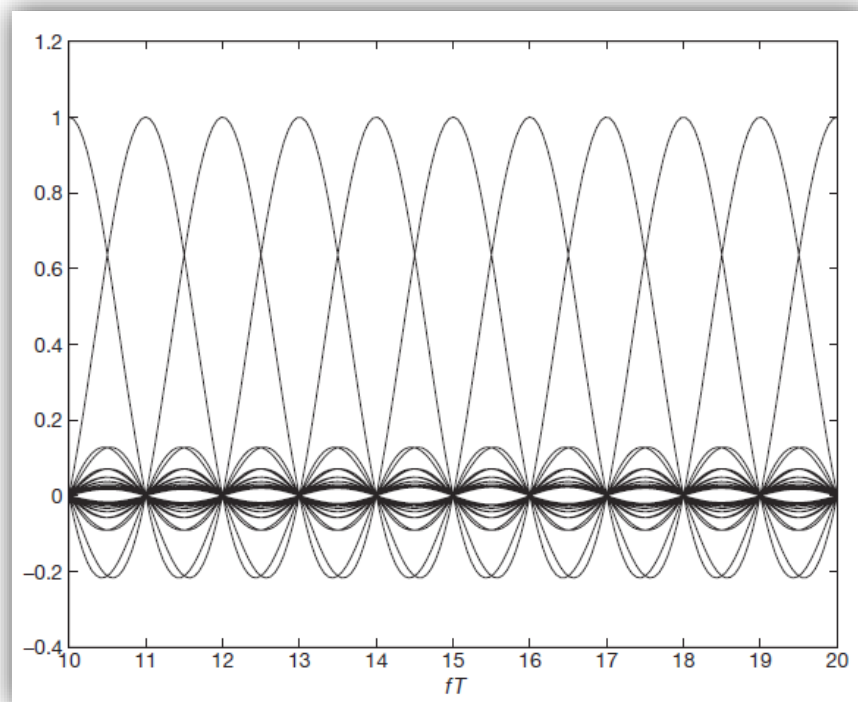


**Figura 2:** El concepto *multicarrier*

La estrategia alternativa que ofrece OFDM respecto a la modulación de portadora única implica que cada símbolo se tramite ocupando un tiempo mayor y un ancho de banda menor que en esta.

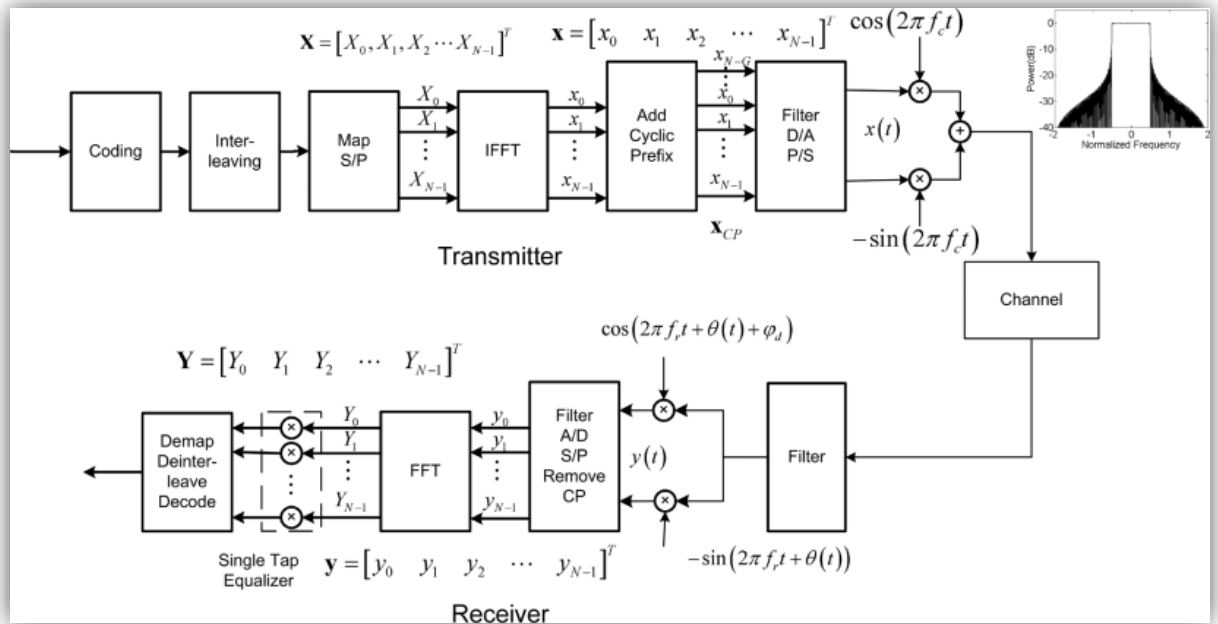
En un diagrama de ocupación tiempo-frecuencia, donde cada rectángulo, ya sea negro o gris, representa la ocupación de un símbolo dado, se puede observar la diferencia entre ellas [3].

Las señales portadoras deben mantener la condición de ortogonalidad para que en el receptor se puedan separar los símbolos que corresponden a cada una de ellas sin interferencias. El algoritmo utilizado para mantener esta condición es IFFT (*Inverse Fast Fourier Transform*) en la modulación y FFT (*Fast Fourier Transform*) en la demodulación. El algoritmo IFFT es útil para enviar cada símbolo QAM a transmitir a la frecuencia que debe ocupar en el equivalente paso bajo.



**Figura 3:** Formas espectrales superpuestas ortogonales para OFDM

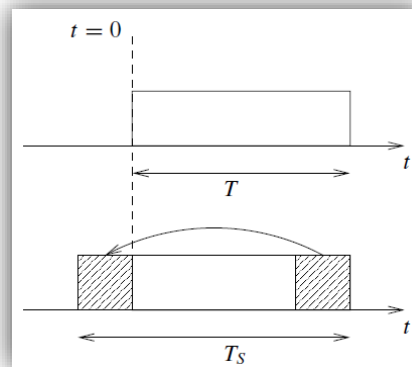
La secuencia completa de transmisión a través de un canal analógico y recepción de un conjunto de símbolos expresada en un diagrama de bloques utilizando OFDM [4] podría ser:



**Figura 4:** Diagrama de bloques de la implementación OFDM

En un canal de comunicación real la señal transmitida experimenta distintos rebotes sobre los obstáculos físicos que rodean al transmisor y el receptor y, en lugar de recibir una única replica de la señal se recibe un conjunto de réplicas debido a los diferentes ecos o retardos respecto al haz o camino principal entre ambos. Este fenómeno está caracterizado por el parámetro dispersión del retardo o *delay spread*. Una de las principales ventajas de utilizar la modulación multiportadora OFDM es que el efecto de la dispersión del retardo del canal se verá mitigado y, por lo tanto, la tarea de ecualización será mucho más sencilla de realizar.

Además, mediante la introducción de un tiempo de guarda entre símbolo y símbolo, durante el que se transmite una repetición parcial del siguiente símbolo, o prefijo cíclico, se evita la interferencia intersimbólica (ISI) y se preserva la ortogonalidad entre las diferentes frecuencias portadoras, evitando la interferencia entre portadoras (ICI), siempre que se cumpla la condición de que la duración del tiempo de guarda sea mayor o igual que la duración de la respuesta impulsional del canal.



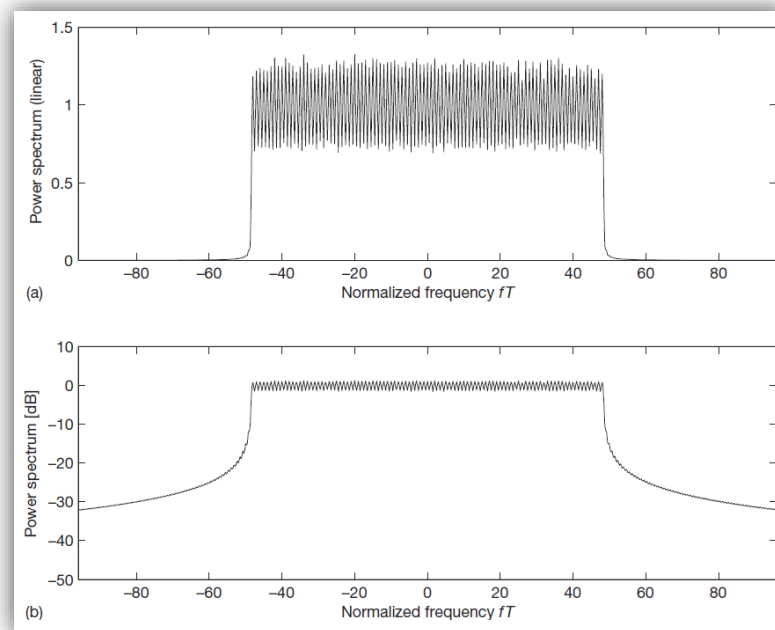
**Figura 5:** Introducción del tiempo de guarda

Para funcionalidades de sincronismo y demodulación (estimando la función de transferencia del canal) se introducen los símbolos piloto, a unas determinadas frecuencias portadoras y con una determinada periodicidad temporal, que no transportan información, sino bits de datos ya conocidos por el receptor. Esta estrategia implica que se pierda eficiencia, ya que disminuye la velocidad de transmisión de símbolos útiles respecto al ancho de banda.

Cuando la modulación OFDM se utiliza tras procesar la secuencia de bits por un codificador de canal con el objetivo de corregir los errores producidos en la transmisión se denomina COFDM, del inglés *Coded OFDM*.

El espectro resultante de la modulación es muy importante, ya que es un recurso limitado y a repartir, por lo que hay estándares muy estrictos en su definición. Se usan estrategias de supresión de portadoras centrales y laterales y conformación de pulsos temporales para cumplir dichos requerimientos.





**Figura 6:** Espectro de densidad de potencia de una señal OFDM con tiempo de guarda en escala lineal (a) y en escala logarítmica (b)

OFDM es ampliamente utilizado como estándar en la radiodifusión europea de audio digital (DAB), la difusión de televisión digital terrestre (DVB-T), ambos favorecidos por el hecho de soportar retardos y ecos sobre la señal recibida, y los sistemas de comunicaciones inalámbricas de corto alcance y área local Wifi IEEE 802.11a,g y Wimax 802.16. Es también la tecnología clave en LTE y, convenientemente adaptado, usado en comunicaciones ópticas.

## 2.2. Simulaciones OFDM en MATLAB

En el entorno de programación MATLAB se pueden encontrar funcionalidades explicadas en el apartado anterior ya implementadas completamente, disponibles para su uso bajo una única función. Permiten diseñar, probar y simular formas de onda OFDM, personalizando sus parámetros, así como realizar llamadas a funciones específicas que generen formas de onda OFDM personalizadas [5].

Además, se puede hacer una simulación mediante bloques de todo el sistema de comunicación con la herramienta Simulink, que es un programa dentro de MATLAB que permite al personal encargado de su uso colocar visualmente bloques de funciones en una secuencia de mapeo para completar una matriz de diferentes aplicaciones del sistema. Para OFDM, Simulink ofrece modulación, ruido gaussiano, FFT, mapeo de constelaciones y

bloques de tasa de error que soportan directamente las tareas relacionadas con el procesamiento de señales en esta modulación [6].

En lo referente a trabajos ya realizados, encontramos diferentes aproximaciones en el ámbito de la simulación de la modulación OFDM en el entorno de MATLAB:

En el artículo [7] se realiza la simulación del núcleo del sistema de procesamiento de señal en OFDM. Se obtiene como resultado, a partir de una señal binaria generada aleatoriamente, su representación gráfica tras la modulación en el transmisor, la señal recibida tras la adición de ruido del canal de comunicación, el espectro de transmisión y el mensaje transmitido y recibido.

Vemos que en el artículo de Sha et al [8] la autoría establece el rendimiento de un sistema OFDM formado por transmisor, canal de comunicación y receptor, seleccionando los esquemas de implementación apropiados para cada módulo. Con la simulación se obtiene la probabilidad de error BER para diferentes relaciones señal-ruido SNR en un canal de comunicaciones AWGN.

Bao describe en [9] como obtener también mediante la simulación la relación BER/SNR, pero en este caso para tres posibles algoritmos de interpolación, constante, lineal y DFT, que se utilizan para hacer una estimación del canal en transmisiones inalámbricas.

El capítulo indicado en [10] hace una exposición detallada de la modulación OFDM de forma académica, pero intercalando pequeños ejemplos de código de MATLAB explicando alguno de los conceptos desarrollados, lo que lo sitúa más cercano al espíritu de este trabajo. Así, por ejemplo, hay un código que desarrolla el concepto de ortogonalidad de las señales que conforman el sistema multiportadora, o la BER para un sistema OFDM con 16-QAM.

Sacie Alcázar en su Trabajo Final de Máster [11] investiga sobre comunicaciones ópticas de señales moduladas mediante OFDM. Realiza la simulación en MATLAB de unos datos generados aleatoriamente mediante una modulación lineal DPQPSK, para posteriormente mapearlos de serie a paralelo, transformarlos mediante IFFT e insertar el símbolo cíclico, con lo que obtiene una gráfica del espectro resultante de la señal a transmitir. Después se centra en la transmisión en sí a través de un medio óptico, tema que se aleja del objetivo de este trabajo, para después ejecutar la recepción de la señal, mediante eliminación del prefijo cíclico, FFT y ecualización y estimación del canal, visualizando finalmente el espectro de la señal recibida.

En resumen, no abundan los trabajos que exploten la posibilidad de visualizar información a través de MATLAB respecto a los distintos conceptos involucrados en la modulación OFDM. Algunos textos obtienen un resultado particular, algo limitado, utilizando parámetros fijos, no proporcionan la flexibilidad necesaria para el alumnado que está tratando de comprender este tema tan complejo. Por ello, este trabajo tiene el objetivo de cubrir en la medida de lo posible la necesidad existente en cuanto al aprendizaje de esta modulación.

## 2.3. Descripción de la solución propuesta

El objetivo final de este trabajo es, por un lado, disponer de una herramienta de fácil configuración en la que observar los principales aspectos de la modulación OFDM de una forma amigable para el estudiantado que desee investigar en este aspecto.

Pero también, en la memoria del trabajo se indicará las diferentes opciones que se han barajado para la implementación del proyecto, así como las referencias a la documentación que explican las funcionalidades utilizadas, por lo que intentara tener un carácter lo más didáctico posible.

Para la realización y posterior ejecución del proyecto, es necesario instalar en MATLAB el *add-on* Communication Toolbox, que dispone de las funciones, algoritmos y objetos necesarios para el diseño, simulación y verificación de los elementos involucrados en sistemas de comunicaciones. Aquí encontramos tanto los métodos necesarios para, a partir de unos datos de entrada, realizar la modulación QAM de portadora única necesaria para la posterior modulación OFDM, así como las visualizaciones graficas propias, como las constelaciones de modulaciones, o el mapeo de las portadoras.

A modo de resumen, la aplicación desarrollada despliega por un lado los parámetros de entrada configurables para obtener la modulación OFDM, y por otro, con los valores seleccionados se muestra de forma gráfica la señal modulada en el eje temporal y en el espectro frecuencial.

Siempre se tratará la modulación desde el punto de vista de la transmisión, sin entrar en las modificaciones que introduce el canal de comunicaciones ni en la demodulación a realizar en el receptor de la señal.

Es posible, con las funciones que nos ofrece MATLAB, generar paso a paso una modulación como OFDM, desplegando una a una las etapas que implica, tal y como se especifica en en el diagrama de bloques de la figura [4]. Codificando, mapeando los datos de serie a paralelo, realizando la transformada inversa de Fourier, etc. Sin embargo, resulta más sencillo y claro utilizar las funciones y objetos que el módulo de comunicaciones de MATLAB proporciona.

En un primer lugar se ha investigado la función *ofdmmod* [12], que modula las subportadoras de datos de entrada en el dominio de la frecuencia utilizando el método de multiplexación por división de frecuencia ortogonal (OFDM) con un tamaño FFT y una longitud de prefijo cíclico especificados. Además, se pueden introducir en el algoritmo de modulación los índices de las subportadoras suprimidas y de las frecuencias piloto. Permite también indicar un factor de sobremuestreo.

También se ha analizado el objeto *comm.OFDMModulator* [13], que modula una señal en el dominio de la frecuencia utilizando el método de multiplexación por división de frecuencia ortogonal (OFDM), devolviendo una representación en banda base de la señal modulada. Encapsula alguna de las funcionalidades descritas anteriormente, y añadiendo algunas más:

- Número de puntos FFT. Numero de entrada para la transformada de Fourier inversa que se realiza en la modulación. Es equivalente al número de subportadoras, si no se realiza ningún tratamiento sobre ellas.
- Número de subportadoras en las bandas de guarda derecha e izquierda. Deben estar comprendidas cada una en el intervalo  $[0, (N_{FFT}/2)-1]$ , siendo  $N_{FFT}$  el número de puntos FFT especificado antes.
- Insertar portadora nula en el centro de la banda de frecuencia. Cuando se inserte, el índice central de la banda de frecuencias de la subportadora DC nula será  $(N_{FFT}/2)+1$  cuando  $N_{FFT}$  sea par y  $(N_{FFT}+1) / 2$  cuando  $N_{FFT}$  sea impar.
- Insertar portadoras piloto.
- Índices de las portadoras piloto.
- Longitud del prefijo cíclico. Se puede especificar una longitud para cada símbolo OFDM o aplicar la misma para todos.
- *Windowing*. Opción para aplicar enventanado de coseno alzado entre los símbolos OFDM.
- Longitud de *windowing*. Debe ser menor o igual a la longitud del prefijo cíclico.
- Factor de sobremuestreo.
- Número de símbolos OFDM.
- Número de antenas trasmisoras.

Este componente proporciona dos objetos con información y funcionalidad adicional, muy útil para su análisis:

- *info*. Proporciona información sobre el dimensionamiento del modulador: tamaño de los datos de entrada (*DataInputSize*), de las portadoras piloto (*PilotInputSize*) y de los datos modulados de salida resultantes (*OutputSize*).
- *showResourceMapping*. Muestra gráficamente el mapeo de portadoras de los símbolos OFDM.

El proceso de simulación de la señal modulada comienza por los datos de entrada. La entrada de datos para el modulador se creará en MATLAB, mediante la función *randi*, que devuelve un vector de números enteros pseudoaleatorios distribuidos de manera uniforme, cuyos valores varían entre 0 y el orden utilizado en la modulación de portadora única QAM, como se utiliza en los ejemplos de [14], [15] y [16], aunque podría ser cualquier otro valor máximo, como 1 para tener una secuencia binaria. Su longitud debe ser el resultado de la resta entre el número de portadoras especificado (64 por defecto) y las que se utilicen de guarda o como portadoras piloto. Este número puede ser cualquiera, siempre que cumpla esta condición. Se ha optado por fijar la creación de los valores de entrada bajo estos criterios, pensando que no es el objetivo principal de este trabajo, aunque se podría haber dejado la posibilidad al que use la aplicación de configurarlo también libremente.

A continuación, se realiza la modulación QAM de los datos de entrada obtenidos. Para ello, utilizaremos la función de MATLAB *qammod*, que devuelve la señal modulada con el orden de modulación especificado (se ha fijado en 16 para esta aplicación, aunque podría ser

cualquier otro factor, incluso modulaciones PSK por desplazamiento de fase de portadora única).

Finalmente, con la señal modulada mediante QAM y los parámetros que se han configurado en la zona de seleccionables de la aplicación por las personas que la utilizan se conforma la modulación OFDM.

Para poder simular de una forma amigable todas las funcionalidades que nos ofrece la modulación OFDM, con la capacidad de modificar los parámetros descritos anteriormente, se ha creado una aplicación con interfaz gráfica de usuario. En ella se observan dos zonas diferenciadas: en el panel izquierdo, bajo el título de “Parámetros OFDM” se disponen los elementos parametrizables de la modulación.

### Parámetros OFDM

Puntos FFT 64

#### Subportadoras

**Supresión de portadoras**  
 A la izquierda 6  
 A la derecha 5  
 Portadora central  
 Portadoras de datos 0

↗ Mapping portadoras

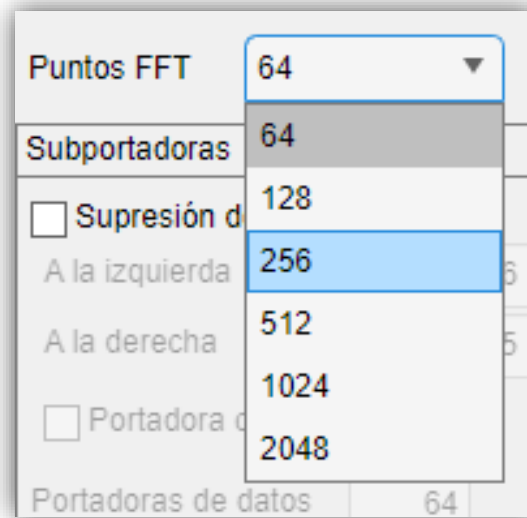
**Prefijo cíclico**  
 N° símbolos 4  
 Oversampling 
1
2
3
4
▾

**Windowing**

Aplicar

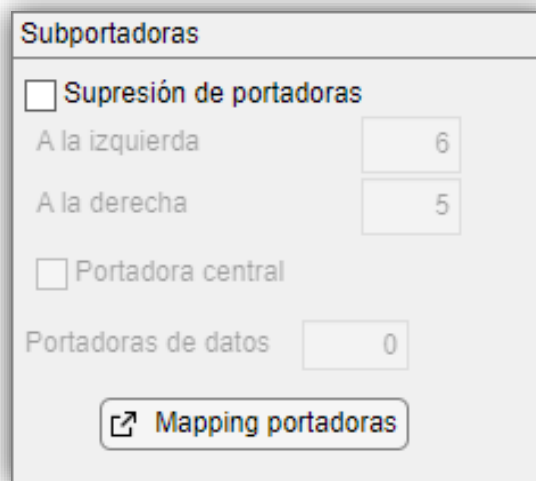
**Figura 7.** Panel “Parámetros OFDM”

En primer lugar, se debe elegir el número de puntos para la transformada inversa de Fourier a realizar en la modulación, es decir, los puntos FFT. Se ofrecen mediante un desplegable que dispone de valores desde 64 a 2048. El valor seleccionado por defecto es 64.



**Figura 8.** Selección de puntos FFT

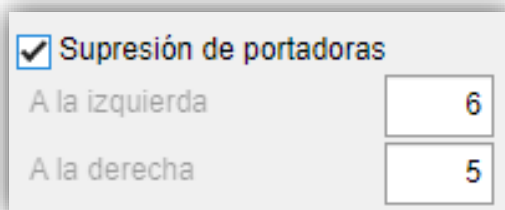
Se podrá gestionar, en función del valor seleccionado en este desplegable, las subportadoras que intervienen en la modulación. Inicialmente la opción de suprimir portadora no está marcada. Hasta que no cambie este estado, no se podrá gestionar ningún atributo más de este mini panel. Con la eliminación de estas portadoras se consigue un ancho espectral más reducido, que hace que se acoten las radiaciones fuera de banda y un sobremuestreo que mejora el efecto de *aliasing* o solapamiento frecuencial.



**Figura 9.** Panel de gestión de subportadoras

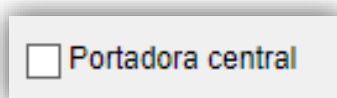
Si se marca la opción de suprimir portadoras, se habilitan primero los campos de entrada para indicar el número de subportadoras que se eliminarán simétricamente de las

frecuencias exteriores de la izquierda y derecha de la banda. En ellos se indican unos valores por defecto, obtenidos de las modulaciones más utilizadas, como son 6 a la izquierda y 5 a la derecha para el número de puntos de FFT de 64 que se utiliza en el protocolo WiFi 802.11 a,g, o los 32 a la izquierda y 31 a la derecha, usado en el modo III (256 puntos FFT) en el sistema de trasmisión DAB [2]. Las personas que hacen uso de la aplicación podrán especificar libremente cualquier otro valor numérico.



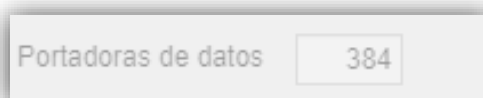
**Figura 10.** Supresión de portadoras laterales

A continuación, se permite la opción de eliminar también la frecuencia portadora central de la banda, de forma que para equipos con problemas de componente continua la componente continua de todas las señales paso banda será nula.



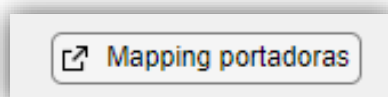
**Figura 11.** Selección de eliminar portadora central

Otro campo de texto indica que numero de portadoras serán las encargadas de transportar información efectiva, ya descontadas las portadoras nulas y la central en el caso de que así se haya seleccionado.



**Figura 12.** Número de portadoras de datos

Por último, este subpanel de portadoras dispone del botón “*Mapping portadoras*”.

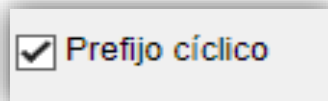


**Figura 13.** Botón *Mapping portadoras*



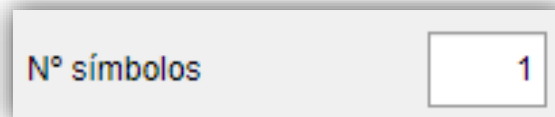
Al pulsarlo se abre una nueva ventana mostrando gráficamente en la disposición de los índices de subportadoras por símbolo.

Se podrá habilitar la casilla de verificación “Prefijo cíclico”, para indicar su inclusión en el tiempo de guarda de cada símbolo OFDM.



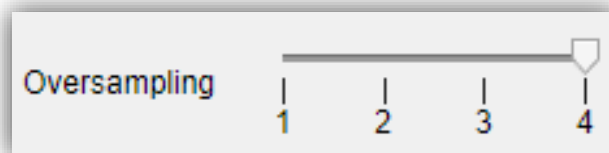
**Figura 14.** Selección de prefijo cíclico

El número de símbolos OFDM a utilizar en la modulación, cuyo valor inicial será 4.



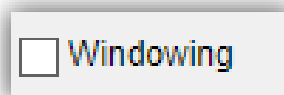
**Figura 15.** Número de símbolos OFDM

La selección a través de un campo deslizador *Oversampling* permite seleccionar un valor de sobremuestreo.



**Figura 16.** *Oversampling*

Como último parámetro configurable, se presenta el seleccionable *Windowing* mediante el que se puede aplicar un enventanado de coseno alzado entre los símbolos OFDM.



**Figura 17.** *Windowing*

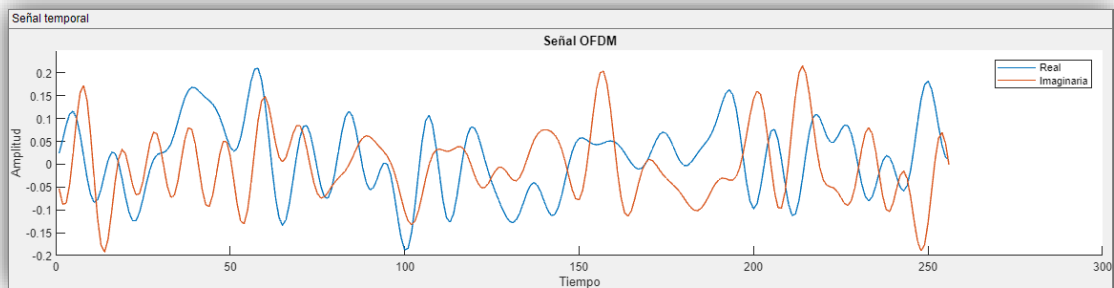
Finalmente, tenemos el botón “Aplicar” para consolidar los datos de configuración seleccionados anteriormente, realizar la modulación OFDM y generar nuevas graficas mostrando los resultados.



**Figura 18.** Botón Apicar

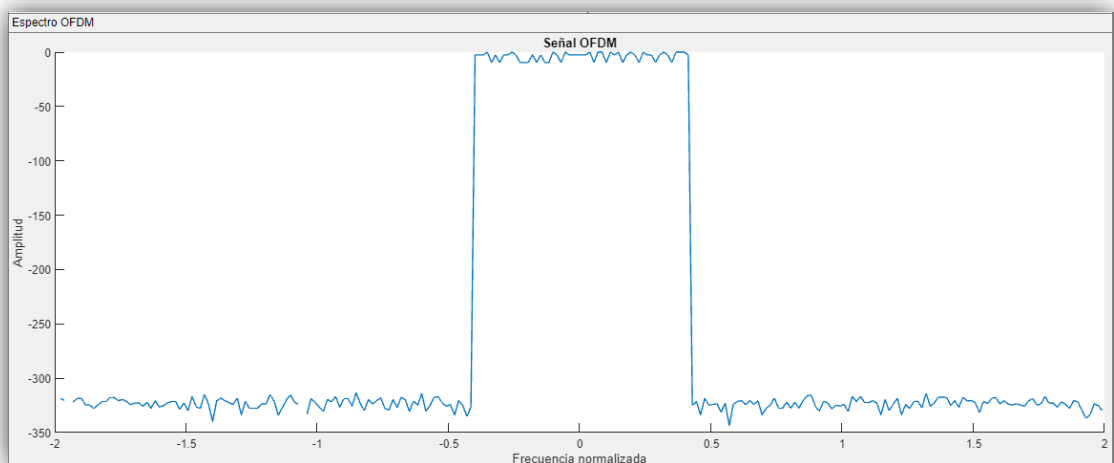
Para obtener la representación gráfica, se divide la sección derecha de la aplicación en dos paneles:

- Señal temporal. Muestra las gráficas de la parte real (en color azul) e imaginaria (color rojo) de los datos modulados OFDM en función del tiempo.



**Figura 19.** Panel de señal temporal

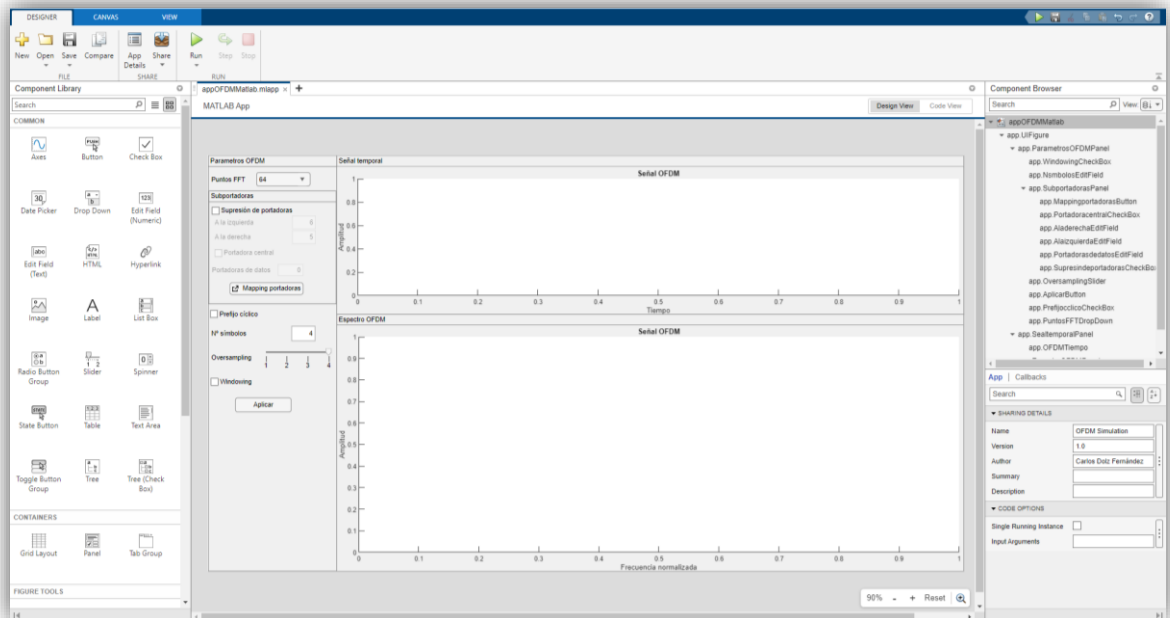
- Espectro OFDM. Se ve la representación sobre un eje frecuencial normalizado del valor absoluto de la transformada de Fourier de los datos modulados mediante OFDM en escala logarítmica.



**Figura 20.** Panel de espectro frecuencial

En el código de MATLAB generado para la simulación de la modulación OFDM tendremos por un lado todo los objetos y funciones relacionadas con la creación y manejo de los componentes visuales mostrados anteriormente para la gestión de los parámetros de configuración de la señal modulada, así como su posterior representación gráfica.

Se ha editado la aplicación utilizando la aplicación *App Designer* de MATLAB. Así, se genera automáticamente todo el código de los componentes visuales que se añadan. Para ello dispones de una vista de diseño, con todos los componentes disponibles y un área donde se pueden añadir, y una vista de código, que muestra el código MATLAB que se genera automáticamente desde la vista diseño, más el código que el personal encargado de la programación quiera añadir.



**Figura 21.** Vista de diseño en MATLAB *App Designer*

Para ejecutar la modulación OFDM en sí, se ha utilizado el objeto *comm.OFDMModulator* analizado anteriormente.

```
% OFDMModulator instance;
ofdmMod = comm.OFDMModulator;
```

En una única función, llamada OFDM se ejecuta todo el código parametrizando el objeto de modulación con los valores obtenidos de los componentes visuales que puede manejar el colectivo usuario de la aplicación.

Así, la primera operación será liberar el objeto *ofdmMod* instanciado, para que de esta forma acepte los posibles cambios que en cada ejecución se produzcan.

```
function OFDM(app)

    % Release resources and allow changes
    release(app.ofdmMod);
```

A continuación, se obtiene el número de puntos FFT seleccionado en el desplegable que lo gestiona y se asigna al objeto modulador tras convertirlo a un número de tipo *double*.

```
% FFTLength
app.ofdmMod.FFTLength = str2double(app.PuntosFFTDropDown.Value);
```

El siguiente proceso es el tratamiento de los parámetros de configuración referentes a las subportadoras. Se controla si está o no habilitado la casilla de verificación “Supresión de portadoras” y, si está habilitada, se ejecutan los siguientes pasos:

- La función llamada “portadorasSuprimidas”, mediante la cual se asignan valores a los campos de texto de portadoras suprimidas a izquierda y derecha en función del número de puntos FFT seleccionado. Para el caso concreto de 64 puntos, se aplican los valores que se especifican en el estándar WiFi 802.11 a.g, que son 6 y 5 respectivamente, y para el resto de valores, se obtiene dividiendo el número de puntos entre 8, con una unidad menos en el caso de la derecha. Si la casilla de supresión de portadoras no está habilitada, los valores de ambos campos de texto serán cero.

```
function portadorasSuprimidas(app, fftLength)
value2= app.SupresindeportadorasCheckBox.Value;
if (value2==1)

    if fftLength == 64
        app.AlaizquierdaEditField.Value = 6;
        app.AladerechaEditField.Value = 5;
    else
        app.AlaizquierdaEditField.Value = fftLength/8;
        app.AladerechaEditField.Value = fftLength/8-1;
    end
else
    app.AlaizquierdaEditField.Value = 0;
    app.AladerechaEditField.Value = 0;
end
```

- Después se validará si se ha chequeado la opción “Portadora central”, activando en ese caso la opción *InsertDCNull* del objeto *comm.OFDMModulator*, y desactivándola en el contrario.

```

if (app.PortadoracentralCheckBox.Value == 1)
    app.ofdmMod.InsertDCNull = true;
else
    app.ofdmMod.InsertDCNull = false;
end

```

- Se actualizan los valores de subportadoras suprimidas a la izquierda y la derecha de la banda de frecuencias con los valores de los campos de texto destinados a tal efecto, actualizando el valor *NumGuardBandCarriers* del objeto modulador *ofdmMod*.

```

leftSC = app.AlaizquierdaEditField.Value;
rightSC = app.AladerechaEditField.Value;
app.ofdmMod.NumGuardBandCarriers = [leftSC;rightSC];

```

- Por último, se actualizará el campo de portadoras de información efectiva en base a los parámetros anteriores.

```

% Number of data subcarriers
app.PortadorasdedatosEditField.Value = app.ofdmMod.FFTLength-leftSC-
rightSC-app.PortadoracentralCheckBox.Value;

```

Si no se ha marcado la opción de suprimir subportadoras, el valor del número de portadoras de datos será igual al de puntos FFT.

```

else
    % Number of data subcarriers
    app.PortadorasdedatosEditField.Value = app.ofdmMod.FFTLength;

```

Si el campo de validación de prefijo cíclico está activado, se indica al objeto modulador la longitud de este, que en nuestro caso será siempre el valor 16.

```

% CyclicPrefixLength
cyclicPrefix = app.PrefijocclicoCheckBox.Value;
if (cyclicPrefix == 1)
    app.ofdmMod.CyclicPrefixLength = 16;
else
    app.ofdmMod.CyclicPrefixLength = 0;
end

```

El valor del campo de texto de número de símbolos servirá tal cual para informar el mismo parámetro del objeto OFDM.

```
% Num Symbols
app.ofdmMod.NumSymbols = app.NsmbolosEditField.Value;
```

Se obtiene el valor de sobremuestreo del deslizador “*Oversampling*”, se redondea y se asigna a la propiedad *OversamplingFactor*.

```
% Oversampling
app.ofdmMod.OversamplingFactor=round(app.OversamplingSlider.Value, 0);
```

Como último parámetro a tratar, si se ha seleccionado la casilla de verificación “*Windowing*”, se activa la propiedad *Windowing* y se le asigna la misma longitud que tiene el prefijo cíclico.

```
% Windowing
windowing = app.WindowingCheckBox.Value;
if (windowing == 1 && cyclicPrefix == 1)
    app.ofdmMod.Windowing = true;
    app.ofdmMod.WindowLength = app.ofdmMod.CyclicPrefixLength;
else
    app.ofdmMod.Windowing = false;
end
```

Con todas las propiedades que se han alimentado en el objeto modulador, ya se puede gestionar para obtener datos de él. Primero vamos a analizar su función *info*, de la que se extraerán los tamaños de los campos de entrada de datos y de salida de la señal ya modulada.

```
modDim = info(app.ofdmMod);
```

Se utilizará el valor de tamaño de campo de entrada para su generación aleatoria. Como ya se ha comentado, utilizamos un tamaño máximo de 16 fijo como posibles valores. Este será también el orden fijado en la modulación QAM de portadora única de los datos de entrada así creados. Obtendremos dos estructuras de datos: *dataIn*, que sirve de entrada para obtener la segunda, *dataQAM*.

```
M = 16;

%Random data In
dataIn = randi([0 M-1],modDim.DataInputSize);

dataQAM = qammod(dataIn,M,UnitAveragePower=true);
```

A partir de la estructura *dataQAM* se invocará a la modulación del objeto *ofdmMod*, obteniendo como resultado una estructura de datos modulados mediante OFDM.

```
sOFDM = app.ofdmMod(dataQAM);
```

Estos datos son los que se representarán gráficamente, respecto al eje temporal y al frecuencial.

Primero, la representación temporal se hace de la parte real e imaginaria de la señal obtenida.

```
%Time
plot(app.OFDMTiempo,1:modDim.OutputSize, [real(sOFDM) imag(sOFDM)]);
legend(app.OFDMTiempo, "Real", "Imaginaria");
```

Para la representación del espectro frecuencial de la señal OFDM se obtiene por un lado el valor absoluto, en escala logarítmica, de la transformada de Fourier de la señal. Hay que recordar que al hacer la transformada de Fourier de una señal se obtiene su información en el dominio de la frecuencia. El eje horizontal del gráfico es la frecuencia normalizada, para lo que obtiene un rango desde  $-(N-1)/2$  hasta  $(N-1)/2$  dividido entre  $N$ , siendo  $N$  el tamaño de los datos modulados. Si se ha seleccionado *Oversampling*, se multiplica por el factor elegido para realizar el sobremuestreo de la señal.

```
% Frequency
N=length(sOFDM);
Spectrum1=20*log10(abs(fft(sOFDM,N)));
% Eje frecuencial en Hz.
EjeFreq=(-(N-1)/2:(N-1)/2)/N*app.ofdmMod.OversamplingFactor;
plot(app.OFDMFrecuencia,EjeFreq,fftshift(Spectrum1)-max(Spectrum1));
```

En el anexo final de esta memoria se incluye todo el código de la aplicación, la parte generada automáticamente desde la vista de diseño y las funciones codificadas para realizar la simulación de la modulación a partir de los parámetros configurables.



### 3. Resultados

Utilizando la metodología descrita en el apartado anterior, la realización de este trabajo resulta en una aplicación de interfaz visual para la simulación de distinto aspecto gráficos de una modulación OFDM mediante la asignación de valores a los distintos parámetros establecidos.

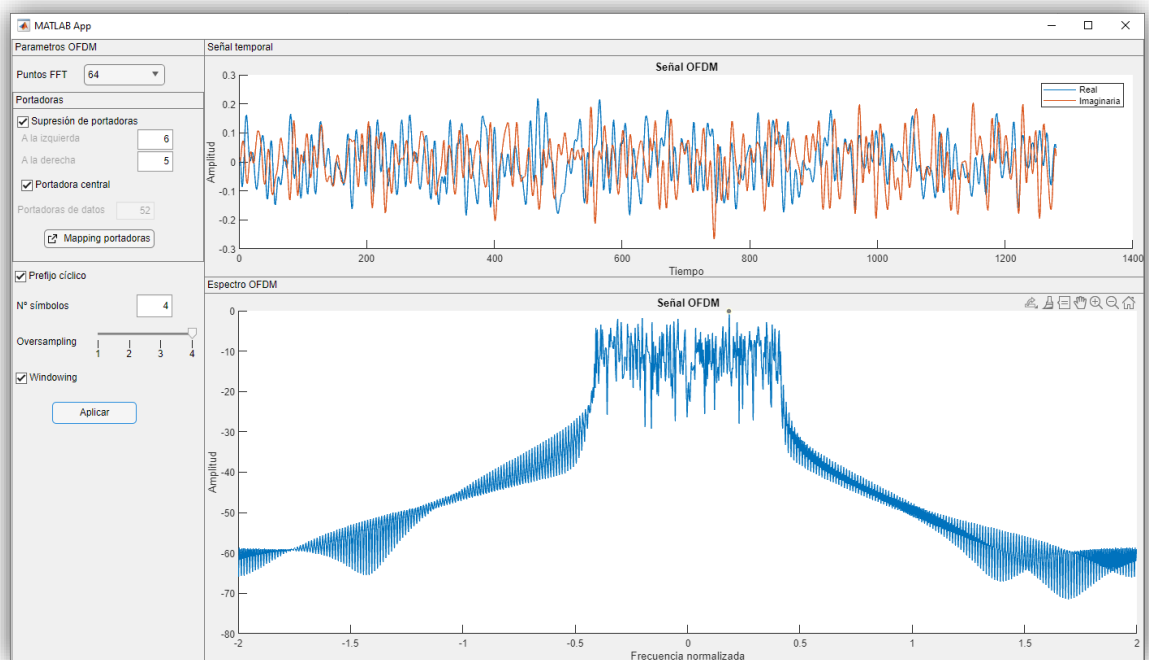


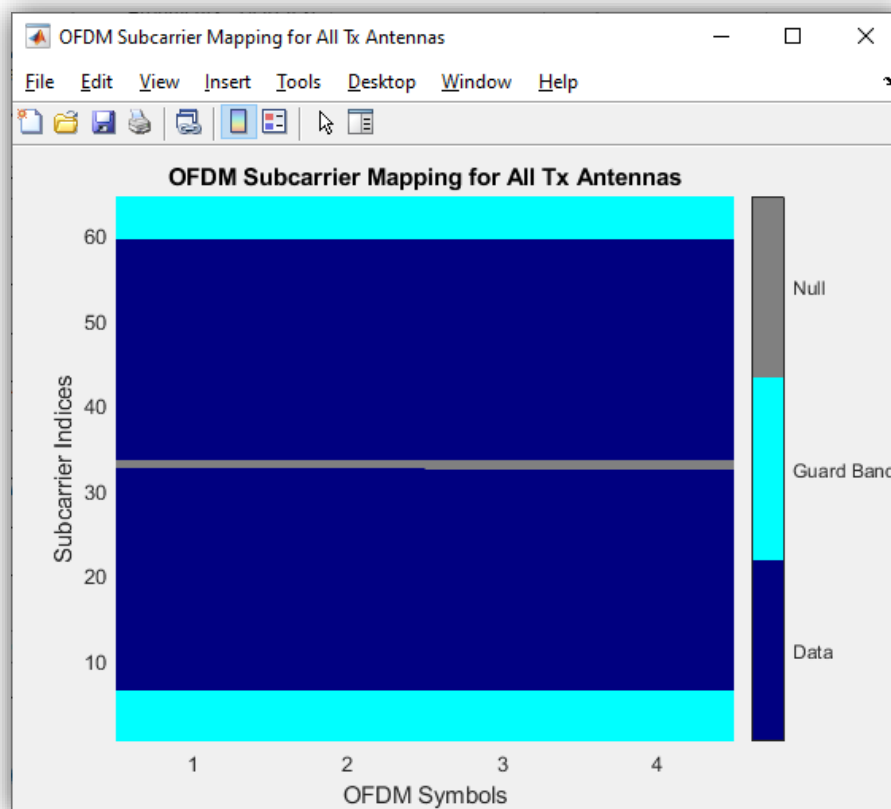
Figura 22. Aplicación de simulación de señal OFDM

#### 3.1. Descripción de los resultados

En los resultados obtenidos mediante la aplicación desarrollada no se trata tanto de obtener la señal modulada para transmisión que mejor se adecue a la máscara espectral con los parámetros especificados según definen los estándares basados en OFDM [17], sino de tener visualmente una representación de como estos afectan a la conformación de la señal. Se observa cómo varía tanto temporal como espectralmente la señal modulada mediante OFDM al modificar individualmente cada uno de los parámetros que se han establecido. Realmente es en la presentación espectral donde se aprecian los efectos de la supresión de subportadoras y la aplicación del prefijo cíclico o la conformación de pulso mediante *windowing*.

Se puede comprobar empíricamente como afectan los distintos parámetros que ofrece la aplicación en la forma de onda tanto temporal como espectral de una señal modulada mediante OFDM. Se ofrecen ambas gráficas, aunque se destaca en mayor tamaño la representación espectral sobre la frecuencia normalizada de la señal modulada, ya que se considera más importante en el estudio de esta modulación.

Además, se ofrece la distribución gráfica de los índices de las portadoras en los símbolos OFDM mediante la opción de “*Mapping portadoras*”. Esto nos muestra la posición de las subportadoras para cada símbolo OFDM, según se hayan eliminado las frecuencias de guarda o la central, indicando mediante un código de colores las portadoras de datos (*Data*), las suprimidas en las bandas de guarda laterales (*Guard Band*) y en frecuencia central (*Null*) e incluso posibles portadoras piloto, fuera del alcance de este trabajo. De esta forma se visualiza la asignación de recursos de tiempo-frecuencia para una modulador OFDM. Los índices de las subportadoras utilizados como frecuencias suprimidas o frecuencias pilotos pueden ser diferentes para cada símbolo.



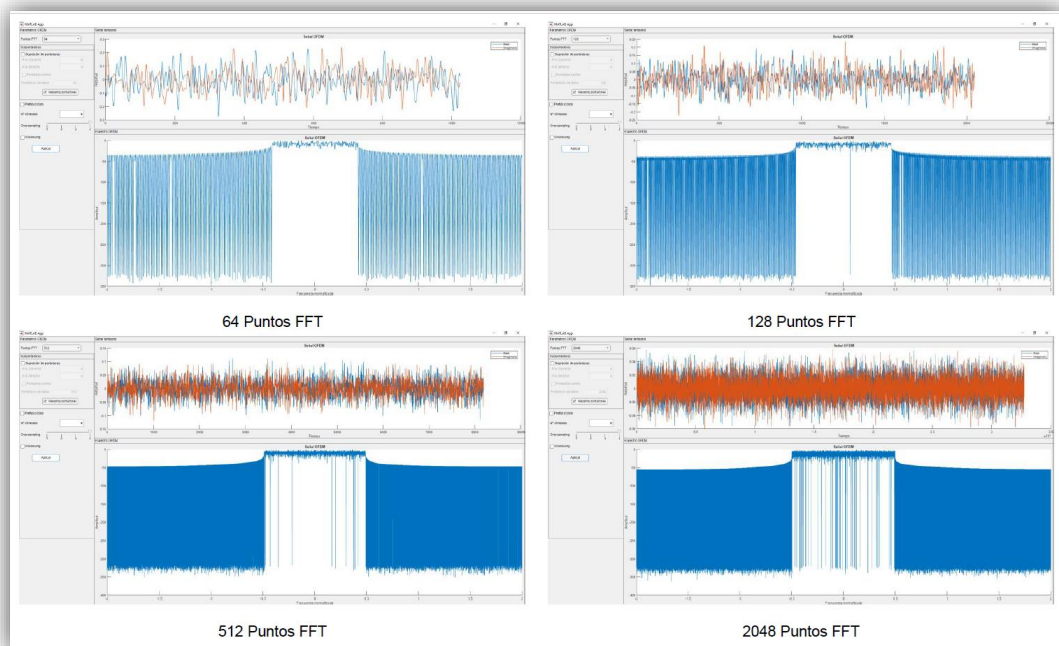
**Figura 23.** Ventana *Mapping portadoras*

### 3.2. Resultados según la parametrización de valores

A continuación, se analizan los resultados obtenidos al interactuar con los parámetros de entrada disponibles en la aplicación, que conformaran así la señal modulada de diferente forma. Cada uno de estos campos de entrada afectan a una o varias características distintas de la señal, siendo algunos más visibles en su forma temporal y otros en el espectro de frecuencias y otros en ambas.

El primer parámetro con el que se puede interactuar es el número FFT, que típicamente viene dado por la menor potencia de 2 que sea mayor que el número de subportadoras deseado. Se debe elegir un número FFT que sea significativamente mayor que el número de subportadoras, para asegurar que los efectos de borde son despreciables a la mitad de la frecuencia de muestreo y para garantizar que la forma del filtro de reconstrucción del DAC (convertidor digital a analógico) no afecte a una parte significativa del espectro.

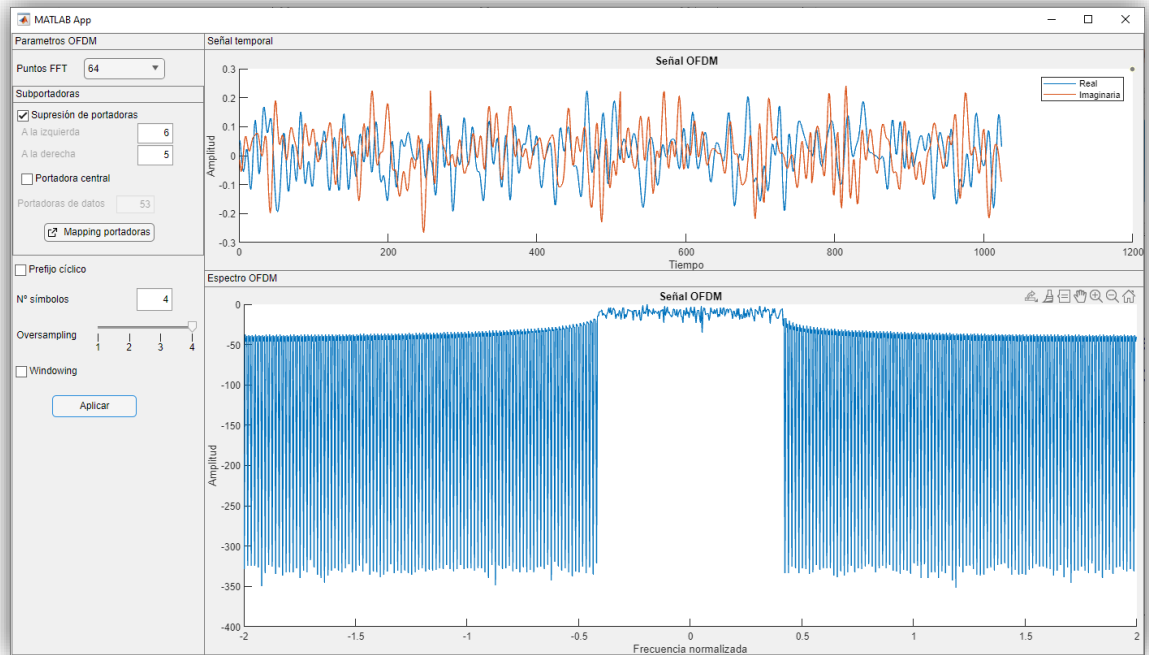
El número de puntos FFT es un parámetro de implementación para el transmisor y el receptor de la señal OFDM.



**Figura 24.** Señal generada con distintos números de puntos FFT

Por ejemplo, en el estándar de difusión de televisión digital terrestre DVB-T hay dos modos de difusión en función del número de puntos FFT que se elija, modo 2k o modo 8k. En el modo 2k el número de puntos FFT es de 2.048 (de ahí el nombre del modo), con 1.074 portadoras de datos no nulas, mientras que en el modo 8k el tamaño de la FFT es de 8.192, trabajando con 6.816 portadoras no nulas. En la segunda generación del estándar (DVB-T2) se llegan a tener tamaños FFT (modos) de 1k, 2k, 4k, 8k, 16k y 32k.

Mediante las portadoras suprimidas se consigue un ancho espectral más reducido, que favorece que las radiaciones fuera de banda estén acotadas, y un efecto de sobremuestreo sobre las señales digitales que mejora el efecto de *aliasing* o solapamiento frecuencial.

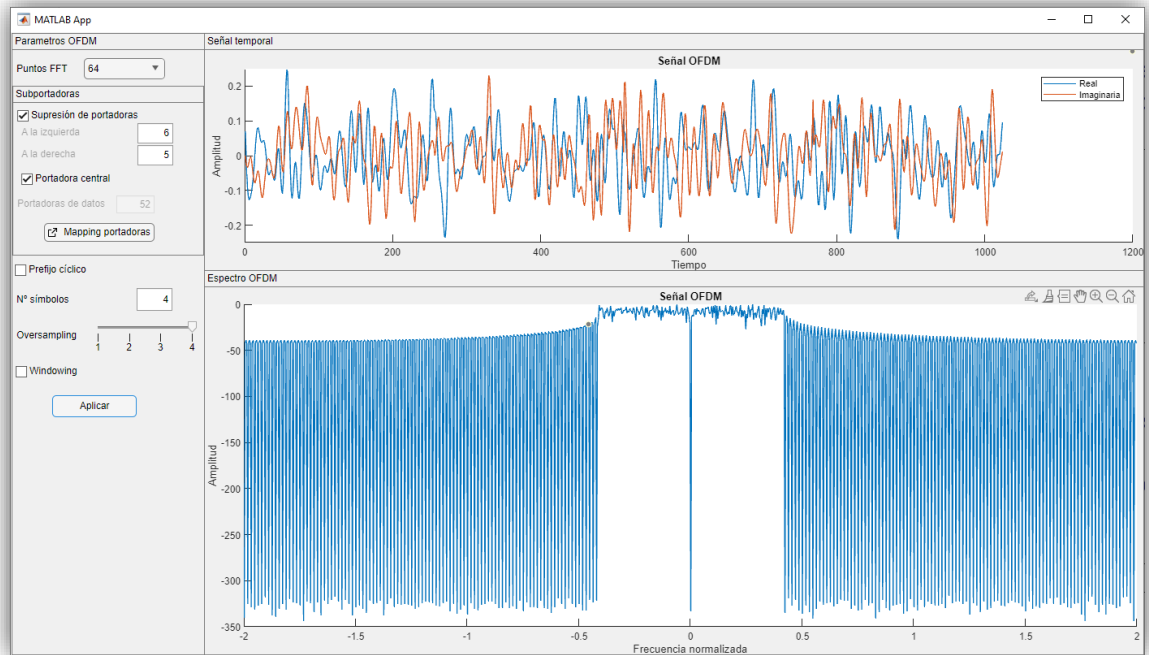


**Figura 25.** Aplicación de supresión de subportadoras

La eliminación de portadoras laterales se debe realizar cuando conozcamos el número de portadoras de datos efectivo. La diferencia entre el numero FFT y este número de portadoras de datos deben ser portadoras nulas, suprimiéndolas de las bandas laterales del espectro frecuencial.

Esta supresión de portadoras se efectuará también al implementar *oversampling*, como veremos posteriormente.

Con la eliminación de la frecuencia central, la componente continua de todas las señales paso bajo es siempre nula, lo que es beneficioso cuando nos podemos encontrar con problemas de componente continua en los equipos.



**Figura 26.** Aplicación de supresión de subportadora central

La subportadora central, que corresponde con la componente continua en la señal compleja en banda base no se modula debido a la implementación del receptor en la práctica, como en el estándar europeo de radiodifusión de audio digital (sistema DAB).

El estándar IEE 802.11a específicamente indica que para evitar problemas en los convertidores DAC y ADC (digital/analógico y analógico/digital) no se usa la subportadora central para transmisión [18].

Al incluir un prefijo cíclico en el tiempo de guarda entre símbolos consecutivos consigue la disminución del nivel de los lóbulos laterales respecto al principal en el dominio de la frecuencia. Además, sirve para combatir la interferencia intersimbólica ISI que puede aparecer en cuando hay un canal con respuesta impulsional no ideal.

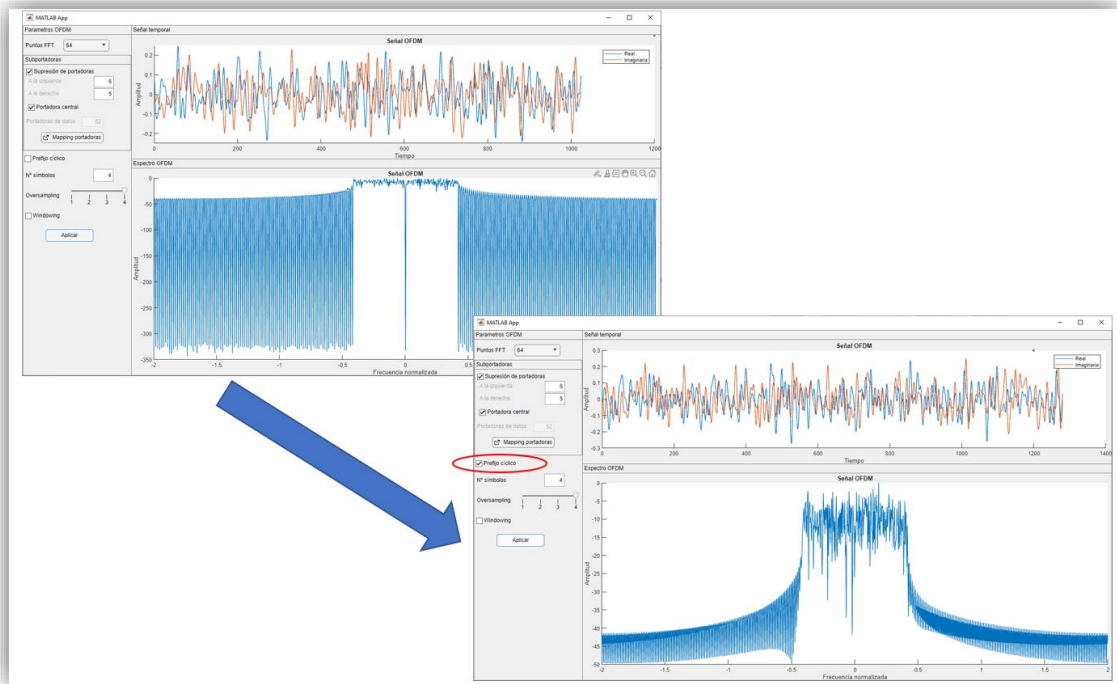


Figura 27. Inclusión de prefijo cíclico

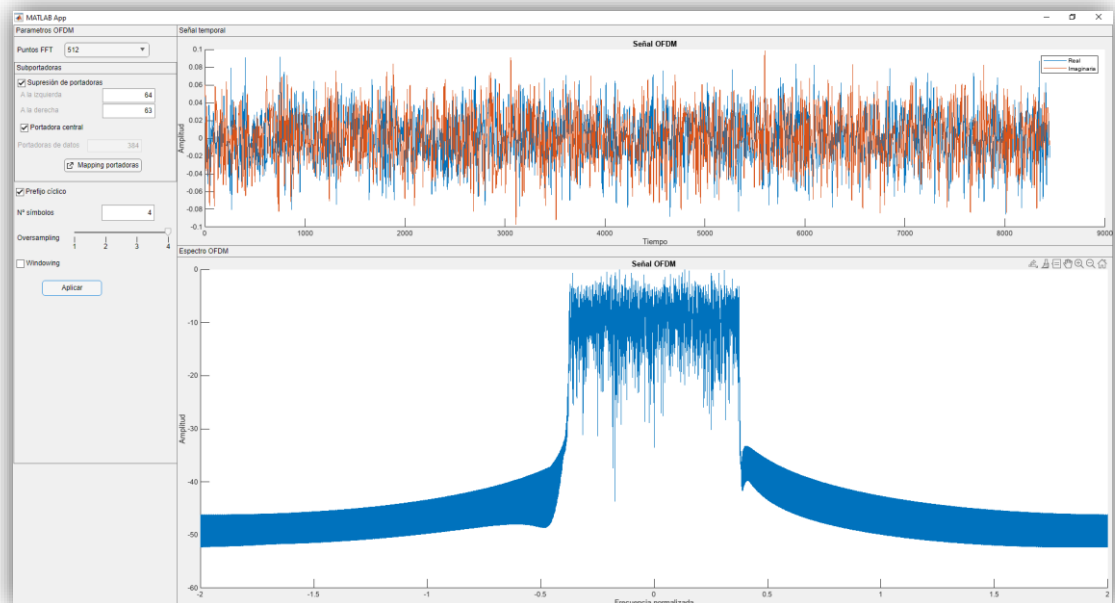
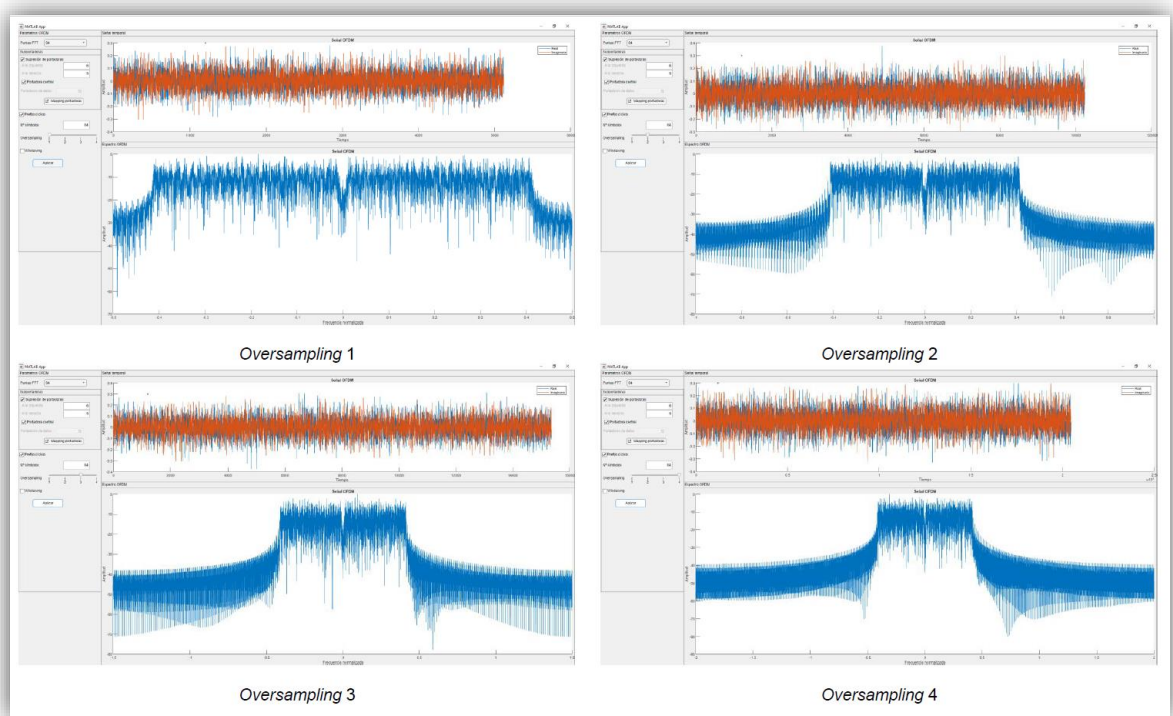


Figura 28. Señal con prefijo cíclico

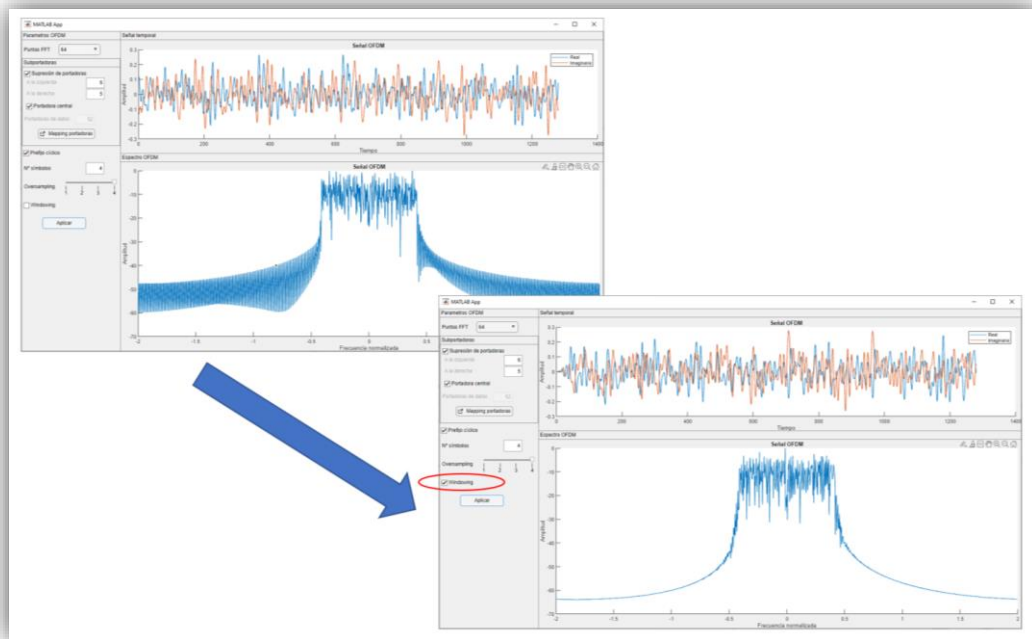


Como se muestra en la figura [4] la señal modulada OFDM se consigue realizando la transformada de Fourier inversa y después se procesa mediante un convertidor digital/análogo (DAC). Si se utiliza el *oversampling*, se consigue simplificar el filtro pasa bajo necesario en recepción que separa los símbolos al reducirse el ancho espectral entre ellos. Se puede implementar usando un número FFT mayor rellenado a ceros las posiciones de las portadoras no utilizadas, o mediante un circuito específico para ello, como se hace en los sistemas comerciales reales.



**Figura 29.** Señal con diferentes valores de *oversampling*

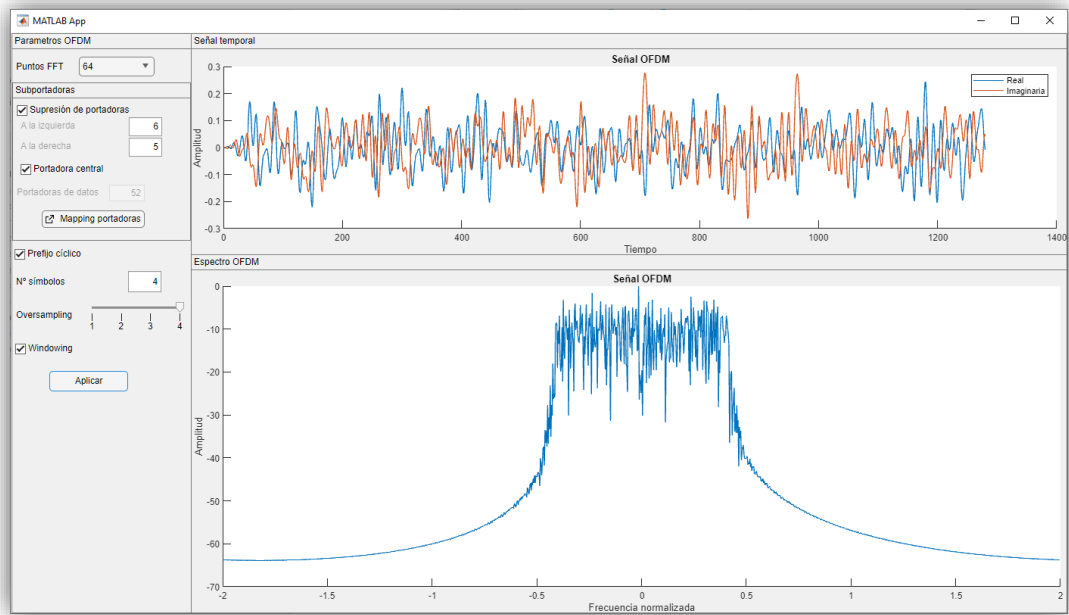
Al aplicar *windowing*, se realiza una conformación de pulso de la señal, solapando temporalmente los símbolos OFDM consecutivos durante un porcentaje del tiempo de guarda, como se explica en [19] y [20]. Mediante esta técnica se intenta mantener el espectro acotado, mejorando las radiaciones fuera de banda, aunque empeoren las condiciones respecto al *delay spreads* o dispersión del retardo, que mide de media la duración temporal de la respuesta impulsional del canal.



**Figura 30.** Aplicación de *windowing*

En la simulación se aplica un enventanado de coseno alzado, con el que se ajusta la forma espectral de la señal suavizando la forma del pulso rectangular de la subportadora. Para ello se toma una ventana de ajuste con la longitud del prefijo cíclico y entre símbolo y símbolo, la señal queda conformada, con lo que se mejora el factor espectral a expensas de la inmunidad al desvanecimiento multitrayecto, debido a que la redundancia en la banda de guarda se reduce porque sus valores de muestra se ven comprometidos por el suavizado.





**Figura 31.** Señal OFDM con *windowing*

## 4. Conclusiones

A modo de conclusión de este trabajo cabe resaltar que se ha conseguido desplegar una aplicación que permita mediante la selección de algunos parámetros, como son el número de puntos FFT, la supresión de subportadoras laterales y central, la inclusión de un prefijo cíclico, el factor de sobremuestreo y aplicación de *windowing*, configurar una señal modulada mediante OFDM, mostrando gráficamente la señal obtenida tanto en el dominio del tiempo como en el de la frecuencia. De esta forma es posible observar cómo varía la señal al modificar los valores de cualquiera de estos parámetros, obtenido así una herramienta que ofrece una representación gráfica simulada para el estudio y comprensión de este tipo de modulación adicional al mero marco teórico.

Se puede observar en los resultados de la memoria como la implementación de la aplicación se centra en la obtención de la visualización gráfica de una señal modulada mediante OFDM que, según van variando los valores de los campos que se han establecido como entradas del sistema, se va moldeando en distintos aspectos, reforzando así la comprensión de estos atributos, propios de este tipo de modulación.

En los objetivos iniciales del proyecto se especificaba el estudio teórico de la modulación OFDM, el diseño e implementación de una aplicación que la simule gráficamente, con parámetros de entrada configurables, así como el análisis de los resultados obtenidos para diferentes casos. Dentro del desarrollo del trabajo se han alcanzado todos estos objetivos.

La planificación establecida al principio del trabajo ha sido seguida con éxito, llegándose a cumplir los hitos establecidos en ella con suficiente puntualidad y completitud. Respecto a la metodología de estudio que se indicaba, esta ha sido lo suficientemente adecuada para llegar a los resultados previstos, sin tener necesidad de realizar cambios en ella, más allá de las líneas de trabajo que han quedado pendientes.

## 5. Trabajos futuros

En las líneas de trabajo futuro que han quedado fuera del trabajo finalmente se pueden incluir los siguientes aspectos:

- Incluir la posibilidad de elección de modulaciones de portadora única QAM de distintos ordenes (4, 32, ...), e incluso PSK (BPSK, QPSK, ...). Esta intervención tan solo cambiaría internamente el proceso de formación de la señal modulada mediante OFDM, por lo que se ha dejado fuera del presente trabajo.
- Uso de símbolos piloto, de valores previamente acordados y estandarizados que no transportan bits de información, sino que transportan símbolos conocidos en recepción y que se utilizan para funcionalidades típicas de sincronismo y de demodulación. Se pueden utilizar los símbolos piloto para que en recepción se pueda estimar la función de transferencia del canal que se utiliza en la ecualización, por lo que la parrilla de colocación de frecuencias piloto debe ser diseñada atendiendo a la variabilidad temporal del canal. Existen en sistemas comerciales como los mencionados en otros apartados de esta memoria posiciones y características definidas de estos símbolos piloto, pero al tratarse de una aplicación lo más abierta posible, la libertad de elección de otros atributos implicaría que o bien se conocen muy profundamente la combinación de estos valores o no tiene sentido mezclarlos.
- Análisis de probabilidad de error BER. Con los parámetros seleccionados que generan la señal OFDM, en un sistema de comunicación a través de un canal AWGN en recepción se puede obtener el BER en función de diferentes SNR del canal. Ya se ha visto que este tipo de simulación en concreto se ha desarrollado por otros autores [8].
- Análisis de PAPR (Peak-to-Average Power Ratio) Se trata del principal problema de las señales moduladas con OFDM. Su valor elevado las hace susceptibles a distorsiones no lineales, provoca radiación fuera de banda, afectando a los canales adyacentes y reduce la radiación dentro de la banda, incrementando la BER.
- Análisis del tiempo de símbolo OFDM, tiempo útil de transmisión de símbolo, tiempo de guarda, velocidad de bit, tiempo de retardo, ... Se puede establecer el número de puntos FFT (y las subportadoras en consecuencia) a partir del ancho de banda del canal, la velocidad de los datos y la duración de símbolo útil.

## 6. Glosario

**OFDM.** Multiplexación por división en frecuencias ortogonales (*Orthogonal Frequency Division Multiplexing*).

**MATLAB.** Plataforma de programación y cálculo numérico para analizar datos, desarrollar algoritmos y crear modelos.

**QAM.** Modulación de amplitud en cuadratura.

**FFT / IFFT.** Transformada rápida de Fourier / Transformada inversa de Fourier.

**ISI.** Interferencia intersimbólica.

**ICI.** Interferencia entre portadoras.

**COFDM.** Coded OFDM.

**DAB.** Transmisión digital de audio (Digital Audio Broadcasting).

**DVB-T.** Difusión de Video Digital – Terrestre (Digital Video Broadcasting – Terrestrial).

**Wifi IEEE 802.11a,g.** Protocolo de red de área local, también conocido como Wireless LAN

**Wimax 802.16.** Tecnología inalámbrica de red de banda ancha bajo el estándar IEEE 802.16.

**LTE.** Estándar para comunicaciones inalámbricas de transmisión de datos de alta velocidad para teléfonos móviles y terminales de datos

**Simulink.** Entorno en MATLAB de diagramas de bloque que se utiliza para diseñar sistemas con modelos multidominio, simular antes de implementar en hardware y desplegar sin necesidad de escribir código.

**BER.** Tasa de error binaria (*Bit Error Rate*) de una señal digital.

**SNR.** Relación señal/ruido (Signal to Noise Ratio).

**AWGN.** ruido gaussiano blanco aditivo (Additive White Gaussian Noise).

**DFT.** Transformada discreta de Fourier.

**DPQPSK.** Modulación por desplazamiento de fase en cuadratura de polarización dual (Dual-Polarization Quadrature Phase Shift Keying).

**Add-on.** Complemento que amplía las capacidades de MATLAB proporcionando otras funcionalidades para tareas y aplicaciones específicas.

**PSK, BPSK, QPSK.** Modulación por desplazamiento de fase (Phase Shift Keying), y sus tipologías binaria (Binary Phase-Shift Keying) y cuaternaria (Quadrature Phase-Shift Keying).

**App Designer.** Entorno integrado en MATLAB para diseñar la interfaz de usuario y escribir el código que determina el comportamiento de una aplicación.

**DAC y ADC.** Convertidor digital/analógico y analógico/digital.

**PAPR.** Relación de potencia de pico a promedio (Peak-to-Average-Power Ratio).

## 7. Bibliografía

- [1] *Objetivos y metas de desarrollo sostenible – Desarrollo Sostenible* [en línea] [consulta: 4 de marzo de 2023]. Disponible en: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [2] CABRERA Margarita y TARRÈS Francesc. *Multiplexación por división en frecuencias ortogonales (OFDM)* [en línea] Barcelona: UOC [consulta: 2 de marzo de 2023]. Disponible en: <http://cvapp.uoc.edu/autors/MostraPDFMaterialAction.do?id=214617>
- [3] SCHULZE, H. y LUDERS, C. *Theory and applications of OFDM and CDMA. Wideband Wireless Communications*. Chichester (GB): Wiley, 2005. ISBN: 978-0-470-01739-5
- [4] AMSTRONG, J. OFDM for Optical Communications. En: *Journal of Lightwave Technology* [en línea]. 2009, vol. 27, no. 3, pp. 189-204. ISSN 1558-2213. [consulta: 16 de marzo de 2023]. Disponible en: <https://ieeexplore.ieee.org/abstract/document/4785281>
- [5] *¿Qué es OFDM? - MATLAB & Simulink* [en línea] [consulta: 2 de marzo de 2023] Disponible en: <https://es.mathworks.com/discovery/ofdm.html>
- [6] *Modulate using OFDM method - Simulink - MathWorks España* [en línea] [consulta: 15 de marzo de 2023] Disponible en: <https://es.mathworks.com/help/comm/ref/ofdmmodulatorbaseband.html>
- [7] GHORPADE, S. S. y SANKPAL, S. V. Behaviour of OFDM System using MATLAB Simulation. En: *International Journal of Advanced Computer Research* [en línea]. 2013, Tomo 3, Nº 2. 67-71. ISSN 2249-7277 [consulta: 13 de marzo de 2023]. Disponible en: <https://www.proquest.com/scholarly-journals/behaviour-ofdm-system-using-matlab-simulation/docview/1444152693/se-2>
- [8] SHA, X. X., ZHOU, J., y SONG, Y. X. Design and simulation of OFDM system based on Matlab. En: *Applied Mechanics and Materials* [en línea]. 2014, Tomo 635-637, 1081-1085. <https://www.proquest.com/scholarly-journals/design-simulation-ofdm-system-based-on-matlab/docview/1563384826/se-2?accountid=15299>
- [9] BAO, Xiurong, *Matlab simulation and performance analysis of OFDM system*. [en línea] Conferencia: International Conference on Computer Science and Information Processing (CSIP), Xi'an, China, 2012, pp. 1423-1426. [consulta: 16 de marzo de 2023]. Disponible en: <https://ieeexplore.ieee.org/abstract/document/6309131>

- [10] CHO, Yong Soo, KIM, Jaekwon, YANG, Won Young y KANG, Chung G. Introduction to OFDM. En: *MIMO-OFDM Wireless Communications with MATLAB®*, [en línea]. IEEE, 2010, pp.111-151. ISBN 9780470825631. [consulta: 16 de marzo de 2023]. Disponible en: <https://ieeexplore.ieee.org/document/5676090>
- [11] SACIE ALCÁZAR, D., 2015. Simulación de sistemas CO-OFDM 4D ópticos [en línea]. Trabajo final de Máster: UOC, 2015. [consulta: 23 de marzo de 2023]. Disponible en: <http://hdl.handle.net/10609/40579>
- [12] Modulate using OFDM method - MATLAB ofdmmod - MathWorks España [en línea] [consulta: 15 de abril de 2023] Disponible en: <https://es.mathworks.com/help/comm/ref/ofdmmod.html>
- [13] Modulate using OFDM method - MATLAB - MathWorks España [en línea] [consulta: 20 de abril de 2023] Disponible en: <https://es.mathworks.com/help/comm/ref/comm.ofdmmodulator-system-object.html>
- [14] *Basic OFDM with No Cyclic Prefix - MATLAB & Simulink - MathWorks España* [en línea] [consulta: 9 de abril de 2023] Disponible en: <https://es.mathworks.com/help/comm/gs/basic-ofdm-with-no-cyclic-prefix.html>
- [15] *OFDM and Equalization with Prependded Cyclic Prefix - MATLAB & Simulink - MathWorks España* [en línea] [consulta: 9 de abril de 2023] Disponible en: <https://es.mathworks.com/help/comm/gs/ofdm-and-equalization-with-prependded-cyclic-prefix.html>
- [16] *OFDM with FFT Based Oversampling - MATLAB & Simulink - MathWorks España* [en línea] [consulta: 9 de abril de 2023] Disponible en: <https://es.mathworks.com/help/comm/gs/ofdm-with-fft-based-oversampling.html>
- [17] S. Kant, M. Bengtsson, G. Fodor, B. Göransson and C. Fischione. EVM-Constrained and Mask-Compliant MIMO-OFDM Spectral Precoding. En: *IEEE Transactions on Wireless Communications*, [en línea]. 2021, vol. 20, no. 1, pp. 590-606. ISSN 1558-2248 [consulta: 13 de mayo de 2023]. Disponible en: <https://ieeexplore.ieee.org/document/9214877>
- [18] IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band. En: *IEEE Std 802.11a-1999* [en línea]. 1999, vol., no., pp.1-102, 30 [consulta: 15 de mayo de 2023]. Disponible en: <https://ieeexplore.ieee.org/document/815305>
- [19] OFDM Raised Cosine Windowing (keysight.com) [en línea] [consulta: 5 de mayo de 2023] Disponible en:

[https://rfmw.em.keysight.com/wireless/helpfiles/n7617a/ofdm\\_raised\\_cosine\\_windowing.htm](https://rfmw.em.keysight.com/wireless/helpfiles/n7617a/ofdm_raised_cosine_windowing.htm)

- [20] Windowing an OFDM Signal in Time Domain | Wireless Pi [en línea] [consulta: 5 de mayo de 2023] Disponible en: <https://wirelesspi.com/windowing-an-ofdm-signal-in-time-domain/>



## 8. Anexos

Se detalla en este anexo el código empleado en la implementación de la aplicación de simulación de señales moduladas mediante OFDM.

```

classdef appOFDMMatlab < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        ParametrosOFDMPanel    matlab.ui.container.Panel
        WindowingCheckBox       matlab.ui.control.CheckBox
        NsmbolosEditField       matlab.ui.control.NumericEditField
        NsmbolosLabel           matlab.ui.control.Label
        SubportadorasPanel      matlab.ui.container.Panel
        MappingportadorasButton matlab.ui.control.Button
        PortadoracentralCheckBox matlab.ui.control.CheckBox
        AladerechaEditField     matlab.ui.control.NumericEditField
        AladerechaEditFieldLabel matlab.ui.control.Label
        AlaizquierdaEditField   matlab.ui.control.NumericEditField
        AlaizquierdaEditFieldLabel matlab.ui.control.Label
        PortadorasdedatosEditField matlab.ui.control.NumericEditField
        PortadorasdedatosEditFieldLabel matlab.ui.control.Label
        SupresindeportadorasCheckBox matlab.ui.control.CheckBox
        OversamplingSlider      matlab.ui.control.Slider
        OversamplingSliderLabel matlab.ui.control.Label
        AplicarButton           matlab.ui.control.Button
        PrefijoclicoCheckBox    matlab.ui.control.CheckBox
        PuntosFFTDropDown       matlab.ui.control.DropDown
        PortadorasNumeroFFTLLabel matlab.ui.control.Label
        SealtemporalPanel       matlab.ui.container.Panel
        OFDMTiempo              matlab.ui.control.UIAxes
        EspectroOFDMPanel       matlab.ui.container.Panel
        OFDMFrecuencia          matlab.ui.control.UIAxes
    end

    properties (Access = private)

        % OFDMModulator instance;
        ofdmMod = comm.OFDMModulator;

    end

    methods (Access = private)

        function OFDM(app)

            % Release resources and allow changes
            release(app.ofdmMod);
        end
    end
end

```

```

% FFTLength
app.ofdmMod.FFTLength = str2double(app.PuntosFFTDropDown.Value);

% Subcarriers treatment
value2= app.SupresindeportadorasCheckBox.Value;
if (value2==1)

    portadorasSuprimidas(app, app.ofdmMod.FFTLength);

    if (app.PortadoracentralCheckBox.Value == 1)
        app.ofdmMod.InsertDCNull = true;
    else
        app.ofdmMod.InsertDCNull = false;
    end

    leftSC = app.AlaizquierdaEditField.Value;
    rightSC = app.AladerechaEditField.Value;
    app.ofdmMod.NumGuardBandCarriers = [leftSC;rightSC];

    % Number of data subcarriers
    app.PortadorasdedatosEditField.Value = app.ofdmMod.FFTLength-
leftSC-rightSC-app.PortadoracentralCheckBox.Value;
    else
        % Number of data subcarriers
        app.PortadorasdedatosEditField.Value = app.ofdmMod.FFTLength;
    end

% Num Symbols
app.ofdmMod.NumSymbols = app.NsmbolosEditField.Value;

% CyclicPrefixLength
cyclicPrefix = app.PrefijoclicoCheckBox.Value;
if (cyclicPrefix == 1)
    app.ofdmMod.CyclicPrefixLength = 16;
else
    app.ofdmMod.CyclicPrefixLength = 0;
end

%app.ofdmMod.PilotInputPort = false;

% Oversampling
app.ofdmMod.OversamplingFactor=round(app.OversamplingSlider.Value, 0);

% Windowing
windowing = app.WindowingCheckBox.Value;
if (windowing == 1 && cyclicPrefix == 1)
    app.ofdmMod.Windowing = true;
    app.ofdmMod.WindowLength = app.ofdmMod.CyclicPrefixLength;
else
    app.ofdmMod.Windowing = false;
end

modDim = info(app.ofdmMod);

```

```

M = 16;

%Random data In
dataIn = randi([0 M-1],modDim.DataInputSize);

dataQAM = qammod(dataIn,M,UnitAveragePower=true);

% if (app.ofdmMod.PilotInputPort)
%   xp = randi([0 M-1],modDim.PilotInputSize);
%   pilotIn = qammod(xp,M,UnitAveragePower=true);
%   sOFDM = app.ofdmMod(dataQAM,pilotIn);
% else
    sOFDM = app.ofdmMod(dataQAM);
% end

% OFDM

%Time
plot(app.OFDMTiempo,1:modDim.OutputSize, [real(sOFDM) imag(sOFDM)]);
legend(app.OFDMTiempo,"Real","Imaginaria");

% Frequency
N=length(sOFDM);
Spectrum1=20*log10(abs(fft(sOFDM,N)));
% Eje frecuencial en Hz.
EjeFreq=(-(N-1)/2:(N-1)/2)/N*app.ofdmMod.OversamplingFactor;
plot(app.OFDMFrecuencia,EjeFreq,fftshift(Spectrum1)-max(Spectrum1));

end

function portadorasSuprimidas(app, fftLength)
value2= app.SupresindeportadorasCheckBox.Value;
if (value2==1)
    if fftLength == 64
        app.AlaizquierdaEditField.Value = 6;
        app.AladerechaEditField.Value = 5;
    else
        app.AlaizquierdaEditField.Value = fftLength/8;
        app.AladerechaEditField.Value = fftLength/8-1;
    end
else
    app.AlaizquierdaEditField.Value = 0;
    app.AladerechaEditField.Value = 0;
end

end

end

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation

```

```

function startupFcn(app)

    OFDM(app);

end

% Button pushed function: AplicarButton
function AplicarButtonPushed(app, event)

    OFDM(app);

end

% Button pushed function: MappingportadorasButton
function MappingportadorasButtonPushed(app, event)

    showResourceMapping(app.ofdmMod);

end

% Value changed function: SupresindeportadorasCheckBox
function SupresindeportadorasCheckBoxValueChanged(app, event)
    value = app.SupresindeportadorasCheckBox.Value;
    if value == 1
        app.AladerechaEditField.Enable = "on";
        app.AladerechaEditField.Editable = "on";
        app.AlaizquierdaEditField.Enable = "on";
        app.AlaizquierdaEditField.Editable = "on";
        app.PortadoracentralCheckBox.Enable = "on";
    else
        app.AladerechaEditField.Enable = "off";
        app.AladerechaEditField.Editable = "off";
        app.AlaizquierdaEditField.Enable = "off";
        app.AlaizquierdaEditField.Editable = "off";
        app.PortadoracentralCheckBox.Enable = "off";
    end
    portadorasSuprimidas(app, str2double(app.PuntosFFTDropDown.Value));
end

% Value changed function: PuntosFFTDropDown
function PuntosFFTDropDownValueChanged(app, event)

    value = str2double(app.PuntosFFTDropDown.Value);
    portadorasSuprimidas(app, value);

end

end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

```

```

% Get the file path for locating images
pathToMLAPP = fileparts(mfilename('fullpath'));

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 1396 767];
app.UIFigure.Name = 'MATLAB App';

% Create EspectroOFDMPanel
app.EspectroOFDMPanel = uipanel(app.UIFigure);
app.EspectroOFDMPanel.Title = 'Espectro OFDM';
app.EspectroOFDMPanel.Position = [239 1 1158 474];

% Create OFDMFrecuencia
app.OFDMFrecuencia = uiaxes(app.EspectroOFDMPanel);
title(app.OFDMFrecuencia, 'Señal OFDM')
xlabel(app.OFDMFrecuencia, 'Frecuencia normalizada')
ylabel(app.OFDMFrecuencia, 'Amplitud')
zlabel(app.OFDMFrecuencia, 'Z')
app.OFDMFrecuencia.Position = [2 0 1155 451];

% Create SealtemporalPanel
app.SealtemporalPanel = uipanel(app.UIFigure);
app.SealtemporalPanel.Title = 'Señal temporal';
app.SealtemporalPanel.Position = [239 474 1158 293];

% Create OFDMTiempo
app.OFDMTiempo = uiaxes(app.SealtemporalPanel);
title(app.OFDMTiempo, 'Señal OFDM')
xlabel(app.OFDMTiempo, 'Tiempo')
ylabel(app.OFDMTiempo, 'Amplitud')
zlabel(app.OFDMTiempo, 'Z')
app.OFDMTiempo.Position = [0 0 1157 268];

% Create ParametrosOFDMPanel
app.ParametrosOFDMPanel = uipanel(app.UIFigure);
app.ParametrosOFDMPanel.Title = 'Parametros OFDM';
app.ParametrosOFDMPanel.Position = [2 1 238 766];

% Create PortadorasNumeroFFTLLabel
app.PortadorasNumeroFFTLLabel = uilabel(app.ParametrosOFDMPanel);
app.PortadorasNumeroFFTLLabel.Position = [6 709 68 26];
app.PortadorasNumeroFFTLLabel.Text = 'Puntos FFT';

% Create PuntosFFTDropDown
app.PuntosFFTDropDown = uidropdown(app.ParametrosOFDMPanel);
app.PuntosFFTDropDown.Items = {'64', '128', '256', '512', '1024',
'2048'};
app.PuntosFFTDropDown.ValueChangedFcn = createCallbackFcn(app,
@PuntosFFTDropDownValueChanged, true);
app.PuntosFFTDropDown.Position = [89 709 100 26];
app.PuntosFFTDropDown.Value = '64';

```

```

% Create PrefijocclicoCheckBox
app.PrefijocclicoCheckBox = uicheckbox(app.ParametrosOFDMPanel);
app.PrefijocclicoCheckBox.Text = 'Prefijo cíclico';
app.PrefijocclicoCheckBox.Position = [4 463 93 22];

% Create AplicarButton
app.AplicarButton = uibutton(app.ParametrosOFDMPanel, 'push');
app.AplicarButton.ButtonPushedFcn = createCallbackFcn(app,
@AplicarButtonPushed, true);
app.AplicarButton.Position = [50 293 104 27];
app.AplicarButton.Text = 'Aplicar';

% Create OversamplingSliderLabel
app.OversamplingSliderLabel = uilabel(app.ParametrosOFDMPanel);
app.OversamplingSliderLabel.Position = [6 383 79 22];
app.OversamplingSliderLabel.Text = 'Oversampling';

% Create OversamplingSlider
app.OversamplingSlider = uislider(app.ParametrosOFDMPanel);
app.OversamplingSlider.Limits = [1 4];
app.OversamplingSlider.MajorTicks = [1 2 3 4];
app.OversamplingSlider.MinorTicks = [];
app.OversamplingSlider.Position = [106 402 116 3];
app.OversamplingSlider.Value = 4;

% Create SubportadorasPanel
app.SubportadorasPanel = uipanel(app.ParametrosOFDMPanel);
app.SubportadorasPanel.Title = 'Subportadoras';
app.SubportadorasPanel.Position = [1 492 236 209];

% Create SupresindeportadorasCheckBox
app.SupresindeportadorasCheckBox = uicheckbox(app.SubportadorasPanel);
app.SupresindeportadorasCheckBox.ValueChangedFcn =
createCallbackFcn(app, @SupresindeportadorasCheckBoxValueChanged, true);
app.SupresindeportadorasCheckBox.Text = 'Supresión de portadoras';
app.SupresindeportadorasCheckBox.Position = [6 161 153 22];

% Create PortadorasdedatosEditFieldLabel
app.PortadorasdedatosEditFieldLabel = uilabel(app.SubportadorasPanel);
app.PortadorasdedatosEditFieldLabel.Enable = 'off';
app.PortadorasdedatosEditFieldLabel.Position = [6 52 113 22];
app.PortadorasdedatosEditFieldLabel.Text = 'Portadoras de datos';

% Create PortadorasdedatosEditField
app.PortadorasdedatosEditField = uieditfield(app.SubportadorasPanel,
'numeric');
app.PortadorasdedatosEditField.Editable = 'off';
app.PortadorasdedatosEditField.Enable = 'off';
app.PortadorasdedatosEditField.Tooltip = {'Numero de portadoras de
datos'};
app.PortadorasdedatosEditField.Position = [128 52 47 22];

% Create AlaizquierdaEditFieldLabel

```

```

app.AlaizquierdaEditFieldLabel = uilabel(app.SubportadorasPanel);
app.AlaizquierdaEditFieldLabel.Enable = 'off';
app.AlaizquierdaEditFieldLabel.Position = [11 140 77 22];
app.AlaizquierdaEditFieldLabel.Text = 'A la izquierda';

% Create AlaizquierdaEditField
app.AlaizquierdaEditField = uieditfield(app.SubportadorasPanel,
'numeric');
app.AlaizquierdaEditField.Editable = 'off';
app.AlaizquierdaEditField.Enable = 'off';
app.AlaizquierdaEditField.Position = [154 138 44 25];
app.AlaizquierdaEditField.Value = 6;

% Create AladerechaEditFieldLabel
app.AladerechaEditFieldLabel = uilabel(app.SubportadorasPanel);
app.AladerechaEditFieldLabel.Enable = 'off';
app.AladerechaEditFieldLabel.Position = [11 113 72 22];
app.AladerechaEditFieldLabel.Text = 'A la derecha';

% Create AladerechaEditField
app.AladerechaEditField = uieditfield(app.SubportadorasPanel,
'numeric');
app.AladerechaEditField.Editable = 'off';
app.AladerechaEditField.Enable = 'off';
app.AladerechaEditField.Position = [154 111 44 25];
app.AladerechaEditField.Value = 5;

% Create PortadoracentralCheckBox
app.PortadoracentralCheckBox = uicheckbox(app.SubportadorasPanel);
app.PortadoracentralCheckBox.Enable = 'off';
app.PortadoracentralCheckBox.Text = 'Portadora central';
app.PortadoracentralCheckBox.Position = [11 83 114 22];

% Create MappingportadorasButton
app.MappingportadorasButton = uibutton(app.SubportadorasPanel, 'push');
app.MappingportadorasButton.ButtonPushedFcn = createCallbackFcn(app,
@MappingportadorasButtonPushed, true);
app.MappingportadorasButton.Icon = fullfile(pathToMLAPP, 'external-
link-line.png');
app.MappingportadorasButton.Position = [39 17 136 23];
app.MappingportadorasButton.Text = 'Mapping portadoras';

% Create NsmbolosLabel
app.NsmbolosLabel = uilabel(app.ParametrosOFDMPanel);
app.NsmbolosLabel.Position = [6 427 79 22];
app.NsmbolosLabel.Text = 'Nº símbolos';

% Create NsmbolosEditField
app.NsmbolosEditField = uieditfield(app.ParametrosOFDMPanel,
'numeric');
app.NsmbolosEditField.Position = [154 424 44 28];
app.NsmbolosEditField.Value = 4;

% Create WindowingCheckBox

```

```

    app.WindowingCheckBox = uicontrol(app.ParametrosOFDMPanel);
    app.WindowingCheckBox.Text = 'Windowing';
    app.WindowingCheckBox.Position = [5 338 81 22];

    % Show the figure after all components are created
    app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = appOFDMMatlab

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
    end
end
end
end

```