

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y  
ELECTRÓNICA**

**ANÁLISIS MEDIANTE SIMULACIÓN DE LA DISTRIBUCIÓN DE  
SUBPORTADORAS DE UN SISTEMA OFDMA CON EL  
ALGORITMO GEOMETRIC WATER-FILLING UTILIZANDO  
MATLAB**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**KEVIN JAIR MACHADO CARVAJAL**

**(kevin.machado@epn.edu.ec)**

**DIRECTOR: Dr. DIEGO JAVIER REINOSO CHISAGUANO**

**(diego.reinoso@epn.edu.ec)**

**Quito, enero 2020**

## **AVAL**

Certifico que el presente trabajo fue desarrollado por Kevin Jair Machado Carvajal, bajo mi supervisión.

---

**Dr. Diego Javier Reinoso Chisaguano**  
**DIRECTOR DEL TRABAJO DE TITULACIÓN**

## **DECLARACIÓN DE AUTORÍA**

Yo, Kevin Jair Machado Carvajal, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

---

Kevin Jair Machado Carvajal

## **DEDICATORIA**

A mi madre Piedad quien siempre me cuida y me da la confianza de seguir adelante, a mi padre Luis, fuente de mi carácter y rectitud ante toda situación y a mi hermano Luis con quien siempre puedo contar.

## **AGRADECIMIENTO**

A mis padres, Luis Machado y Piedad Carvajal, por todo el amor y apoyo que me han brindado durante toda mi vida. Es gracias a ellos que he llegado a quien soy, gracias a su ejemplo y enseñanzas que me ofrecieron a lo largo de mi vida.

Agradezco a mi hermano Luis, por ayudarme y aconsejarme con gran paciencia, sin importar las circunstancias.

A mi familia, por todo el apoyo que me dieron, demostrándome el mejor ejemplo de fraternidad, permaneciendo siempre unidos. Agradezco especialmente a mi madrina Mariana, quien me recibió en su hogar y cuidó de mí durante esta etapa de mi vida estudiantil.

Agradezco al Dr. Diego Reinoso, por su guía y compromiso durante el desarrollo de este trabajo.

A mis amigos por todas las vivencias y alegrías a lo largo de nuestra vida estudiantil.

# ÍNDICE DE CONTENIDO

AVAL.....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VII
ABSTRACT.....	VIII
1. INTRODUCCIÓN .....	1
1.1 OBJETIVOS.....	2
1.2 ALCANCE .....	2
1.3 MARCO TEÓRICO .....	3
1.3.1 OFDM.....	3
1.3.2 OFDMA .....	10
1.3.3 CANAL INALÁMBRICO .....	15
1.3.4 ANÁLISIS DE WATER-FILLING .....	19
1.3.5 GEOMETRIC WATER-FILLING.....	20
2. METODOLOGÍA.....	23
2.1 PROGRAMA PRINCIPAL .....	23
2.2 TRANSMISOR .....	27
2.3 CANAL.....	31
2.4 ASIGNACIÓN DE RECURSOS.....	34
2.5 GEOMETRIC WATER FILLING .....	35
2.6 CAPACIDAD .....	38
2.7 GRÁFICAS.....	41
3. RESULTADOS Y DISCUSIÓN .....	44
3.1. CAPACIDAD DEL SISTEMA CON DOS USUARIOS CON UN CANAL DE 20 MHZ.....	45
3.1.1. CAPACIDAD TOTAL .....	45
3.1.1.2. Modulación 16-QAM.....	45

3.1.1.3.	Modulación 64-QAM.....	46
3.1.1.4.	Modulaciones QPSK, 16-QAM y 64-QAM.....	47
3.1.2.	CAPACIDAD POR USUARIO .....	48
3.1.2.1.	Modulación QPSK .....	48
3.1.2.2.	Modulación 16-QAM.....	49
3.1.2.3.	Modulación 64-QAM.....	49
3.2.	CAPACIDAD DEL SISTEMA CON DOS USUARIOS CON UN CANAL DE 40 MHZ.....	50
3.2.1.	CAPACIDAD TOTAL .....	50
3.2.2.	CAPACIDAD POR USUARIO .....	51
3.2.2.1.	Modulación QPSK .....	51
3.2.2.2.	Modulación 16-QAM.....	52
3.2.2.3.	Modulación 64-QAM.....	52
3.3.	CAPACIDAD DEL SISTEMA CON CUATRO USUARIOS CON UN CANAL DE 20 MHZ .....	53
3.3.1.	CAPACIDAD TOTAL .....	53
3.3.2.	CAPACIDAD POR USUARIO .....	54
3.4.	CAPACIDAD DEL SISTEMA CON CUATRO USUARIOS CON UN CANAL DE 40 MHZ .....	56
3.4.1.	CAPACIDAD TOTAL .....	56
3.4.2.	CAPACIDAD POR USUARIO .....	57
3.4.2.1.	Modulación QPSK .....	57
3.4.2.2.	Modulación 16-QAM.....	57
3.4.2.3.	Modulación 64-QAM.....	58
3.5.	RESULTADOS FINALES .....	59
4.	CONCLUSIONES Y RECOMENDACIONES.....	61
3.1.	CONCLUSIONES .....	61
3.2.	RECOMENDACIONES .....	62
5.	REFERENCIAS BIBLIOGRÁFICAS .....	64
	ANEXOS .....	65

## RESUMEN

En el presente trabajo se presenta la simulación en MATLAB de un sistema OFDMA (*Orthogonal Frequency-Division Multiple Access*) en el cual se evalúa la mejora de la capacidad que se obtiene con el uso del algoritmo GWF (*geometric water-filling*), obteniendo una mejora promedio de hasta un 10.40% en un rango de 8 a 16 dB utilizando la modulación 64-QAM, para los métodos de asignación de subportadoras tipo bloque, aleatorio y alternado. Las simulaciones se realizan para 2 y 4 usuarios, utilizando como modelo la capa física del estándar 802.11n. Como resultado se obtienen curvas de capacidad versus SNR, así como tablas que reflejan la mejora porcentual de la capacidad que se obtiene con el algoritmo GWF.

En el capítulo 1 se definen los conceptos presentes en un sistema OFDMA, como es la tecnología OFDM, análisis del transmisor y receptor OFDMA, métodos de asignación de subportadoras, medio de transmisión inalámbrico, entre otros. También describe el funcionamiento del algoritmo GWF que se implementará para la optimización de la asignación de potencia de cada subportadora.

En el capítulo 2 se describe la implementación en MATLAB de cada parte del sistema OFDMA descrita en el capítulo 1, incluyendo la implementación del cálculo de la capacidad total y por usuario del sistema OFDMA.

En el capítulo 3 se muestran y analizan las gráficas de la capacidad total y por usuario que se obtienen con y sin el uso del algoritmo GWF para cada método de asignación de subportadoras, así como tablas que reflejen la mejora porcentual de la capacidad.

Finalmente, en el capítulo 4 se muestran las conclusiones y recomendaciones del desarrollo y resultados obtenidos del presente trabajo.

**PALABRAS CLAVE:** OFDM, OFDMA, asignación de potencia, asignación de subportadoras, IEEE 802.11n, *geometric water-filling*.

## **ABSTRACT**

This project presents the MATLAB simulation of an OFDMA (Orthogonal Frequency-Division Multiple Access) system that evaluates the improvement of the capacity offered by the GWF (geometric water-filling) algorithm, obtaining an average improvement of up to 10.40% in a range of 8 to 16 dB using 64-QAM modulation, for three different subcarrier allocation methods. The simulation works with 2 or 4 users, based on the physical layer of the 802.11n standard. As a result of the simulation, capacity curves versus SNR are obtained, as well as tables that specify the improvement in capacity obtained with the GWF algorithm.

Chapter 1 defines the concepts of an OFDMA system, such as OFDM technology, OFDMA transmitter and receiver, subcarrier allocation methods and wireless transmission. We also describe the operation of the GWF algorithm that is implemented for the optimization of subcarrier power allocation.

Chapter 2 describes the MATLAB implementation of each part of the OFDMA system described in Chapter 1, including the implementation of the total and per-user capacity calculation of the OFDMA system.

Chapter 3 shows and analyzes the graphs of the total and per-user capacity that are obtained with and without the use of the GWF algorithm for each method of subcarrier allocation, as well as tables that reflect the percentage improvement in capacity.

Finally, chapter 4 shows the conclusions and recommendations of the development and results obtained from this work.

**KEYWORDS:** OFDM, OFDMA, power allocation, sub-carrier allocation, IEEE 802.11n, geometric water-filling.

# 1. INTRODUCCIÓN

En la actualidad, los servicios de telecomunicaciones se han vuelto más accesibles gracias a la flexibilidad y cobertura que brinda la tecnología inalámbrica a la cual día a día se le exige un mayor rendimiento para satisfacer la necesidad de los usuarios. El canal inalámbrico se caracteriza por ser un medio de transmisión dinámico, es decir, que cambia a cada instante sin importar si el transmisor o el receptor permanecen en una misma posición o se encuentran en movimiento durante la transmisión ya que durante el trayecto los obstáculos pueden variar y estos afectan las condiciones del mismo.

Entre las ventajas de utilizar un medio inalámbrico frente a otros medios de transmisión se tiene la movilidad, altas tasas de transmisión y simultaneidad para múltiples usuarios. Para satisfacer las altas tasas de transmisión requeridas se debe emplear una modulación lo suficientemente robusta para las condiciones del canal sobre el cual se va a transmitir. Entre las modulaciones más utilizadas en medios inalámbricos se encuentran QPSK (*Quadrature Phase-Shift Keying*), 16-QAM (*Quadrature Amplitude Modulation*) y 64-QAM, las cuales aumentan la tasa de transmisión pero a su vez reducen su robustez contra errores ocasionados por el canal. Para satisfacer la alta demanda del acceso de múltiples usuarios se emplea la técnica de OFDM (*Orthogonal Frequency-Division Multiplexing*) en combinación con FDMA (*Frequency Division Multiple Access*), TDMA (*Time Division Multiple Access*) o CDMA (*Code Division Multiple Access*). OFDM es una técnica de transmisión en la que la información viaja en un conjunto de subportadoras ortogonales en paralelo a diferentes frecuencias las cuales son moduladas en PSK o QAM.

En OFDM-FDMA, también conocido como OFDMA, se asignan cierto número de subportadoras a cada usuario. Esta asignación puede ser por subportadoras adyacentes el cual consiste en asignar grupos consecutivos de subportadoras a cada usuario, por subportadoras alternadas asignando, como su nombre lo indica, de manera alternada cada subportadora a un usuario y por diversidad de subportadoras en la cual se realiza una asignación aleatoria para cada usuario. Solo utilizando este tipo de asignaciones de subportadoras no se aprovecha de manera óptima el espectro disponible ya que cada grupo de subportadoras que reciba un usuario puede verse afectado de manera negativa por las condiciones del canal.

Para mejorar la tasa de transmisión de cada usuario, este trabajo presenta una optimización en la distribución de la potencia con el uso del algoritmo *geometric water-filling*, el cual, en base a la respuesta del canal, redistribuye la potencia de cada

subportadora. De esta forma se compensa la selectividad en frecuencia del canal, ya que a las subportadoras que presenten un mayor SNR (*Signal-to-Noise Ratio*) se les asignará una mayor potencia para aprovechar el buen estado del subcanal. Al incrementar el SNR de cada subportadora se consigue optimizar la distribución de los recursos disponibles con lo cual se aumenta de manera considerable la capacidad del canal y se mejoran las tasas para cada usuario ya que pese a la atenuación presente en el canal, cada subportadora que se utilice tendrá la máxima potencia posible para la transmisión.

El presente trabajo consiste en la implementación mediante simulación de un sistema OFDMA el cual emplea el algoritmo *geometric water-filling* para la asignación de potencia para las subportadoras en conjunto con otros métodos de asignación. Se podrá maximizar la capacidad del sistema, optimizando la asignación de las subportadoras para cada usuario. Además, se podrá comparar el rendimiento de este algoritmo con otras técnicas convencionales, así como los efectos de cada modulación para cierto número de usuarios.

## 1.1 OBJETIVOS

El objetivo general de este Trabajo de Titulación es:

- Analizar mediante simulación la distribución de subportadoras de un sistema OFDMA con el algoritmo *Geometric Water-Filling* utilizando MATLAB.

Los objetivos específicos de este Trabajo de Titulación son:

- Describir el sistema de comunicación OFDMA y la forma en la que se asignan los recursos (subportadoras).
- Implementar en la simulación los tipos de asignaciones comúnmente utilizados en OFDMA (tipo bloque, tipo combinación, tipo randómico).
- Implementar en la simulación el algoritmo *geometric water-filling* para la asignación óptima de los recursos.
- Analizar los resultados obtenidos por la simulación.

## 1.2 ALCANCE

En este trabajo se desarrollará una simulación en MATLAB de un sistema de comunicaciones inalámbricas que emplea OFDM-FDMA sobre un canal selectivo en

frecuencia con una distribución de *Rayleigh*. En el receptor se considerará una sincronización perfecta y que se conoce el estado del canal inalámbrico. Además, se implementarán cuatro diferentes tipos de asignación de subportadoras. Estos cuatro tipos de asignación son: 1) tipo bloque, 2) tipo combinación, 3) tipo randómico y 4) con el algoritmo *geometric water-filling*. Se utilizarán 3 tipos de modulaciones QPSK, 16-QAM y 64-QAM por separado. Adicionalmente, se podrá seleccionar el número de usuarios (entre 2 y 4), para evaluar el efecto de los mismos sobre el rendimiento del sistema. Se comparará el rendimiento con 64 subportadoras en un canal de 20 MHz y con 128 subportadoras en un canal de 40 MHz.

Los resultados serán presentados en gráficas de la capacidad total vs SNR comparando los diferentes tipos de asignación. Los métodos de asignación se presentarán bajo diferentes parámetros como: número de usuarios, número de portadoras y tipo de modulación para comprobar su rendimiento. Finalmente, se realizará una discusión de los resultados obtenidos de las simulaciones.

## 1.3 MARCO TEÓRICO

### 1.3.1 OFDM

OFDM es una técnica de transmisión multiportadora que al contrario de técnicas monoportadora, divide la señal de entrada en múltiples flujos paralelos, cada uno de los cuales es asignado a una subportadora. La diferencia de OFDM con respecto a otros esquemas multiportadora es que esta técnica emplea subportadoras ortogonales entre sí con lo que se consigue una alta eficiencia espectral [1]. En OFDM el ancho de banda disponible  $B$  se divide en  $N$  canales de banda estrecha en paralelo, cada uno con un ancho de banda  $B/N$  y una frecuencia  $f_k$ , donde  $k = 0, \dots, N - 1$ . Las subportadoras tienen una separación dada por la Ecuación 1.1.

$$\Delta f = \frac{B}{N} = \frac{1}{T_s} \quad (1.1)$$

donde  $T_s$  es la duración del símbolo, por lo tanto si la frecuencia de la primera subportadora es  $f_0$ , las siguientes subportadoras se encuentran distribuidas como se muestra en la Ecuación 1.2.

$$f_k = f_0 + k \frac{B}{N} \quad (1.2)$$

Si  $\Delta f$  es menor que el ancho de banda de coherencia del canal cada subportadora experimentará un desvanecimiento plano en frecuencia ya que en el ancho de banda de

coherencia se considera que el canal tiene la misma ganancia y el incremento de fase es lineal y por lo tanto no se generará distorsión en el espectro de la señal transmitida [2].

Si se considera que se va a transmitir  $N$  datos, cada uno representado por  $x_k$  que es un número complejo correspondiente a la constelación de la modulación empleada, que puede ser QPSK, 16-QAM o 64-QAM, el símbolo resultante viene dado por la Ecuación 1.3 el cual tiene un valor complejo.

$$x(t) = \sum_{k=0}^{N-1} x_k e^{j 2\pi f_k t} \quad (1.3)$$

Dado que las frecuencias de las subportadoras tienen una separación constante dada por la Ecuación 1.1 con lo que la frecuencia de cada subportadora se define como  $f_k = k * f_s$  donde  $f_s$  es la frecuencia de separación entre cada subportadora. Reemplazando esa característica, en el dominio de la frecuencia se tiene el símbolo dado por la Ecuación 1.4 [3].

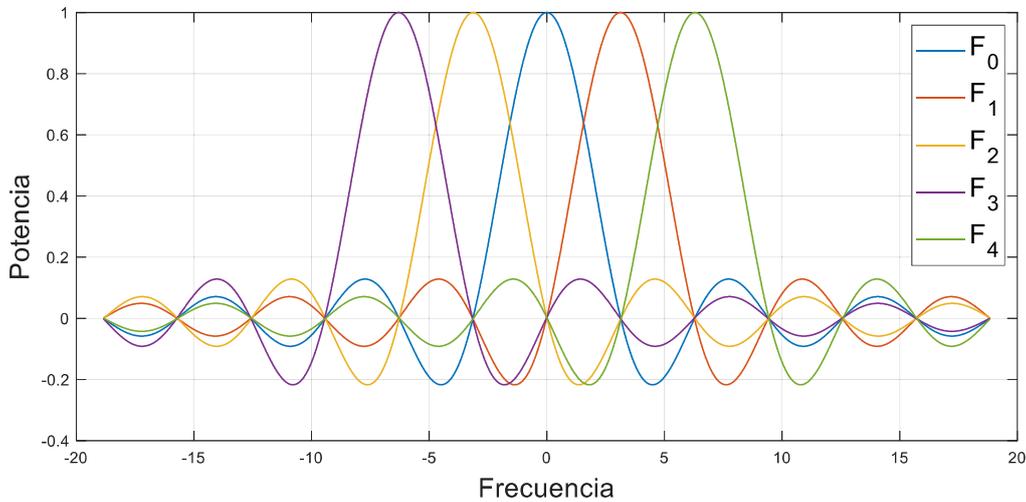
$$X_n = \sum_{k=0}^{N-1} x_k e^{\frac{j2\pi nk}{N}} \quad (1.4)$$

La eficiencia de OFDM radica en que las subportadoras ortogonales se pueden sobreponer una a otra sin causar interferencia entre sí como se muestra en la Figura 1.1 en la cual cada punto máximo de potencia de una subportadora coincide con el cruce por cero de las demás subportadoras logrando de esta manera reducir la interferencia entre sí. En OFDM cada subportadora transmite datos a una baja velocidad, pero se transmiten varias de las mismas en paralelo con lo que se consigue una elevada tasa de transmisión como se muestra en la Figura 1.2 en la que entra una señal con una alta tasa de transmisión en serie y se convierte en varias señales con una menor tasa [4].

Como se mencionó con anterioridad en cada subportadora se transmite una parte de la información modulada de manera independiente de las demás subportadoras y pese a que cada subcanal presenta un desvanecimiento plano en frecuencia, en conjunto el canal presenta un desvanecimiento selectivo en frecuencia y desde un punto de vista general ciertos canales se ven en peores condiciones que los otros. Frente a este problema en sistemas OFDM se codifica su información para de esta manera poder recuperarla con mayor facilidad si una parte de ella presenta daños [2].

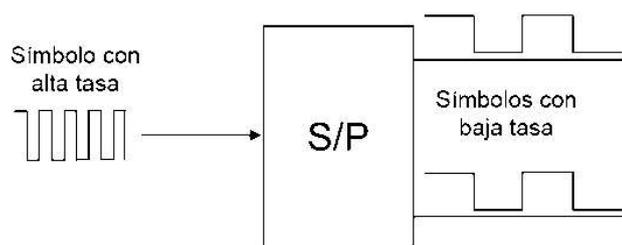
Además de la codificación, en sistemas OFDM normalmente no transmite información en las subportadoras de los extremos, ya que de esta manera se evita interferencia con

bandas adyacentes. Aquellas subportadoras que no llevan información se denominan subportadoras null o banda de guarda [3].



**Figura 1.1.** Subportadoras ortogonales para OFDM.

Entre las características favorables de OFDM se tiene su fácil implementación para el proceso de modulación y demodulación con el uso de la transformada rápida inversa de Fourier (*Inverse Fast Fourier Transform*, IFFT) y la transformada rápida de Fourier (*Fast Fourier Transform*, FFT). Otra característica de OFDM es su facilidad para adaptarse al canal inalámbrico gracias a la subcanalización y poseer una de las mayores eficiencias espectrales frente a las demás técnicas de transmisión lo que la hace ser la técnica más utilizada en aplicaciones de comunicación sobre canales inalámbricos [5].



**Figura 1.2.** Conversión Serie a Paralelo para transmisión de datos en OFDM [4].

### 1.3.1.1 APLICACIONES DE OFDM

Cuando se planteó OFDM no se podía aplicar de manera efectiva ya que no se contaba con la tecnología necesaria para realizar el procesamiento digital de la señal y su correspondiente FFT e IFFT. Los avances de la tecnología y el desarrollo de circuitos integrados favorecieron la expansión de OFDM gracias a lo cual esta tecnología se convirtió en la favorita para sistemas inalámbricos. La principal ventaja de OFDM en el

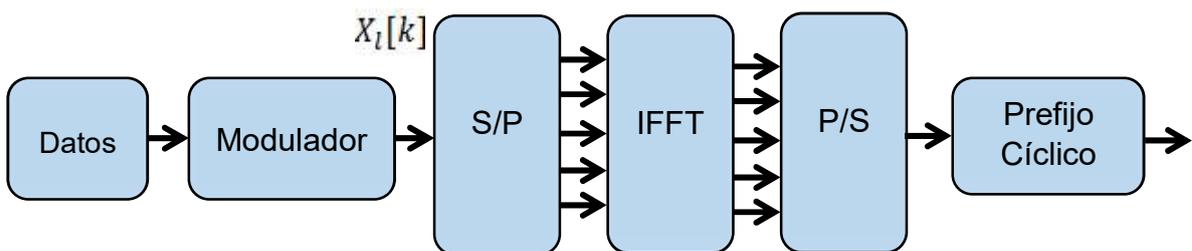
ámbito comercial es la facilidad de la implementación del receptor OFDM en el dispositivo de cada usuario [5].

Para aplicaciones de telefonía móvil OFDM presenta las ventajas previamente mencionadas, un receptor de baja complejidad y la capacidad de OFDM para adaptarse a un canal dinámico. Actualmente OFDM se emplea en el estándar IEEE 802.16 conocido como WiMAX (*Worldwide Interoperability for Microwave Access*), en el enlace de bajada de LTE (*Long Term Evolution*) que es el estándar de cuarta generación para telefonía móvil [6].

La versión inicial del estándar 802.11 utilizó como esquema de modulación el salto de frecuencias la cual solo alcanzaba tasas de 1 Mbps o 2 Mbps, por lo que hubo quejas por parte de los usuarios. Como respuesta a la demanda de mayores tasas de transmisión se cambió a un esquema de modulación diferente llamado OFDM. Esta mejora de esquema impulsó las tasas de transmisión de 802.11a/g hasta 54 Mbps. Pese a este incremento se vio en la necesidad de aumentar las tasas por lo que en 802.11n se utilizan bandas de frecuencia más amplias y hasta cuatro antenas por nodo, logrando tasas de hasta 540 Mbps [7].

### 1.3.1.2 TRANSMISOR OFDM

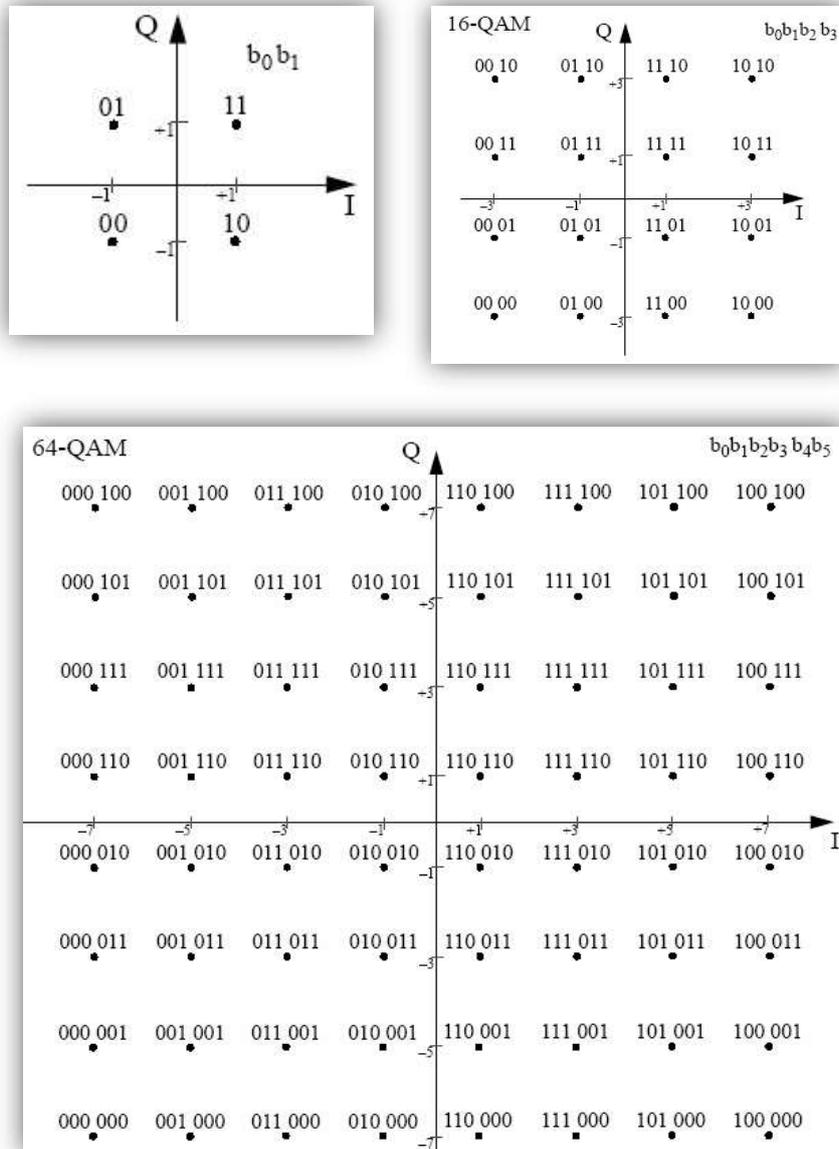
Considerando que se desea transmitir una señal de datos binarios que se originan en un dispositivo móvil, ya sean datos de una aplicación o de una llamada, dicha señal debe seguir una serie de pasos, los cuales serán descritos a continuación. En la Figura 1.3 se muestra el diagrama de bloques de un transmisor OFDM.



**Figura 1.3.** Diagrama de bloques para el transmisor OFDM.

Como primer paso, la señal binaria se mapea según el esquema de modulación empleado, ya sea QPSK, 16-QAM o 64-QAM. Cada modulación tiene su propia constelación como se muestra en la Figura 1.4 y dependiendo de la separación de los símbolos de la constelación se obtendrá una mayor resistencia a los errores. Como se muestra en la Figura 1.4 la constelación de QPSK puede aceptar un desfase de  $\pm 45$  grados y una gran atenuación ya que prácticamente un cuadrante pertenece a un solo

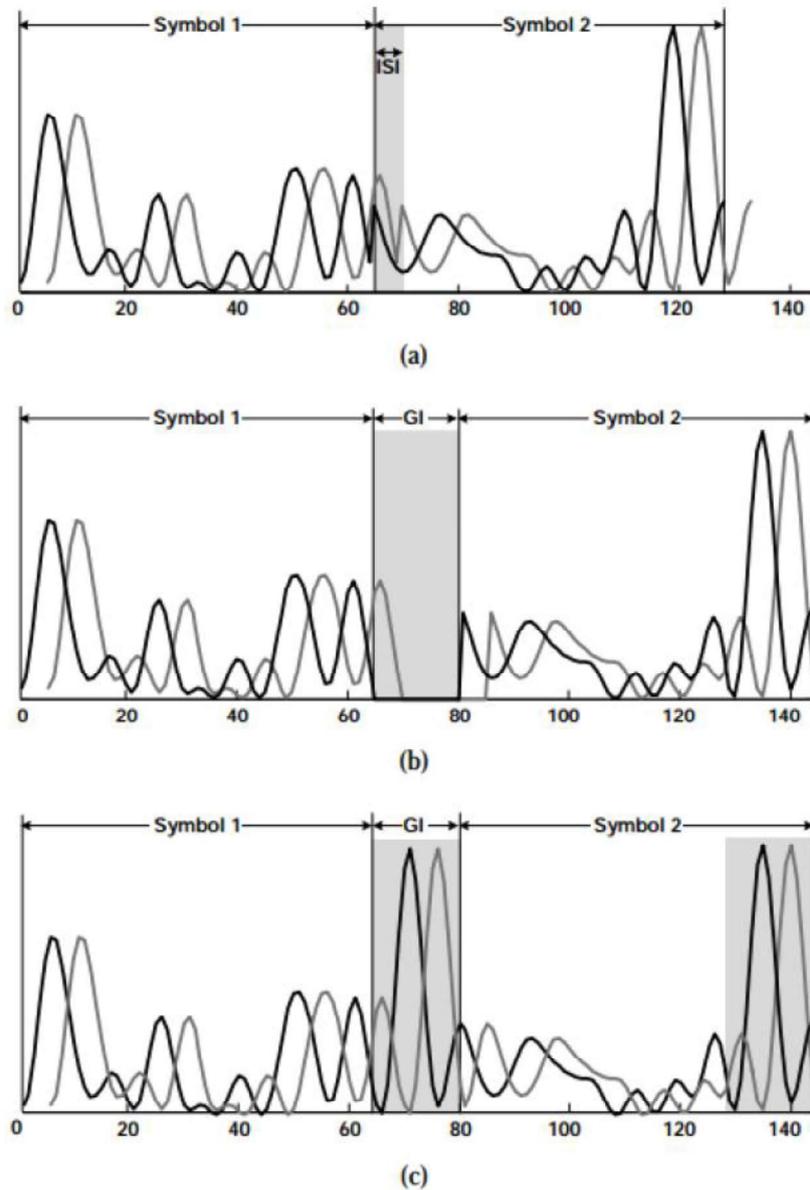
símbolo, mientras que en 64-QAM el mismo cuadrante es compartido por 16 símbolos los cuales se encuentran muy próximos, lo que reduce la variación que estos aceptan y aumenta la posibilidad de confundirlos al existir cambios en fase o amplitud [5].



**Figura 1.4.** Diagrama de constelación de QPSK, 16-QAM Y 64-QAM

Una vez los símbolos son modulados, estos ingresan al convertor serie a paralelo (S/P), obteniendo como resultado un flujo de  $N$  datos en paralelo los cuales serán asignados a cada subportadora. En las primeras versiones de OFDM se empleaba un banco de moduladores para obtener el símbolo complejo lo cual resultaba ser un proceso muy

complicado, en reemplazo de ese proceso se utiliza IFFT obteniendo como resultado los símbolos  $x_l[k]$  donde  $k = 0, \dots, N - 1$  [2].



**Figura 1.5.** (a) Interferencia entre símbolos, (b) Uso de zero padding, (c) Uso del prefijo cíclico [7].

A continuación, se convierte la señal en serie para añadir el prefijo cíclico durante el intervalo de guarda de la señal OFDM. OFDM utiliza el intervalo de guarda para combatir el ISI, el cual consiste en dejar una separación entre los símbolos en la cual se puedan alojar las copias y estas no afecten al símbolo contiguo. En el intervalo de guarda de OFDM se puede aplicar 2 formas para combatir el ISI las cuales son: *Zero Padding* (ZP) y el prefijo cíclico (CP, Cyclic Prefix).

*Zero Padding* consiste en llenar el intervalo de guarda con subportadoras null como se muestra en la Figura 1.5. (b) en la cual al ser subportadoras null no llevan información y prácticamente el espacio de separación entre los símbolos no es utilizado, mientras que si se usa el prefijo cíclico se selecciona una fracción final del símbolo y se repite en el inicio del mismo generando redundancia como se muestra en la Figura 1.5 (c) [7].

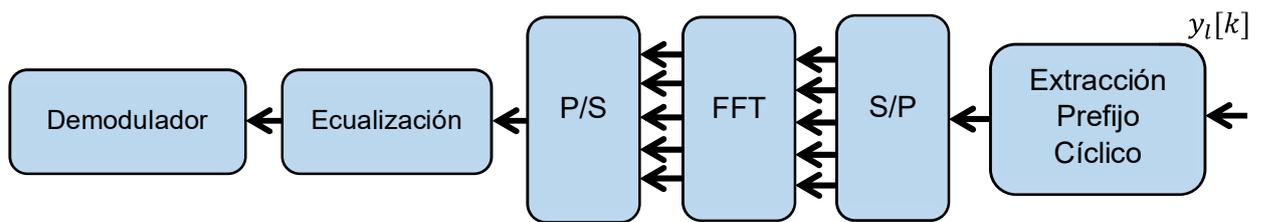
### 1.3.1.3 RECEPTOR OFDM

Después de que la señal atraviesa el canal inalámbrico, dicha señal se ve afectada por el ruido propio del canal y por el desvanecimiento (para este trabajo se considera un desvanecimiento de tipo Rayleigh debido a que no existe línea de vista entre el transmisor y el receptor). En el receptor se realiza el proceso inverso al del transmisor como se muestra en la Figura 1.6. Primero se extrae el prefijo cíclico de la señal recibida  $y_l$  la cual es igual a la convolución en tiempo de la respuesta impulsiva del canal y la señal transmitida descrita en la Ecuación 1.5.

$$y_l = H * x_l + w \quad (1.5)$$

donde  $H$  es la respuesta impulsiva del canal la cual puede ser estimada o asumida,  $x_l$  es la señal transmitida y  $w$  es el ruido propio del canal. A continuación, la señal se convierte a paralelo y se le aplica la FFT en la cual se obtienen los símbolos transmitidos en cada subportadora. Estos símbolos se ven afectados por el desvanecimiento de *Rayleigh* el cual ocasiona desplazamientos en la fase de los símbolos y atenuación en sus amplitudes lo que puede llegar a incrementar la tasa de bits errados (BER, *Bit Error Rate*). Este problema puede ser corregido por una adecuada ecualización la cual en OFDM se puede realizar de manera más sencilla que en técnicas de transmisión de una sola portadora, que para estos tipos de técnicas se requiere de ecualizadores adaptativos en dominio del tiempo. Para realizar la ecualización se requiere conocer la respuesta impulsiva del canal, la cual puede ser estimada en base al uso de señales piloto. El receptor conoce la frecuencia y amplitud con la que dichas señales son transmitidas con lo que puede medir el estado del canal para ese instante y a las frecuencias de las señales piloto. Dado que la estimación del canal no es 100% segura, los símbolos que se obtienen después de la ecualización son símbolos estimados [5].

Por último los símbolos estimados por el ecualizador se demodulan, obteniendo como resultado bits estimados.



**Figura 1.6.** Diagrama de bloques para el receptor OFDM.

Otro factor que interviene en el receptor es el ruido agregado por el canal, factor que es combatido con la modulación previamente aplicada en el transmisor. La elección del tipo de modulación se debe realizar en base a la robustez necesaria para combatir el ruido del canal y a la tasa de transmisión que se desea conseguir [4].

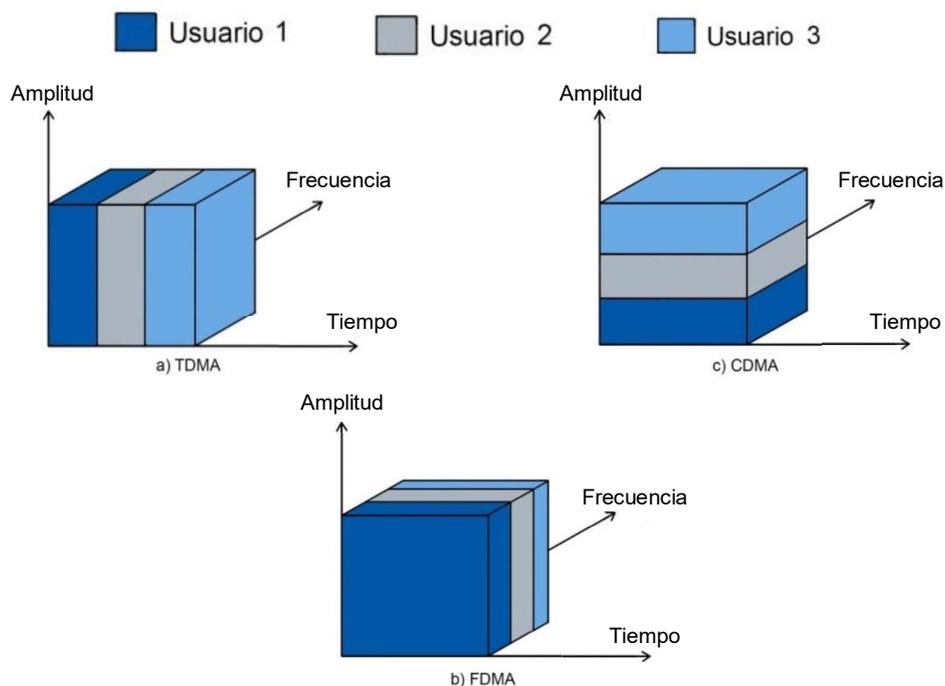
### 1.3.2 OFDMA

OFDMA extiende la tecnología OFDM para brindar un sistema de acceso para múltiples usuarios. Previamente se analizó como OFDM entrega todas sus subportadoras a un único usuario durante todo el tiempo requerido por el mismo, en cambio OFDMA distribuye sus subportadoras a diferentes usuarios, gracias a lo cual pueden recibir datos de manera simultánea [2]. La forma más simple de distribuir las subportadoras a los usuarios es mediante la asignación de subportadoras contiguas para disminuir la sobrecarga que conlleva indicar a cada usuario en que momento y cuales subportadoras debe emplear para transmitir su información. Pese que este método si reduce el procesamiento computacional, también reduce la eficiencia del sistema de comunicaciones lo cual puede afectar los usuarios.

En sistemas OFDMA, tanto recursos en tiempo como en frecuencia son utilizados para la asignación de la información de los usuarios y dependiendo de si la aplicación del usuario requiere altas tasas por cortos periodos de tiempo o un acceso total por largos periodos de tiempo se puede combinar OFDM con diferentes métodos de acceso. Los métodos de acceso utilizados pueden ser: acceso múltiple por división de tiempo (TDMA, *Time Division Multiple Access*), acceso múltiple por división de frecuencia (FDMA, *Frequency Division Multiple Access*) y acceso múltiple por división de código (CDMA, *Code Division Multiple Access*) como se muestra en la Figura 1.7 en la cual cada color representa a un usuario y el uso de los recurso tanto de potencia, frecuencia y tiempo que utiliza durante su transmisión [8].

El método TDMA se basa en el esquema de multiplexación por división de tiempo en el cual se divide el tiempo de acceso en ranuras las cuales son asignadas a cada usuario. Si el usuario desea transmitir información deberá esperar por su correspondiente ranura y

solo podrá transmitir durante la duración de la misma como se muestra en la Figura 1.8. (a) en la cual se puede observar que el usuario 1 dispone de toda la potencia y todo el espectro durante cierto periodo de tiempo. Al combinar este método con OFDM se asignan todas las subportadoras a un solo usuario durante una ranura de tiempo y deberá esperar a que los demás no utilicen el medio para poder volver a transmitir. Dependiendo de los requerimientos de la aplicación del usuario se puede asignar un mayor o menor número de ranuras. Una ventaja de este método es que al utilizar todos los recursos se pueden alcanzar altas tasas de transmisión ya que prácticamente durante ese periodo de tiempo se tiene un equivalente de un enlace dedicado. La desventaja de este método de acceso es que se debe tener una buena sincronización para que cada usuario acceda en su correspondiente ranura de tiempo [8].



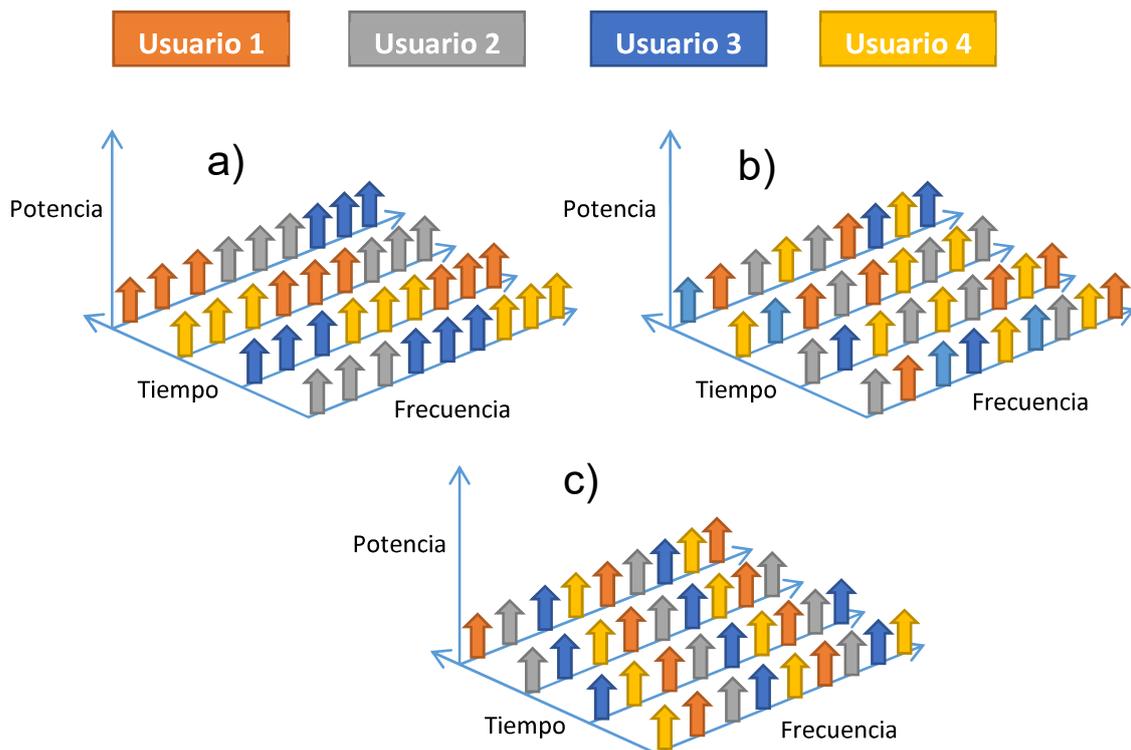
**Figura 1.7. Mecanismos de Acceso Múltiple**

Para el caso de FDMA el ancho de banda total es dividido en varios canales que son asignados a cada usuario permitiéndoles transmitir su información de manera simultánea pero con un ancho de banda menor, lo que implica una reducción en la tasa de transmisión. Cuando se combina este método con OFDM se asigna cierto grupo de subportadoras a cada usuario las cuales no son compartidas y dicho usuario puede acceder a las mismas cuando requiera transmitir información ya que son de uso exclusivo de ese usuario como se observa en la Figura 1.7 (b) en la que el usuario 2 dispone de un espectro limitado pero durante todo el tiempo que requiera [6].

En CDMA los usuarios pueden transmitir de manera simultánea y a una misma frecuencia gracias al uso de un código de propagación que es único para cada usuario como se muestra en la Figura 1.7 (c) en la que cada usuario puede ocupar todo el espectro disponible durante todo el tiempo que requiera [6].

### 1.3.2.1 ASIGNACIÓN DE SUBPORTADORAS EN OFDMA

Una de las ventajas mencionadas con anterioridad es que en OFDMA existe una asignación dinámica de recursos ya sean temporales o en frecuencia. Para el caso de OFDM-FDMA existen tres formas en las que se puede asignar las subportadoras. La primera es el método de subportadoras adyacentes (ASM, *Adjacent Subcarrier Method o Sequential Subcarrier Allocation*), la segunda es el método de diversidad de subportadoras (DSM, *Diversity Subcarrier Method o Random Subcarrier Allocation*) y la tercera es el método de subportadoras alternadas en la que como su nombre lo indica asigna subportadoras de manera alternada para cada usuario [6].



**Figura 1.8.** Métodos de asignación de subportadoras a) Asignación por bloques, b) Asignación aleatoria, c) Asignación alternada [6].

Con el método ASM se asignan bloques contiguos de subportadoras para cada usuario como se muestra en la Figura 1.8 a) en la cual cada color representa a un usuario y su grupo de subportadoras las cuales se encuentran una junto a la otra. Debido a la cercanía de las subportadoras, éstas se encuentran correlacionadas y por lo tanto

presentan respuestas en frecuencia muy similares. Si las subportadoras no estuvieran correlacionadas se debería enviar un mensaje de retroalimentación para cada una indicando sus condiciones [6]. El algoritmo empleado para este método se muestra a continuación:

1.  $s_u = C_T / u$
2.  $C_u = C(1:s_u)$
3.  $C = C(s_u:N)$
4. *Volver al paso 2*

donde  $S_u$  es el número de subportadoras por usuario,  $C_T$  es el número de subportadoras disponibles,  $u$  es el número de usuarios,  $C$  es el conjunto de subportadoras disponibles el cual es inicializado como  $[1, \dots, N]$  y  $C_u$  es el conjunto de subportadoras asignadas al usuario [1].

Como se muestra en el algoritmo, el primer paso es encontrar el número de subportadoras que se asignan a cada usuario. Segundo se toma el subconjunto de  $s_u$  subportadoras del conjunto  $C$  y se le asignan al usuario  $u$ . En el paso 3 se retira el subconjunto de subportadoras previamente asignadas, actualizando el valor del conjunto  $C$  y finalmente se repite el proceso hasta terminar la asignación a todos los usuarios.

En DSM las subportadoras se asignan de manera aleatoria como se muestra en la Figura 1.8 b). Al aumentar el número de usuarios, aumenta la probabilidad de que las subportadoras no se encuentren correlacionadas debido a la separación de las mismas por lo que existe diversidad en frecuencia. Esto se puede aprovechar cuando los datos que se envían están codificados e intercalados. De esta manera se evitan errores por ráfaga, facilitando la recuperación de la información [6]. El algoritmo empleado en este método se muestra a continuación:

1.  $s_u = C_T / u$
2.  $C_u = \text{randslct}(s_u, C)$
3.  $C = C - C_u$
4. *Volver al paso 2*

donde  $\text{randslct}(s_u, C)$  selecciona un subconjunto de  $s_u$  subportadoras del conjunto  $C$  y se asigna al usuario  $u$  [1].

Como se muestra en el algoritmo, el primer paso es encontrar el número de subportadoras que se asignan a cada usuario. Segundo se toma el subconjunto aleatorio de  $s_u$  subportadoras del conjunto  $C$  y se le asignan al usuario  $u$ . En el paso 3 se retira el subconjunto de subportadoras previamente asignadas, actualizando el valor del conjunto  $C$  y finalmente se repite el proceso hasta terminar la asignación a todos los usuarios.

En el método de subportadoras alternadas se asignan de manera alternada como se muestra en la Figura 1.8 c) utilizando el algoritmo que se muestra a continuación:

1.  $s_u = C_T / u$
2.  $C_u = \text{altslct}(s_u, C, i, p)$
3.  $i = i + +$
4. *Volver al paso 2*

donde  $\text{altslct}(s_u, C, i, p)$  selecciona un subconjunto alternado de  $s_u$  subportadoras del conjunto  $C$  y se asigna al usuario  $u$ . El parámetro  $p$  establece el tamaño del salto, el cual se configura dependiendo del número de usuarios y el parámetro  $i$  establece el inicio del subconjunto [1].

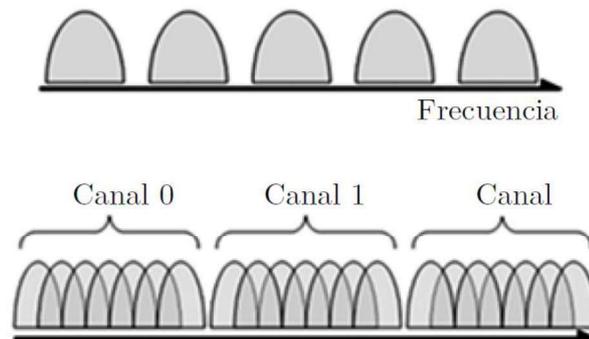
Como se muestra en el algoritmo, el primer paso es encontrar el número de subportadoras que se asignan a cada usuario. Segundo se crea el subconjunto de  $s_u$  subportadoras del conjunto  $C$  para asignarlo al usuario  $u$ . Dicho subconjunto sigue una progresión aritmética que empieza con la  $i$  –ésima subportadora del conjunto  $C$  y con una distancia  $p$ . En el paso 3 se aumenta el valor de  $i$  para proseguir con el siguiente usuario. El valor de  $i$  va de 1 hasta  $u$  y finalmente se repite el proceso hasta terminar la asignación a todos los usuarios.

### **1.3.2.2 VENTAJAS Y DESVENTAJAS DE OFDMA**

Como se mencionó anteriormente, la modulación OFDM presenta un gran número de ventajas lo que ha permitido un gran desarrollo de las comunicaciones, especialmente en el ámbito inalámbrico. A su vez OFDMA presenta varias ventajas que se muestran a continuación:

- Gracias a la inserción del prefijo cíclico OFDMA es muy robusta frente a la ISI ocasionada por la propagación multitrayecto.
- La implementación de ecualizadores en dominio de la frecuencia son más eficientes y menos complejos que los ecualizadores en el dominio temporal [2].

- OFDMA tiene una alta eficiencia espectral en comparación tanto a sistemas multiportadora como a sistemas con una sola portadora. Esto se logra con el uso de subportadoras ortogonales las cuales no requieren de bandas de guarda y permiten agrupar un mayor número de las mismas en un ancho de disponible como se muestra en la Figura 1.9 [2].



**Figura 1.9.** Distribución de subportadoras versus subportadoras ortogonales [2].

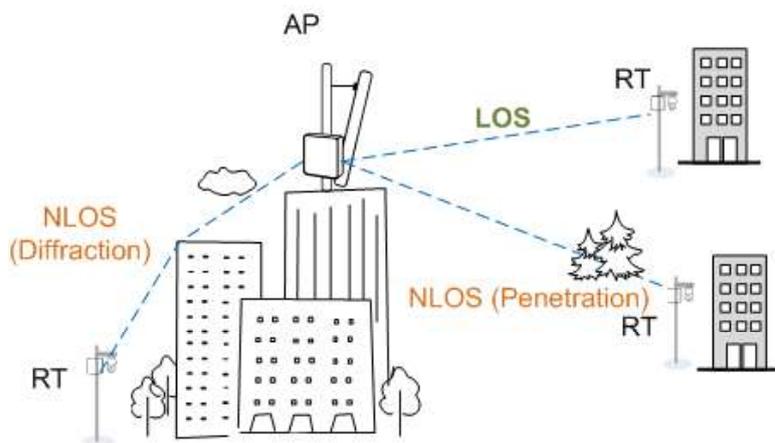
- Se pueden asignar las subportadoras de manera dinámica, lo que permite asignar un número de subportadoras mayor o menor a un usuario en función de los requerimientos del servicio.
- Al dividir el ancho de banda disponible en un elevado número de subportadoras se consigue una elevada granularidad al momento de asignar más o menos recursos a cada usuario. Esto resulta de suma utilidad al momento de acomodar los servicios con diferentes requerimientos de calidad [2].

Una de las desventajas de OFDMA es que presenta sensibilidad frente a errores de frecuencia. Si existe un desplazamiento de frecuencia se puede llegar a perder la ortogonalidad de las subportadoras, ocasionando interferencia entre las mismas. El principal causante de este problema es el efecto *Doppler* asociado al desplazamiento de los receptores.

### 1.3.3 CANAL INALÁMBRICO

En el canal inalámbrico la señal se propaga por el aire en un ancho de banda determinado. El canal inalámbrico es dinámico y variante en el tiempo y debido a esto la señal es muy susceptible a daños durante su propagación, por lo que las prestaciones de la señal se ven reducidas. Además, la señal transmitida viaja por diferentes trayectorias lo que genera ISI (Intersymbol Interference) en el receptor y en consecuencia existe pérdida de datos. Otra dificultad que experimenta la señal es el ruido generado por interferencias o el ruido térmico así como el desvanecimiento del trayecto para el cual se emplea un

modelo de desvanecimiento estadístico ya sea de tipo *Rayleigh* si no tiene línea de vista (NLOS, *Non Line of Sight*) o el modelo *Rice* si existe línea de vista (LOS, *Line of Sight*) como se muestra en la Figura 1.10 [5].



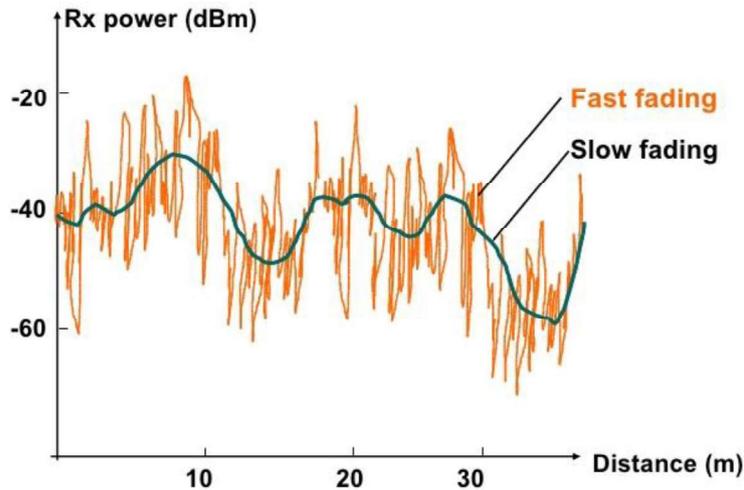
**Figura 1.10.** NLOS y LOS [5].

En el caso NLOS la señal llega al receptor penetrando los obstáculos o por medio del fenómeno de la difracción el cual produce una desviación del rayo cuando pasa por un cuerpo opaco o por una abertura de diámetro menor o igual a la longitud de onda como se muestra en la Figura 1.10 [5].

### 1.3.3.1 Desvanecimiento

El desvanecimiento provoca la reducción de la intensidad de la señal cuando esta atraviesa ciertos obstáculos. El grado del impacto que recibe la señal dependerá de su frecuencia ya que a mayores frecuencias, obstáculos tan diminutos como gotas de agua pueden afectar el nivel de la señal.

Existen diferentes tipos de desvanecimiento, por ejemplo en la Figura 1.11 se muestra el desvanecimiento rápido (*Fast Fading*) el cual se origina principalmente por una alta presencia del efecto Doppler y si el tiempo de coherencia es menor a la duración del símbolo. Además del desvanecimiento rápido existe el desvanecimiento lento (*Slow Fading*) el cual atenúa la señal debido a los múltiples obstáculos existentes en la trayectoria como árboles o edificios [10].



**Figura 1.11.** Desvanecimiento rápido y lento [10].

Existe otro tipo de desvanecimiento el cual se denomina desvanecimiento plano que se obtiene cuando el ancho de banda de coherencia es mayor al ancho de banda de la señal. Si se cumple dicha condición, la señal se ve atenuada por una misma magnitud a lo largo de todo su espectro. Si la señal sufre diferentes niveles de atenuación durante su transmisión, se lo conoce como desvanecimiento selectivo en frecuencia y éste se origina debido a que el ancho de banda de coherencia es menor al ancho de banda de la señal.

### 1.3.3.2 Power Delay Profile

En canales multitrayecto cada versión de la señal llega con tiempos diferentes y aleatorios lo que genera copias innecesarias en el receptor. Estos tiempos se denominan como retardos de propagación y sus parámetros son definidos en el perfil de retardo de potencia (PDP, *Power Delay Profile*). El PDP describe la distribución temporal de la potencia de una onda tipo impulso propagada sobre el canal dado por la Ecuación 1.6 que define la respuesta impulsiva del canal con multitrayectoria

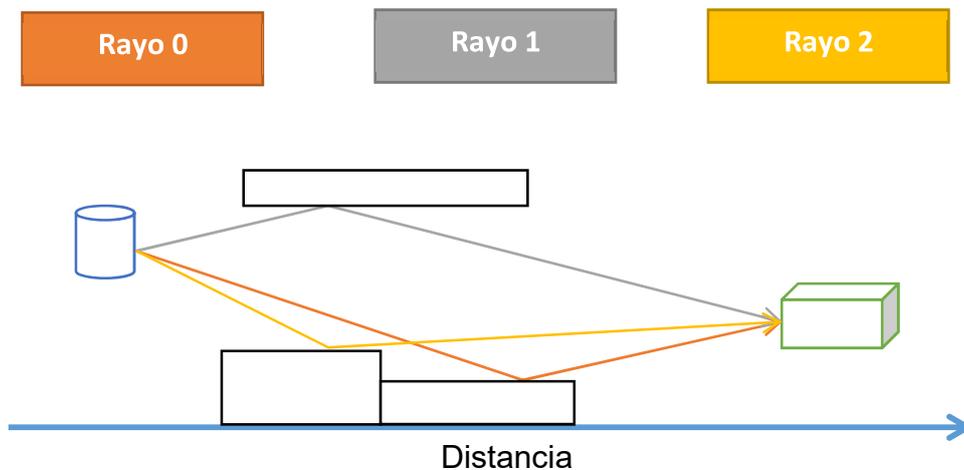
$$h(\tau; t) = \sum_r \beta_r(t) e^{j\theta_r(t)} \delta(t - \tau_r) \quad 1.6$$

donde  $r$  es el índice del trayecto,  $\beta_r(t)$  es la ganancia del canal,  $\theta_r(t)$  es el cambio de fase,  $\tau_r$  el retardo del trayecto número  $r$  y  $\delta$  es la función impulsiva delta de Dirac con lo que el PDP se puede representar como se muestra en la Ecuación 1.7 [3].

$$P(\tau; t) = |h(\tau; t)|^2 = \sum_r \beta_r(t)^2 \delta(t - \tau_r) \quad 1.7$$

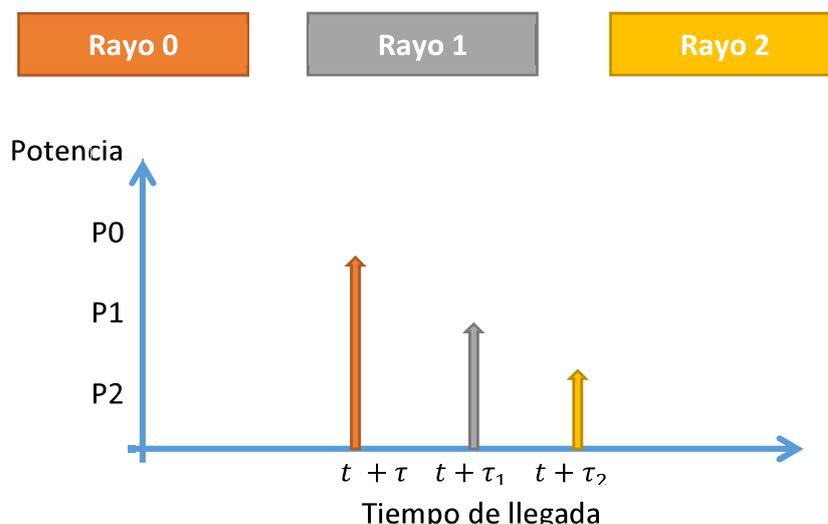
Como se muestra en la Ecuación 1.7 el PDP se define como el valor absoluto al cuadrado de la respuesta impulsiva del canal y este representa la potencia relativa recibida en base a las múltiples copias y sus retardos respecto a la primera trayectoria [3].

Para comprender el PDP, se considera el escenario de la Figura 1.12 en la cual se observa la transmisión sobre un medio inalámbrico en la cual el cilindro representa el transmisor y el cubo el receptor. Los cuadrados negros representan obstáculos durante la trayectoria y tanto el rayo 1 y 2 son copias del rayo 0 que recibe el transmisor debido al canal multitrayecto.



**Figura 1.12.** Escenario multitrayecto.

En el receptor las copias llegan con una variación tanto de potencia como de tiempo respecto al rayo principal como se muestra en la Figura 1.13. La diferencia de tiempo corresponde al retardo de propagación  $\tau_n$ .



**Figura 1.13.** Tiempos y potencias de los rayos en el receptor.

Una vez establecidos los parámetros del escenario se puede aplicar la Ecuación 1.7 para obtener la potencia relativa recibida. Dichos parámetros se pueden obtener mediante mediciones o con el uso de modelos establecidos

### 1.3.4 ANÁLISIS DE WATER-FILLING

Los problemas de optimización consisten en encontrar los máximos o mínimos de funciones con múltiples variables, las cuales se encuentran sujetas a varias condiciones. Una forma de resolver este tipo de problema es el método de los multiplicadores de Lagrange el cual consiste en reducir el problema restringido de  $n$  variables a uno sin restricciones de  $n + k$  variables, en el que  $k$  es el número de restricciones. Una generalización del método de Lagrange es conocida como las condiciones de Karush-Kuhn-Tucker (KKT) cuya función objetivo y restricciones se muestran en la Ecuación 1.8 [12].

$$\begin{aligned} & \min f(x) \\ \text{sujeto a: } & g_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_j(x) = 0, \quad j = 1, \dots, l \end{aligned} \quad (1.8)$$

Donde  $f(x)$  es la función objetivo,  $g_i(x)$  son las restricciones de desigualdad,  $h_j(x)$  son las restricciones de igualdad,  $m$  y  $l$  el número de restricciones de desigualdad e igualdad, respectivamente [12].

Este tipo de problemas puede describirse como verter un volumen limitado de agua (*water-filling*) en un tanque, cuyo fondo tiene los niveles de escalera determinados por el inverso de las ganancias del subcanal. Para este caso se considera una potencia total (volumen total de agua)  $P > 0$ , la potencia asignada para cada canal  $P_i$ , la variancia del ruido AWGN (*Additive white Gaussian noise*)  $\sigma^2$  y la ganancia del trayecto de cada canal  $a_i$ , para  $i = 1, 2, \dots, k$ , donde  $k$  es el número total de canales. La función objetivo es la ecuación de la capacidad del canal descrita por Shannon (Ecuación 2.5).

La ecuación de Shannon es una función logarítmica, la cual es una función cóncava y para poder aplicar las condiciones KKT se debe cumplir que la función objetivo sea convexa, lo cual se consigue al minimizar la sumatoria propuesta cambiando de signo a la misma. Una vez aplicado el cambio se establecen las condiciones KKT dando como resultado un problema de optimización convexo como se describe en la Ecuación 1.9.

Como se puede observar en la Ecuación 1.9, las potencias de los canales no pueden ser negativas y la suma de las mismas debe ser igual a la potencia total [3].

$$\begin{aligned}
\min \quad & - \sum_{i=1}^k \log \left( 1 + \frac{P_i a_i}{\sigma^2} \right) \\
\text{sujeto a:} \quad & P_i \geq 0 \quad \forall i \\
& \sum_{i=1}^k P_i = P
\end{aligned} \tag{1.9}$$

Una vez descrita la función objetivo y las condiciones KKT se utilizan los multiplicadores de Lagrange dando como resultado la Ecuación 1.10 que representa el problema de optimización binario [3]:

$$\begin{aligned}
P_i &= \left( \frac{1}{\mu} - \frac{\sigma^2}{a_i} \right)^+, \quad \text{para } i = 1, \dots, k \\
\mu &\geq 0
\end{aligned} \tag{1.10}$$

Donde  $\mu$  es el nivel de agua escogido para satisfacer la restricción de suma de potencia dada por la Ecuación 1.9, considerando que  $x^+$  es igual a  $x$  si  $x \geq 0$  y 0 si  $x < 0$ . La solución es conocida como la solución del problema de *Water-Filling* [3].

### 1.3.5 GEOMETRIC WATER-FILLING

En la Figura 1.14 (a) se representa en enfoque geométrico del algoritmo *Water-Filling*. En este caso se consideran 4 escalones ( $K = 4$ ) con un mismo ancho dentro de un tanque de agua. La línea punteada es el nivel de agua  $\mu$ , el cual debe ser determinado para la asignación de potencia de cada canal [3].

La profundidad del  $i$ -ésimo escalón ( $d_i$ ) se determina por la Ecuación 1.11:

$$d_i = \frac{\sigma^2}{a_i}, \quad \text{para } i = 1, \dots, K \tag{1.11}$$

La secuencia  $a_i$  se encuentra ordenada de manera decreciente, por lo que la profundidad de los escalones será creciente. También se define  $\delta_{i,j}$  como la diferencia de profundidad de paso del  $i$ -ésimo escalón y se expresa con la Ecuación 1.12.

$$\delta_{i,j} = a_i - a_j = \frac{\sigma^2}{a_i} - \frac{\sigma^2}{a_j}, \quad \text{donde } i \geq j \text{ y } 1 \leq i, j \leq K \tag{1.12}$$

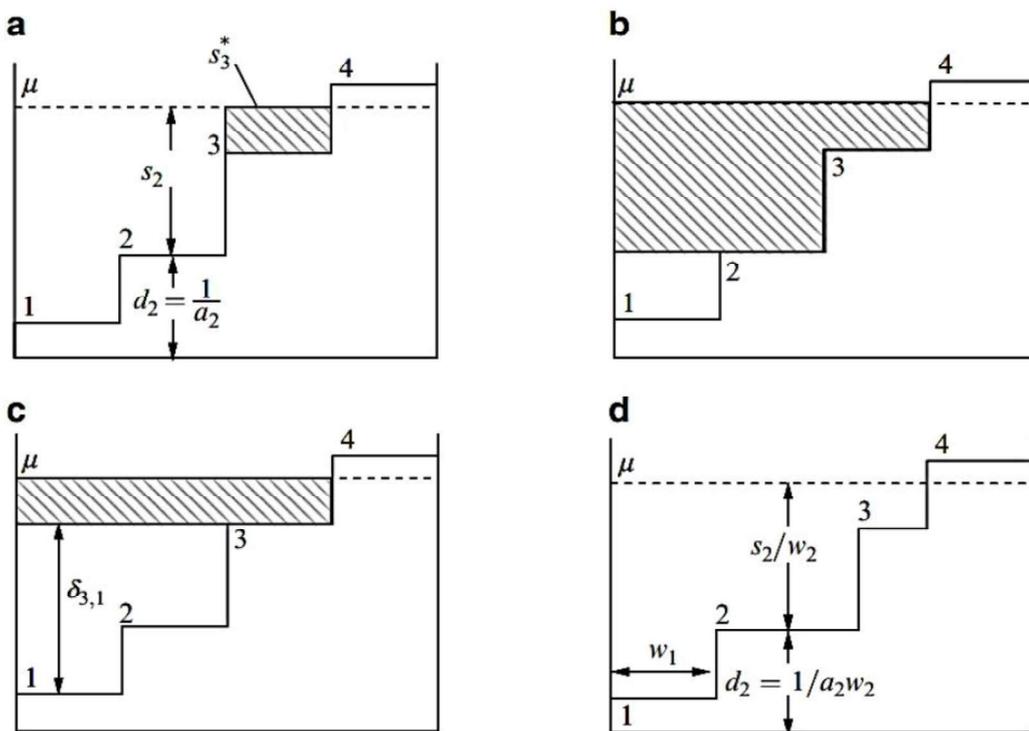
En lugar de determinar el nivel de agua  $\mu$ , *Geometric Water-Filling* determina el nivel de paso que es un número entre 1 y  $K$  representado por  $k^*$ . De esta manera se determina la

asignación de potencia a cada canal. En la Figura 1.14 (a) se ilustra el concepto de  $k^*$  donde el mayor nivel bajo el agua es el tercer canal, por lo tanto  $k^* = 3$ . El área sombreada indica la potencia asignada ( $s_3^*$ ) a ese escalón [3].

Al no conocer el nivel de agua  $\mu$  se considera un nivel de agua sobre el paso  $k$  descrito como  $P_2(k)$ , si el nivel es mayor al considerado entonces  $P_2(k) = 0$ .  $P_2(k)$  puede ser determinado restando el volumen total de agua bajo el paso  $k$  de la potencia total  $P$  como se describe en la Ecuación 1.13 [3].

$$P_2(k) = \left\{ P - \left[ \sum_{i=1}^{k-1} \left( \frac{\sigma^2}{a_i} - \frac{\sigma^2}{a_j} \right) \right] \right\}^+$$

$$P_2(k) = \left[ P - \left( \sum_{i=1}^{k-1} \delta_{i,j} \right) \right]^+, \quad \text{para } k = 1, \dots, K \quad (1.13)$$



**Figura 1.14.** Representación de *Geometric-Water-Filling*. (a) asignación de potencia para el tercer escalón; (b) nivel de agua (potencia) sobre el paso  $k=2$ ; (c)  $P_2(k)$  cuando  $k=3$ ; (d) ilustración para casos ponderados [3].

En la Figura 1.14 los ejemplos (b) y (c), muestran el concepto de  $P_2(k)$  para  $K = 2$  y  $K = 3$ , respectivamente. En el ejemplo de la Figura 1.14 (c) el nivel de agua bajo el tercer

paso puede ser expresado como la suma de dos términos: la diferencia entre la profundidad del tercer y primer paso ( $\delta_{3,1}$ ), y la diferencia entre la profundidad del tercer paso y el segundo paso ( $\delta_{3,2}$ ). De esta manera a  $P_2(k) = 3$  puede ser descrito como se menciona en la Ecuación 1.14 [3]:

$$P_2(k = 3) = (P - \delta_{3,1} - \delta_{3,2})^+ \quad (1.14)$$

## 2. METODOLOGÍA

En el siguiente capítulo se realiza la descripción de la metodología empleada para el desarrollo del sistema OFDM, incluyendo la implementación de cada algoritmo de asignación de recursos, así como el cálculo de la capacidad total y por usuario del sistema. Los diagramas de bloques propuestos brindan una visión general sobre el funcionamiento del sistema OFDMA implementado y están acompañados del código desarrollado.

El código diseñado permite obtener la capacidad total y por usuario de un sistema OFDMA. Al inicio del código se configura el número de usuarios entre 2 y 4 así como el ancho de banda en función del estándar 802.11n, el cual especifica 2 tipos de canales, el primero con un ancho de banda de 20 MHz dividido en 64 subportadoras y el segundo con un ancho de banda de 40 MHz con 128 subportadoras. El estándar puede utilizar QPSK, 16-QAM o 64-QAM como esquema de modulación. Dichas modulaciones son implementadas en el código y evaluadas por separado para varios SNR. El medio de transmisión se implementa con una distribución de tipo Rayleigh y un PDP diferente para cada usuario. El símbolo que se transmite sobre el canal contiene la información de los diferentes usuarios la cual es asignada de acuerdo a los métodos mencionados en la sección 1.3.2.1. Una vez obtenido el símbolo OFDMA se utiliza el algoritmo *geometric water-filling* para reasignar los recursos de cada subportadora para los 3 tipos de asignaciones. Finalmente, se calcula la capacidad con el símbolo OFDMA y el estado del canal. Dichas variables se guardan para su futuro análisis mediante graficas de capacidad versus SNR.

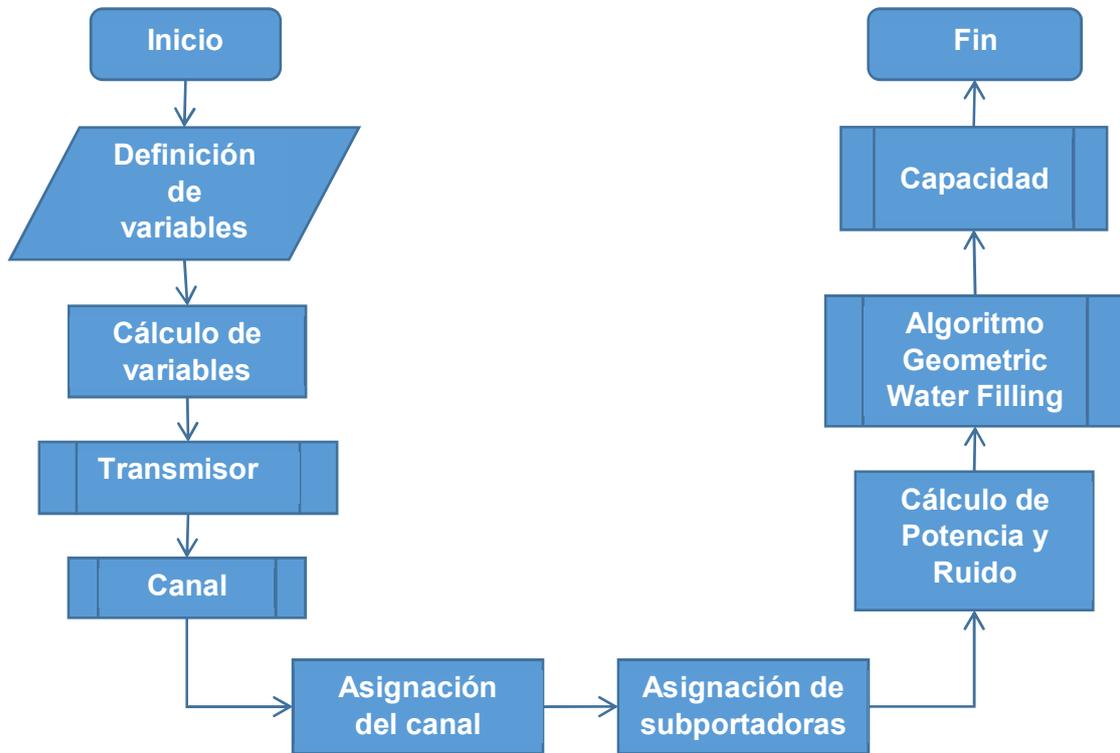
### 2.1 PROGRAMA PRINCIPAL

En la Figura 2.1 se presenta el diagrama de bloques del programa principal. El diagrama muestra los procesos del sistema OFDMA utilizados para determinar la eficiencia de los métodos de asignación de subportadoras, así como la maximización de la capacidad con el uso del algoritmo *geometric water filling*.

En el primer bloque, definición de variables, se establecen los parámetros de la capa física del estándar 802.11n así como el número de iteraciones del código y los valores de SNR sobre los cuales se evalúa el sistema OFDMA. El estándar 802.11n establece el ancho de banda del canal, esquema de modulación y la duración del prefijo cíclico como se muestra en la Tabla 2.1.

El bloque de cálculo de variables permite asegurar que se cumpla el estándar 802.11n. Se establecen condiciones respecto al ancho de banda total del canal ya que para cada

caso se tiene un valor fijo de subportadoras totales, piloto y de datos, así como el ancho de banda de cada subportadora.



**Figura 2.1.** Diagrama de bloques del programa principal.

El bloque del transmisor se encarga de construir el símbolo OFDM de cada usuario. El proceso inicia con la generación de los bits de cada usuario y su modulación de acuerdo al esquema establecido. Con los datos modulados se procede a construir el símbolo OFDM, asignando la información a las subportadoras de datos correspondientes, así como la configuración de las subportadoras piloto y las subportadoras null.

En el bloque asignación del canal se selecciona los subcanales sobre los cuales se transmite la información y se los agrupa en una solo vector para el cálculo de la capacidad del sistema.

El bloque de asignación de subportadoras se encarga de crear el símbolo OFDMA, el cual toma ciertas subportadoras del símbolo OFDM de cada usuario y las unifica en el símbolo OFDMA.

Cuando se obtiene el símbolo OFDMA se calcula la potencia del mismo así como el ruido presente en función del SNR. Esta acción se realiza en el bloque de cálculo de potencia y ruido.

En el bloque del algoritmo *geometric water-filling* se calcula la nueva potencia de las subportadoras del símbolo OFDMA en función de los valores obtenidos en el bloque previo.

Con los datos de la potencia, ruido y potencia asignada por el algoritmo *geometric water-filling* se calcula la capacidad del sistema en el bloque de capacidad.

**Tabla 2.1.** Parámetros de la capa física del estándar 802.11n.

Ancho de Banda del Canal [MHz]	Modulación	Subportadoras			Duración del Prefijo Cíclico
		Totales	Piloto	Datos	
20	QPSK	64	4	52	¼ de la duración del símbolo OFDM
	16-QAM				
	64QAM				
40	QPSK	128	6	108	¼ de la duración del símbolo OFDM
	16-QAM				
	64QAM				

Los parámetros detallados en la Tabla 2.1 se definen en las variables que se muestran en la Tabla 2.2, así como otras variables de inicialización.

**Tabla 2.2.** Definición de variables.

Variables	Definición
Nsubt	Número de subportadoras totales
usuarios	Número de usuarios
AB	Ancho de banda del canal
mod	Modulación utilizada
rep_snr	Número de iteraciones
ig	Intervalo de guarda
SNR	Signal to Noise Ratio en dB
Norm	Normalización de la modulación
Nsubd	Número de subportadoras de datos
AB_Sub	Ancho de banda por subportadora
M	Orden de la modulación

A continuación se detalla el código implementado para el diagrama de bloques de la Figura 2.1, código que se encuentra en el anexo A.

Al inicio del programa principal se encuentra la configuración de las variables en función de la Tabla 2.1 y 2.2. Las variables que se deben configurar para evaluar los diferentes escenarios son número de usuarios, la cual puede tomar un valor de 2 o 4, y el ancho de

banda del canal, entre 20 MHz y 40 MHz. Dichas variables se configuran en las líneas 6 y 7 del segmento de código 2.1. Las líneas 9 y 11 se utilizan para la modulación y se detallan en la sección 2.2.

```

1.  %-----PROGRAMA PRINCIPAL-----
6.  usuarios=2;      % Numero de usuarios  2 o 4
7.  AB=20e6;        % Ancho de Banda 20MHz o 40 MHz
8.  mod=[2 4 6];    % Bits por simbolo para cada modulacion
9.                  % Modulacion QPSK(2), 16QAM(4) O 64QAM(6)
10. norm=[sqrt(2) sqrt(10) sqrt(42)];
11. rep_snr=5000;   % Repeticiones para cada SNR
12. ig=1/4;        % Intervalo de Guarda
13. SNR=0:1:20;    % SNR
14. if AB==20e6
15.     Nsubt=64;
16.     Nsubd=52;
17.     AB_Sub=AB/Nsubt;
18. elseif AB==40e6
19.     Nsubt=128;
20.     Nsubd=108;
21.     AB_Sub=AB/Nsubt;
22. end

```

#### Segmento de código 2.1. Parámetros iniciales.

Las líneas 14 a la 21 permiten establecer el número de subportadoras totales ( $N_{subt}$ ), de datos ( $N_{subd}$ ) y el ancho de banda de cada subportadora ( $AB\_Sub$ ) en función del ancho de banda del canal establecido por el usuario. El programa principal se repetirá  $rep\_snr$  veces para cada valor de SNR y a su vez este proceso se realizará para cada modulación como se muestra en el segmento de código 2.2.

```

24. for modlc=1:3
25. m=mod(modlc);
26. M=2^m;
27. n=norm(modlc);
28.     %Selección de SNR
29.     for iter=1:length(SNR)
30.         Snr=SNR(iter);
31.         snr_veces=10^(Snr/10);
32.         for rep=1:rep_snr

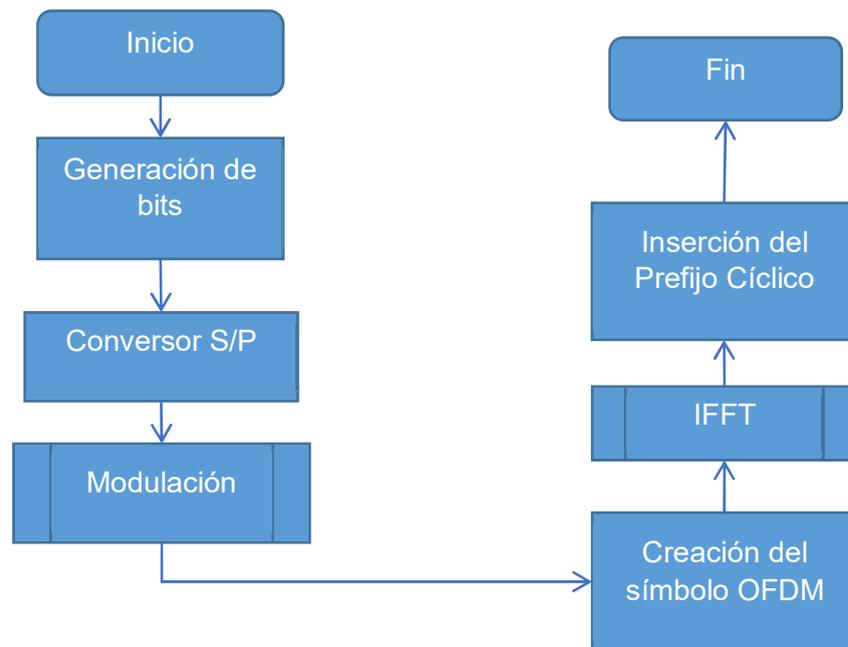
```

#### Segmento de código 2.2. Número de repeticiones del programa principal.

Dentro del programa principal se utilizan múltiples funciones creadas para diferentes partes del sistema OFMDA. Dichas funciones se encuentran detalladas en las próximas secciones.

## 2.2 TRANSMISOR

La Figura 2.2 representa el diagrama de bloques del transmisor implementado. El primer bloque se encarga de generar el conjunto de bits que cada usuario transmite. El segundo bloque realiza la conversión de serie a paralelo de la serie de bits para su futura modulación. El bloque de modulación, como su nombre lo indica, se encarga de aplicar el esquema de modulación correspondiente ya sea QPSK, 16-QAM o 64-QAM. En el siguiente bloque se construye el símbolo OFDM, distribuyendo los datos modulados y las señales piloto a su correspondiente subportadora según el estándar 802.11n, como se muestra en la Tabla 2.4 y en la Figura 2.3. Una vez obtenido el símbolo OFDM se realiza la IFFT en su correspondiente bloque. En el último bloque se realiza la inserción del prefijo en el intervalo de guarda.



**Figura 2.2.** Diagrama de bloques del transmisor (Función TX).

Dentro del programa principal se llama a la función TX para crear el símbolo que cada usuario transmite. Como se muestra en el segmento de código 2.3, en la línea 36 se verifica el número de usuarios de la simulación y en función de dicho número genera los símbolos para cada uno. En la línea 34 se observa que el símbolo del primer usuario se guarda en la variable `Sym_OFDMA_tx_1`, mientras que en la variable `Sym_D_1` se guarda la parte del símbolo que contiene los datos, es decir se excluye las subportadoras piloto y null. El símbolo OFDM se construye en función de los parámetros indicados en la Tabla 2.3 y este proceso se repite para el número de usuarios como muestra en las líneas 34 y 35 para 2 usuarios y en las líneas 37 y 38 si se trabaja con 4 usuarios.

Como salida la función TX obtiene 2 variables, la primera es el símbolo total a transmitir, incluido el prefijo cíclico y la segunda corresponde a los datos modulados.

```

1.  %-----PROGRAMA PRINCIPAL-----

33. %-----Transmisor-----
34.  [Sym_OFDMA_tx_1, Sym_D_1] = TX(Nsubt, n, m, M, ig, Nsubd);
35.  [Sym_OFDMA_tx_2, Sym_D_2] = TX(Nsubt, n, m, M, ig, Nsubd);
36.  if usuarios==4
37.    [Sym_OFDMA_tx_3, Sym_D_3] = TX(Nsubt, n, m, M, ig, Nsubd);
38.    [Sym_OFDMA_tx_4, Sym_D_4] = TX(Nsubt, n, m, M, ig, Nsubd);
39.  end

```

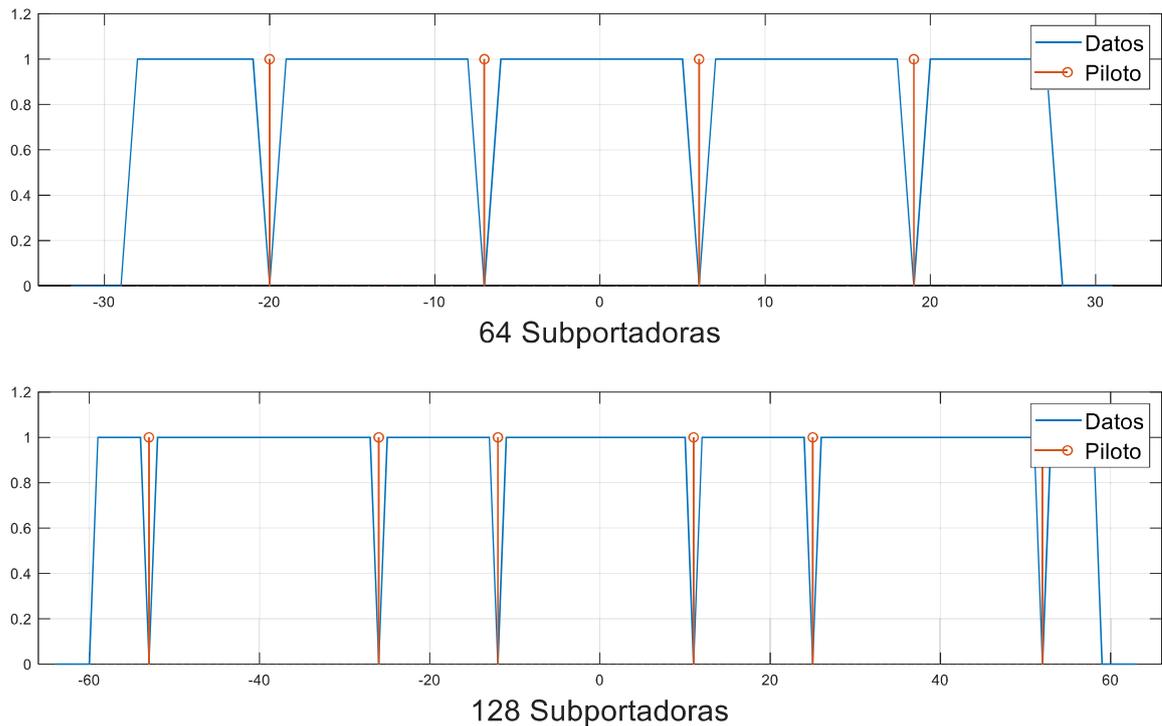
**Segmento de código 2.3.** Llamado de la función TX dentro del programa principal.

**Tabla 2.3.** Variables de entrada y salida para la función TX.

<b>Variables de Entrada</b>	<b>Definición</b>
Nsubt	Número de subportadoras totales
n	Normalización de la modulación utilizada
m	Número bits utilizados por la modulación
M	Tamaño de la constelación de la modulación
ig	Tamaño del intervalo de guarda
Nsubd	Número de subportadoras de datos
<b>Variables de Salida</b>	<b>Definición</b>
Sym_OFDMA_tx_X	Símbolo OFDMA de cada usuario
Sym_D_1	Datos modulados y normalizados

**Tabla 2.4.** Distribución de subportadoras para el estándar 802.11n.

<b>Número total de subportadoras</b>	<b>Posición de las subportadoras</b>		
	<b>Datos</b>	<b>Piloto</b>	<b>Null</b>
64	[-28:-22, -20:-8, -6:-1] [1:6, 8: 20, 22: 28]	-21, -7, 7, 21	[-32: -29] [ 29: 32 ]
128	[-58:-54, -52:-26, -24:-12, -10: -1 ] [1: 10, 12: 24, 26: 52, 54: 58 ]	-53, -25, -11, 11, 25, 53	[-64: -57] [ 57: 64 ]



**Figura 2.3.** Posición de las subportadoras de datos y piloto para el caso de 64 y 128 subportadoras.

A continuación se detalla el código implementado para el diagrama de bloques de la Figura 2.2, código que se encuentra en el anexo B.

El primer paso consiste en generar el conjunto de bits aleatorios, los cuales fueron creados con la función `randi` como se muestra en la línea 3 del segmento de código 2.4. La función `randi` genera una matriz de  $1 \times N_{\text{subd}} \times m$  números enteros pseudoaleatorios uniformemente distribuidos entre 0 y 1. Este vector representa a la serie de bits totales de información de un usuario, serie que se convierte a paralelo, es decir a un vector de dimensiones  $N_{\text{subd}} \times m \times 1$ . Posteriormente se realiza la modulación y normalización de los datos con el uso de la función `qammod` y dividiendo para el factor de normalización como se muestra en la línea 7 y 8 del segmento de código 2.4.

La función `qammod` realiza la modulación de los datos en función de la constelación  $M$  que se desea utilizar. El parámetro `InputType` permite especificar el tipo de variable de entrada, es decir los datos que se van a modular. Cuando se especifica el tipo de entrada como *bit* la señal de entrada debe tener números binarios y ser un conjunto múltiplo de  $\log_2(M)$ , razón por la cual el conjunto de entrada tiene una dimensión de  $N_{\text{subd}} \times m \times 1$ , donde  $m = \log_2(M)$ , garantizando que los datos entrada puedan ser utilizados por la función `qammod`.

```

1. function [Sym_OFDMA_tx, Sym_D] = TX(Nsubt,n,m,M,ig,Nsubd)
2. %Datos
3. Datos_tx=randi([0 1], 1, Nsubd*m);
4. %Conversion serie paralelo
5. datos_tx_par=reshape(Datos_tx,Nsubd*m,1);
6. %Modulacion y normalizacion
7. datos_tx_mod_nn=qammod(datos_tx_par, M,'InputType','bit');
8. datos_tx_mod=datos_tx_mod_nn./n;
9. Sym_D=datos_tx_mod';

```

#### Segmento de código 2.4. Generación y modulación de los datos a transmitir.

Para crear del símbolo OFDM se utiliza la Tabla 2.4 para la distribución de las subportadoras. En el caso de utilizar 64 subportadoras, en la línea 11 del segmento de código 2.5 se crea un vector de 64 ceros. En las líneas 14 a la 17 se agregan las subportadoras piloto, las cuales tienen una potencia fija y corresponden a los espacios -21, -7, 7 y 21. En las líneas 19 a la 24 se agregan los datos a las subportadoras correspondientes que son de la -28 a la -22, de la -20 a la -8, de la -6 a la -1, de la 1 a la 6, de la 8 a la 20 y de la 22 a la 28.

Si se utiliza 128 subportadoras se realiza un proceso similar, basado en la Tabla 2.4 como se muestra en las líneas 25 a la 41 del anexo B.

```

11. Sym=zeros(Nsubt,1);
12. if Nsubt==64
13. %Senales piloto en las subportadoras -21,-7,21 y7
14.     Sym(12)=1;
15.     Sym(26)=1;
16.     Sym(39)=1;
17.     Sym(53)=1;
18. %Datos en las subportadoras -28 a la -1 y de la 1 a la 28
19.     Sym(5:11)=datos_tx_mod(27:33);
20.     Sym(13:25)=datos_tx_mod(34:46);
21.     Sym(27:32)=datos_tx_mod(47:52);
22.     Sym(33:38)=datos_tx_mod(1:6);
23.     Sym(40:52)=datos_tx_mod(7:19);
24.     Sym(54:60)=datos_tx_mod(20:26);

```

#### Código 2.5. Creación del símbolo OFDM.

A continuación se realiza la IFFT con el uso de la función `ifft`. Dicha función es propia de MATLAB y se encarga de calcular la transformada rápida inversa de Fourier con `Nsubt` puntos. Finalmente se agrega el prefijo cíclico seleccionando la cuarta parte del

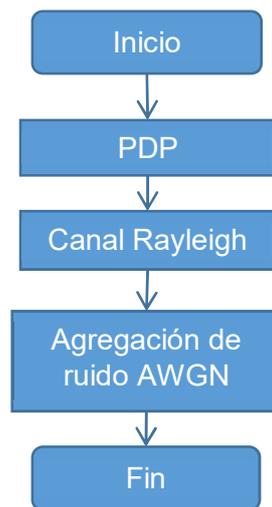
símbolo y colocándola al inicio del mismo como se muestra en las líneas 44 y 47 del segmento de código 2.6.

```
44. Sym_ifft=ifft(Sym,Nsubt);  
47. Sym_OFDMA=[Sym_ifft(Nsubt-Nsubt*ig+1:end), Sym_ifft];
```

**Código 2.6.** IFFT y agregación del prefijo cíclico.

## 2.3 CANAL

En la Figura 2.4 se presenta el diagrama de bloques del canal implementado. En el primer bloque se configuran los parámetros del PDP detallados en la sección 1.3.3.2. En el bloque Canal Rayleigh se modela el desvanecimiento selectivo en frecuencia con una distribución de tipo Rayleigh. Finalmente en el último bloque se agrega el ruido AWGN al canal en función de la SNR configurada.



**Figura 2.4.** Diagrama de bloques de la función CANAL\_X.

A continuación se detalla el código implementado para simular el canal, el cual se encuentra en el anexo C para el primer usuario, anexo D para el segundo usuario, anexo E para el tercer usuario y anexo F para el cuarto usuario.

Dentro del programa principal se llama a la función CANAL\_X, donde X puede ser 1, 2, 3 o 4 dependiendo del usuario al cual corresponda. Se utiliza canales diferentes ya que los usuarios se encuentran en distintas posiciones por lo que cada uno experimenta un canal diferente.

Para encontrar la respuesta del canal para cada usuario, se envían los parámetros establecidos en la Tabla 2.5 a la función CANAL\_X como se muestra en el segmento de

código 2.7 en las líneas 41, 42 para 2 usuarios y adicionalmente las líneas 44 y 45 en el caso de 4 usuarios.

```

40. %-----Canal-----
41.     H_D_1 = CANAL_1(Nsubt, Sym_OFDMA_tx_1, Snr);
42.     H_D_2 = CANAL_2(Nsubt, Sym_OFDMA_tx_2, Snr);
43.     if usuarios==4
44.         H_D_3 = CANAL_3(Nsubt, Sym_OFDMA_tx_3, Snr);
45.         H_D_4 = CANAL_4(Nsubt, Sym_OFDMA_tx_4, Snr);
46.     end

```

**Segmento de código 2.7.** Creación de los canales para cada usuario.

**Tabla 2.5.** Variables de entrada y salida para la función CANAL\_X.

Variables de Entrada	Definición
Nsubt	Número de subportadoras totales
Sym_OFDMA_tx	Símbolo OFDMA que se transmite sobre el canal
Snr	SNR utilizado
Variables de Salida	Definición
H_D_X	Subcanales de datos del usuario X

Dentro de la función CANAL\_X se establecen los parámetros del PDP, los cuales se muestran en el segmento de código 2.8.

```

1. function [H_D] = CANAL_1(Nsubt, Sym_OFDMA_tx, Snr)
2. %Parametros del canal multitrayecto (PDP)
3. p_rayos = [0 -3 -6]; % Potencia de los rayos en dB
4. retardos= [0 6 12]; % Retardos de los rayos
5. N_rayos=length(retardos); % Numero de rayos
6. p_rayos_veces = 10.^(p_rayos/10); % Potencia en veces
7. p_pm=p_rayos_veces/sum(p_rayos_veces); % Potencia promedio
8. p_pm_n=sqrt(p_pm); % Potencia promedio normalizada

```

**Segmento de código 2.8.** Implementación del PDP.

En la línea 3 del segmento de código 2.8 se establece el valor de la potencia en dB del rayo inicial así como de su copias, mientras que en la línea 6 se convierten los dB a veces. En la línea 4 se define el retardo del rayo inicial y de sus copias, es decir la diferencia de tiempo respecto al primer rayo. En la línea 5 se calcula el total de rayos que llegan al usuario. En la línea 7 y 8 se obtiene la potencia promedio de los rayos

normalizados, la cual se utilizará para asignar la potencia al canal con desvanecimiento de *Rayleigh* como se muestra en la línea 11 del segmento de código 2.9.

```
1. function [H_D] = CANAL_1(Nsubt, Sym_OFDMA_tx, Snr)
9. %Canal Rayleigh
10. canal=(randn(1,N_rayos)+i*randn(1,N_rayos))./sqrt(2);
11. h_t=canal.*p_pm_n;
12. h=zeros(1,Nsubt);
13. h(1,retardos+1)=h_t; % Canal
14. Sym_OFDMA_h=conv(Sym_OFDMA_tx,h); % Respuesta del Simbolo
OFDMA en el canal
15. H=fft(h,Nsubt); % Respuesta en frecuencia del canal
```

### Segmento de código 2.9. Canal con desvanecimiento tipo *Rayleigh*.

La distribución de *Rayleigh* se representa con números complejos en los cuales sus componentes reales e imaginarias son independientes y siguen una distribución normal, dicha representación se desarrolla en la línea 10 en la cual la distribución normal se realiza con la función *randn* propia de MATLAB. Una vez establecido el canal se obtiene su respuesta impulsiva como se muestra en la línea 13. Para obtener el símbolo transmitido con el efecto del canal, se realiza la convolución entre el símbolo OFDMA y la respuesta impulsiva del canal como se muestra en la línea 14 y finalmente en la línea 34 del anexo C se obtiene el símbolo con ruido AWGN, es decir, el símbolo que viajó sobre el canal y está listo para ser recibido.

Para obtener la capacidad del sistema se utiliza el estado del canal, el ruido y la potencia del símbolo. En las líneas 17 a la 22 del segmento de código 2.10 se crea un vector que contiene el estado del canal sobre el cual viajan solo los datos para el caso de 64 subportadoras. La selección del canal de datos se basa en la Tabla 2.4.

```
1. function [H_D] = CANAL_1(Nsubt, Sym_OFDMA_tx, Snr)
16. if Nsubt==64
17.     H_D(27:33)= H(5:11);
18.     H_D(34:46)= H(13:25);
19.     H_D(47:52)= H(27:32);
20.     H_D(1:6)= H(33:38);
21.     H_D(7:19)= H(40:52);
22.     H_D(20:26)= H(54:60);
```

### Segmento de código 2.10. Canal de datos.

## 2.4 ASIGNACIÓN DE RECURSOS

Como se mencionó en la sección 1.3.3.1 se desarrollan 3 tipos de asignación de subportadoras. La asignación se implementa en las líneas 47 a la 61 para 2 usuarios y en las líneas 62 a la 73 para 4 usuarios dentro del programa principal en el anexo A.

Para realizar la asignación de recursos se utilizan variables que contienen la posición de cada asignación en cada fila para cada usuario. Los variables se guardan con el nombre `as_x`, donde `x` representa cada tipo de asignación. A continuación se detalla la implementación de las asignaciones de recursos para 2 usuarios.

Para la asignación por bloques se crea la variable `as_1` que en la fila número 1 contiene la dirección de la primera mitad de las subportadoras de datos y en la fila número 2 la segunda mitad de las subportadoras de datos como se muestra en la línea 52 del segmento de código 2.11.

La asignación de subportadoras alternadas consiste en tomar 1 subportadora para un usuario y la contigua para el otro usuario, así sucesivamente hasta utilizar el total de subportadoras de datos. Para la asignación alternada se crea la variable `as_2` en la cual, la fila número 1 contiene las direcciones de las subportadoras del primer usuario y la fila número 2 las direcciones del segundo usuario como se muestra en la línea 53 de segmento de código 2.11.

La variable `as_3` utiliza la función `randperm(a,b)` para crear las direcciones de la asignación de subportadoras aleatoria, la cual genera un conjunto de `b` números aleatorios tomados del conjunto `[1: a]`. La razón por la cual se utiliza la función `randperm` es que los números generados no se repiten asegurando de esta manera que no exista superposición de los subcanales. Esta asignación se ve implementada en la línea 54 del segmento de código 2.11.

Una vez creadas las variables de cada tipo de asignación de subportadoras, se construye la información que se transmite en el símbolo OFDMA como se muestra en las líneas 59, 60 y 61. La variable `Sym_D` tiene en cada fila un símbolo OFDMA correspondiente a cada tipo de asignación. Como se mencionó previamente, para el cálculo de la capacidad se necesita del estado del canal correspondiente a cada símbolo. Para obtener dicho estado del canal se utilizan las variables de las direcciones de las asignaciones y en lugar de tomar los datos de los símbolos OFDM, se utilizan los canales de cada usuario como se muestra en las líneas 56, 57 y 58 del segmento de código 2.11 en donde la variable `H`

tiene en cada fila un canal correspondiente al símbolo OFDMA de cada tipo de asignación.

```

47. %-----Asignacion de recursos-----
48. DSM= randperm(Nsubd,Nsubd);
49. H=zeros(3,Nsubd);
50. Sym_D=zeros(3,Nsubd);
51. if usuarios == 2
52.     as_1=[1:Nsubd/2; Nsubd/2+1:Nsubd];
53.     as_2=[1:2:Nsubd; 2:2:Nsubd];
54.     as_3=[DSM(as_1(1,:)); DSM(as_1(2,:))];
55.     %H(asignacion,:)
56.     H(1,:)= [H_D_1(as_1(1,:)) H_D_2(as_1(2,:))];
57.     H(2,:)= [H_D_1(as_2(1,:)) H_D_2(as_2(2,:))];
58.     H(3,:)= [H_D_1(as_3(1,:)) H_D_2(as_3(2,:))];
59.     Sym_D(1,:)= [Sym_D_1(as_1(1,:)) Sym_D_2(as_1(2,:))];
60.     Sym_D(2,:)= [Sym_D_1(as_2(1,:)) Sym_D_2(as_2(2,:))];
61.     Sym_D(3,:)= [Sym_D_1(as_3(1,:)) Sym_D_2(as_3(2,:))];

```

**Segmento de código 2.11.** Asignación de subportadoras para 2 usuarios.

Para el caso de 4 usuarios se crean diferentes variables que contengan las direcciones de los diferentes tipos de asignación de subportadoras como se muestra en las líneas 63 a la 65 del anexo A.

## 2.5 GEOMETRIC WATER FILLING

En la función principal se llama a la función GWF, la cual se utiliza para reasignar la potencia en los canales  $H$  creados en la sección 2.4. La función GWF trabaja con los parámetros establecidos en la Tabla 2.6. Para realizar el cálculo de la nueva potencia del símbolo se requiere de la potencia del símbolo y del estado del canal, las cuales son calculadas en las líneas 76 y 77 del segmento de código 2.12. Debido a que existen 3 tipos de asignaciones, se utiliza un lazo en el que en cada iteración se trabaje con una asignación en específico, dando como resultado la matriz  $PA$ . Las filas de la matriz representan cada asignación mientras que las columnas representan el valor de la nueva potencia.

```

74. %-----GWF-----
75. for t=1:3
76.     Ptx_D(t,:)=sum(abs(Sym_D(t,:).^2));
77.     ruido(t,:)=(Ptx_D(t,:)/snr_veces)/Nsubt;
78.     [PA(t,:)] = GWF(Nsubd, Ptx_D(t,:), H(t,:), ruido(t,:));
79. end

```

**Segmento de código 2.12.** Función GWF dentro del programa principal.

**Tabla 2.6.** Variables de entrada y salida para la función GWF.

<b>Variabes de Entrada</b>	<b>Definición</b>
Nsubd	Número de subportadoras de datos
Ptx_D	Potencia del símbolo OFDMA (solo los datos)
H(t,:)	Canal con el método de asignación número t
ruido	Ruido presente en cada subcanal
<b>Variabes de Salida</b>	<b>Definición</b>
PA(t, :)	Potencia asignada para el método de asignación t

El cálculo de la potencia para cada símbolo OFDMA se realiza con la Ecuación 2.1.

$$P_{datos} = \sum |x|^2 \quad (2.1)$$

donde  $x$  es el valor complejo del símbolo de datos. De igual manera en el segmento de código 2.3 se calcula el ruido existente en cada subcanal con el uso de la Ecuación 2.2.

$$Ruido_T = \frac{P_T}{SNR}$$

$$Ruido_{Subcanal} = \frac{Ruido_T}{Nsubt} \quad (2.2)$$

donde  $Ruido_T$  es la potencia del ruido existente,  $P_T$  es la potencia total del símbolo OFDMA,  $SNR$  es la relación de la señal al ruido,  $Ruido_{Subcanal}$  es la potencia del ruido en cada subcanal y  $Nsubt$  es el número total de subportadoras.

El algoritmo GWF implementado en el anexo G utiliza como parámetros la potencia disponible, el estado del canal sobre el cual se va a transmitir, el número de subportadoras a las cuales se les va a reasignar la potencia y el ruido presente en el canal como se muestra en la Tabla 2.6.

```

1. function [PA] = GWF(Nsubd, Ptx, H_D, ruido)
2. H=abs(H_D);
3. canal_datos=H;
4. B=abs(canal_datos).^2/ruido;
5. [a,I]=sort(B, 'descend');
6. d=1./a;

```

**Segmento de código 2.13.** Inicialización de variables en GWF.

En la línea 4 del segmento de código 2.13 se obtiene la ganancia del canal la cual será utilizada para obtener la función objetivo como se muestra en la Ecuación 1.8. Para

obtener la representación del tanque equivalente a la Figura 1.10, se ordena la ganancia del canal de manera descendente y se obtiene su inversa lo cual se indica en las líneas 5 y 6 dando como resultado la profundidad de los subcanales.

```

1. function [PA] = GWF(Nsubd, Ptx, H_D, ruido)

10. while (P2>0) && (k<=Nsubd)
11.     suma=0;
12.     for i=1:k
13.         suma=suma+(d(k)-d(i));
14.     end
15.     P2=Ptxvc-suma;
16.     if (P2<0)
17.         P2=0;
18.     else
19.         P2k_ast=P2;
20.         k_ast=k;
21.         k=k+1;
22.     end
23. end
24. Sk_ast=P2k_ast/k_ast;
25. S=zeros(1,Nsubd);

26. for i=1:k_ast
27.     S(i)=Sk_ast+d(k_ast)-d(i); %Potencia del escalon determinada
28. end
29. Pasig(I)=S; %Restaura el orden original de las potencias
30. PA=zeros(1,Nsubd);
31. PA= Pasig; %Nueva potencia asignada
32. end

```

#### Segmento de código 2.14. GWF.

Para encontrar el nivel de subportadora se define la diferencia de subcanal como se muestra en la Ecuación 1.12 y un nivel de potencia para cada subportadora  $P_2(k)$  como se muestra en las líneas 11 a la 15 del segmento de código 2.14. Para determinar el nivel de cada subportadora se utiliza la Ecuación 1.13 como se muestra en la línea 27 del segmento de código 2.14. El valor de  $k^*$  se determina cuando el nivel de la subportadora tome un valor negativo, condición establecida en las líneas 16 y 17 del segmento de código 2.14. Cuando se obtiene el valor  $k^*$  se puede calcular la potencia de la subportadora perteneciente al subcanal con el uso de la Ecuación 2.3.

$$s_{k^*} = \frac{P_2(k^*)}{k^*} \quad (2.3)$$

Con  $s_{k^*}$  se puede determinar la potencia de los demás subportadoras con el uso de la Ecuación 2.4.

$$s_i = \begin{cases} s_{k^*} + (d_{k^*} - d_i) & 1 \leq i \leq k^* \\ 0 & k^* < i \leq K \end{cases} \quad (2.4)$$

## 2.6 CAPACIDAD

Dentro del programa principal se llama a la función CAPACIDAD como se muestra en la línea 81 de segmento de código 2.13. Esta función regresa como resultado las capacidades de los usuarios con y sin el uso del algoritmo GWF las cuales se guardan en las variables CAP y CAP\_GWF en función de la modulación y la iteración correspondiente como se muestra en las líneas 82 y 83.

```
80. %-----Capacidad Para Multiples Usuarios-----
81. [cap, cap_gwf] = CAPACIDAD(usuarios,H,Sym_D,ruido,PA,as_1);
82. CAP_GWF(modlc,rep,:)=cap_gwf(:, :, :);
83. CAP(modlc,rep,:)=cap(:, :, :);
```

**Segmento de código 2.13.** Función CAPACIDAD dentro del programa principal.

La función CAPACIDAD da como resultado la capacidad por usuarios en función de las variables de entrada que se muestran en la Tabla 2.7.

**Tabla 2.7.** Variables de entrada y salida para la función CAPACIDAD.

<b>Variables de Entrada</b>	<b>Definición</b>
usuarios	Número de usuarios
H	Canales de datos
Sym_D	Símbolo OFDMA de datos
ruido	Ruido presente en cada subcanal
PA	Potencia asignada por GWF
as_1	Variable de la asignación número 1
<b>Variables de Salida</b>	<b>Definición</b>
cap	Capacidad de los usuarios
cap_gwf	Capacidad de los usuarios con GWF

Dentro de la función CAPACIDAD se establece la condición del número de usuarios para realizar el cálculo de sus capacidades con el uso de la Ecuación 2.5.

$$C_i = \log_2 \left( 1 + \frac{|H_i^2| * |S_i^2|}{ruido} \right) \quad i = 1, \dots, N_{subd}$$

$$C_T = \sum_{i=1}^{N_{subd}} C_i \quad (2.5)$$

donde  $C_i$  es la capacidad de la  $i$ -ésima subportadora,  $H_i$  es el subcanal correspondiente a la  $i$ -ésima subportadora,  $S_i$  es la potencia de la  $i$ -ésima subportadora,  $N_{subd}$  es el número de subportadoras de datos,  $ruido$  es el ruido presente en cada subcanal y  $C_T$  es la capacidad total obtenida.

En las líneas 4 y 5 del segmento de código 2.14 se implementa la Ecuación 2.5 para el cálculo de la potencia por usuario y con cada tipo de asignación. El resultado se guarda en un arreglo de matrices en la cual las filas representan el tipo de modulación, las columnas representan la capacidad y cada matriz representa a un usuario como se muestra en la Tabla 2.8.

```

1. function [cap, cap_gwf] = CAPACIDAD(usuarios, H_D, Sym_D,
ruido, PA, as_1)
2. if usuarios == 2
3.     %Método de subportadoras adyacentes
4.
cap(1, :, 1)=sum(log2(1+abs(H_D(1, as_1(1, :)).^2).*abs(Sym_D(1, as_
1(1, :)).^2)/ruido(1, :)));
5.
cap(1, :, 4)=sum(log2(1+abs(H_D(1, as_1(2, :)).^2).*abs(Sym_D(1, as_
1(2, :)).^2)/ruido(1, :)));
6.
cap_gwf(1, :, 1)=sum(log2(1+abs(H_D(1, as_1(1, :)).^2).*PA(1, as_1(1
, :))/ruido(1, :)));
7.
cap_gwf(1, :, 4)=sum(log2(1+abs(H_D(1, as_1(2, :)).^2).*PA(1, as_1(2
, :))/ruido(1, :)));

```

**Segmento de código 2.14.** Cálculo de la capacidad para la asignación tipo bloque.

Tanto la variable `cap` y la variable `cap_gwf` utilizan la misma ecuación, a diferencia que en el cálculo de la capacidad con el algoritmo `gwf` se utiliza la potencia asignada por dicho algoritmo.

El primer parámetro de la variable `cap` se utilizará en el programa principal para permitir elegir el tipo de modulación, el segundo parámetro contiene el valor de la capacidad del usuario y el último parámetro permite seleccionar el tipo de usuario y el método de asignación de subportadoras como se muestra en la Tabla 2.8.

**Tabla 2.8.** Guía arreglo de matrices para la función capacidad para 2 usuarios.

<b>Arreglo de matrices cap (a, b, c)</b>		
<b>Variable</b>	<b>Definición</b>	<b>Valor</b>
a	Modulación utilizada	1, 2, 3
b	Capacidad	:
c	Usuario 1	1, 2, 3
	Usuario 2	4, 5, 6
<b>Arreglo de matrices cap_gwf (a, b, c)</b>		
<b>Variable</b>	<b>Definición</b>	<b>Valor</b>
a	Modulación utilizada	1, 2, 3
b	Capacidad	:
c	Usuario 1	1, 2, 3
	Usuario 2	4, 5, 6

Para el primer método de asignación se utiliza la variable c configurada en 1 y 4, para el segundo método se configura en 2 y 5 y para el último método se configura en 3 y 6. En el caso de existir 4 usuarios el arreglo de matrices se maneja de acuerdo a la Tabla 2.9.

**Tabla 2.9.** Guía arreglo de matrices para la función capacidad para 4 usuarios.

<b>Arreglo de matrices cap (a, b, c)</b>		
<b>Variable</b>	<b>Definición</b>	<b>Valor</b>
a	Modulación utilizada	1, 2, 3
b	Capacidad	:
c	Usuario 1	1, 2, 3
	Usuario 2	4, 5, 6
	Usuario 3	7, 8, 9
	Usuario 4	10, 11, 12
<b>Arreglo de matrices cap_gwf (a, b, c)</b>		
<b>Variable</b>	<b>Definición</b>	<b>Valor</b>
a	Modulación utilizada	1, 2, 3
b	Capacidad	:
c	Usuario 1	1, 2, 3
	Usuario 2	4, 5, 6
	Usuario 3	7, 8, 9
	Usuario 4	10, 11, 12

En las líneas 85 y 86 del segmento de código 2.15 se asigna a la variable `capGWF` y `capM` la capacidad final de los usuarios, en estas variables se muestra la capacidad según su modulación, SNR y usuario con los valores indicados en las Tablas 2.8 y 2.9. Adicionalmente se utiliza un factor de  $1e6$ , el cual permite expresar la capacidad en Mbps.

```

85.
capGWF(modlc,iter,:)=AB_Sub*sum(CAP_GWF(modlc,,:,:))/(1e6*rep_snr)
86.
capM(modlc,iter,:)=AB_Sub*sum(CAP(modlc,,:,:))/(1e6*rep_snr);

```

### Segmento de código 2.15. Capacidad de todas las subportadoras.

Finalmente, en las líneas 88 a la 101 del anexo A se guardan las variables para su análisis para el caso de 2 usuarios, como se muestra en el segmento de código 2.16. Para el caso de 4 usuarios se realiza un proceso similar, cambiando el nombre de las variables en las cuales se va guardar la información.

```

88. %-----Respaldo de las variables-----
89. if usuarios==2
90.     if Nsubt==64
91.         save('Capacidades_2_64', 'SNR', 'capM', 'capGWF')
92.     elseif Nsubt==128
93.         save('Capacidades_2_128', 'SNR', 'capM', 'capGWF')
94.     end
95. elseif usuarios==4
96.     if Nsubt==64
97.         save('Capacidades_4_64', 'SNR', 'capM', 'capGWF')
98.     elseif Nsubt==128
99.         save('Capacidades_4_128', 'SNR', 'capM', 'capGWF')
100.    end
101. end

```

### Segmento de código 2.16. Respaldo de resultados.

## 2.7 GRÁFICAS

Para la visualización de los resultados se creó la función grafica para 2 y 4 usuarios, las cuales se encuentran implementadas en los anexos I y J. Esta función carga los datos previamente guardados como se muestra en el segmento de código 2.17. Para separar los resultados, se los dividió por usuarios y por el número de subportadoras, lo cual se debe configurar en la línea 5 del segmento de código 2.17.

```

1. function [ ] = GRAFICA_2_Usuarios()
5. Nsubt=64;
6.     if Nsubt==64
7.         load Capacidades_2_64.mat
8.     elseif Nsubt==128
9.         load Capacidades_2_128.mat
10.    end

```

### Segmento de código 2.17. Código para cargar los datos.

Una vez cargados los datos, se calcula la capacidad total ya que los datos cargados contienen la capacidad por usuario. En la línea 11 del segmento de código 2.18 se muestra un ejemplo de cómo se realiza el cálculo de la capacidad total para el caso de 2 usuarios con el uso de la asignación tipo bloque. La capacidad total ( $CapT(:, :, 1)$ ) es igual a la suma de la capacidad de cada usuario ( $capM(:, :, 1) + capM(:, :, 4)$ ). Los valores de 1 y 4 corresponden a la primera asignación como se muestra en la Tabla 2.8.

En las líneas 12 y 13 se calcula la capacidad total para los otros métodos de asignación de subportadoras mientras que en las líneas 14, 15 y 16 se calcula la capacidad total con el uso de *geometric water-filling* para el caso de 2 usuarios. Las variables de capacidad total con y sin *geometric water-filling* ( $CapT$  y  $CapT\_GWF$  respectivamente) para 2 y 4 usuarios mantiene el arreglo de matrices establecido en las Tablas 2.8 y 2.9

```

11. CapT(:, :, 1)=capM(:, :, 1)+capM(:, :, 4);
12. CapT(:, :, 2)=capM(:, :, 2)+capM(:, :, 5);
13. CapT(:, :, 3)=capM(:, :, 3)+capM(:, :, 6);
14. CapT_GWF(:, :, 1)=capGWF(:, :, 1)+capGWF(:, :, 4);
15. CapT_GWF(:, :, 2)=capGWF(:, :, 2)+capGWF(:, :, 5);
16. CapT_GWF(:, :, 3)=capGWF(:, :, 3)+capGWF(:, :, 6);

```

**Segmento de código 2.18.** Cálculo de la capacidad total.

Como un ejemplo de una gráfica se presenta el segmento de código 2.19. Con dicho código se grafica la capacidad total del sistema para los 3 tipos de asignación con una modulación 16-QAM con y sin el algoritmo *geometric water-filling*, configurando las variables  $CapT\_GWF$  y  $CapT$  según las Tablas 2.10 y 2.11.

**Tabla 2.10.** Guía arreglo de matrices para la función capacidad para 2 usuarios.

Arreglo de matrices CapT (a, b, c)		
Variable	Valor	Definición
a	2	Modulación 16-QAM
b	:	Todas las capacidades para cada SNR
c	1, 2, 3,	Método Bloque, Alternado y Aleatorio

**Tabla 2.11.** Guía arreglo de matrices para la función capacidad con GWF para 2 usuarios.

Arreglo de matrices CapT_GWF (a, b, c)		
Variable	Valor	Definición
a	2	Modulación 16-QAM
b	:	Todas las capacidades para cada SNR
c	1, 2, 3	Método Bloque, Alternado y Aleatorio

```

21. %% 16QAM total con y sin GWF
22. figure(1)
23. plot(SNR,CapT_GWF(2,:,1),'-o', SNR,CapT_GWF(2,:,2),'-
p',SNR,CapT_GWF(2,:,3),'-*')
24. hold on
25. plot(SNR,CapT(2,:,1),'--*',SNR,CapT(2,:,2),'--<',
SNR,CapT(2,:,3),'--+')
26. grid on
27. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
28. legend('GWF + Metodo Bloque','GWF + Metodo Alternado','GWF +
Metodo Aleatorio','Sin GWF + Metodo Bloque','Sin GWF + Metodo
Alternado','Sin GWF + Metodo Aleatorio','Location','northwest')
29. title('Capacidad Total del Sistema (16-QAM)')

```

**Segmento de código 2.19.** Código ejemplo de la gráfica de la capacidad Total.

En la línea 23 del segmento de código 2.19 se grafica la capacidad total (índice de columna configurado en :, para obtener las capacidades para todos los SNR) con el uso de la modulación 16-QAM (índice de fila configurado en 2) con el algoritmo *geometric water-filling* para los 3 usuarios (tercer índice configurado en 1, 2 y 3). Para los tres métodos de asignación con y sin el uso del algoritmo *geometric water-filling* se utiliza el comando `hold on`, el cual permite presentar las gráficas de la capacidad total, línea 23, y las gráficas de la capacidad total con el uso del algoritmo *geometric water-filling*, línea 25, en la misma figura. Para una mejor visualización de las gráficas se utiliza el comando `grid on` que se encarga de habilitar una cuadrícula dentro de la figura. Para identificar de mejor manera las unidades en las que se encuentran las gráficas se utiliza el comando `xlabel` para colocar una etiqueta en el eje X y `ylabel` para colocar una etiqueta en el eje Y como se muestra en la línea 27 del segmento de código 2.19. Para identificar cada método de asignación dentro de la gráfica se utiliza el comando `legend` el cual identifica el color y tipo de línea que se utilizó en el método de asignación y agrega una leyenda del mismo color e indicando el tipo de línea que utilizó como se muestra en la línea 28 del segmento de código 2.19. Finalmente, a cada grafica se le asigna un título como se muestra en la línea 29 del segmento de código 2.19.

### 3. RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados obtenidos mediante simulación. Los resultados se muestran en curvas de capacidad versus SNR. La capacidad se muestra en el eje Y y tiene como unidad el megabit por segundo (Mbps) mientras que en el eje X se muestra la SNR en decibelios (dB). Además de las curvas, se muestran tablas que contienen la mejora porcentual de la capacidad que se obtiene con el algoritmo *geometric water-filling*. Dentro del trabajo de titulación se considera el uso de 3 esquemas de modulación, QPSK, 16-QAM y 64-QAM, los cuales serán analizados conjuntamente con las asignaciones de subportadoras alternadas, bloque y aleatoria. La simulación se realizó con los parámetros establecidos en las Tablas 3.1 y 3.2.

**Tabla 3.1.** Configuración para 2 y 4 usuarios.

<b>Variables</b>	<b>64 Subportadoras</b>	<b>128 Subportadoras</b>
usuarios	2	2
AB	20e6	40e6
rep_snr	5000	5000
ig	$\frac{1}{4}$	$\frac{1}{4}$
SNR	[ 0: 1: 20 ]	[ 0: 1: 20 ]
<b>Variables</b>	<b>64 Subportadoras</b>	<b>128 Subportadoras</b>
usuarios	4	4
AB	20e6	40e6
rep_snr	5000	5000
ig	$\frac{1}{4}$	$\frac{1}{4}$
SNR	[ 0: 1: 20 ]	[ 0: 1: 20 ]

**Tabla 3.2.** Configuración del PDP para cada usuario.

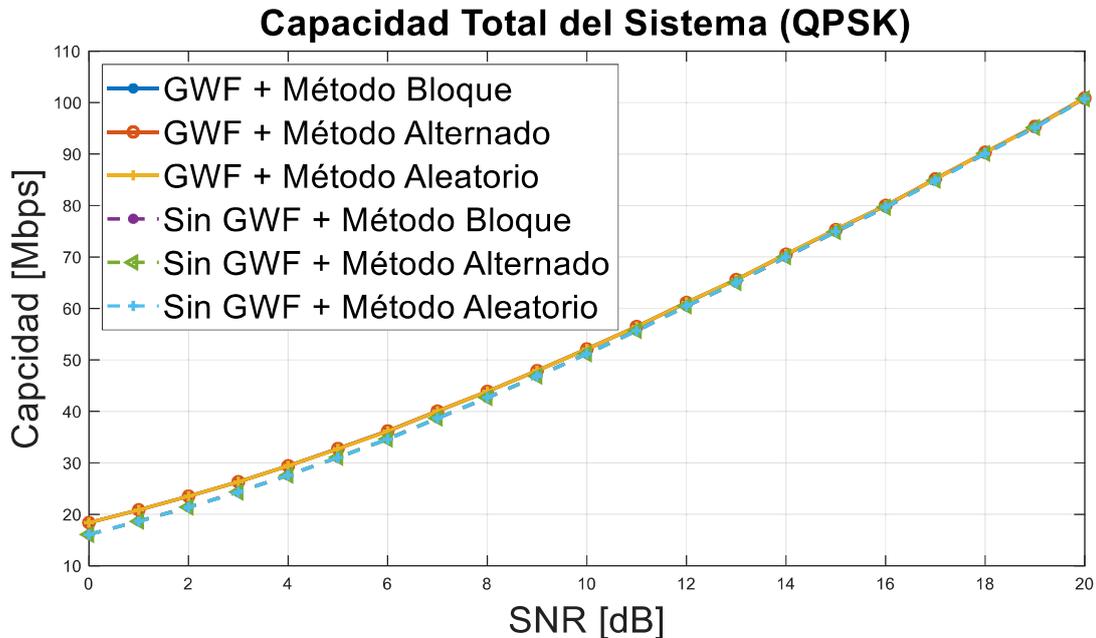
<b>Usuario</b>	<b>Potencia [dB]</b>	<b>Retardo</b>
1	[0 -3 -6]	[0 6 12]
2	[0 -3 -6 -9]	[0 3 6 9]
3	[0 0]	[0 5]
4	[0 -3 -6 -9]	[0 2 4 6]

El desempeño de las asignaciones de subportadoras varía con el número de usuarios ya que el incremento de usuarios permite obtener una mayor variedad de canales. Al trabajar con una mayor diversidad de canales, cada usuario obtiene una capacidad diferente [12]. Mientras menor sea el número de usuarios, aumenta la probabilidad de que los canales sean similares así como la capacidad que pueden alcanzar.

### 3.1. CAPACIDAD DEL SISTEMA CON DOS USUARIOS CON UN CANAL DE 20 MHZ

#### 3.1.1. CAPACIDAD TOTAL

##### 3.1.1.1. Modulación QPSK



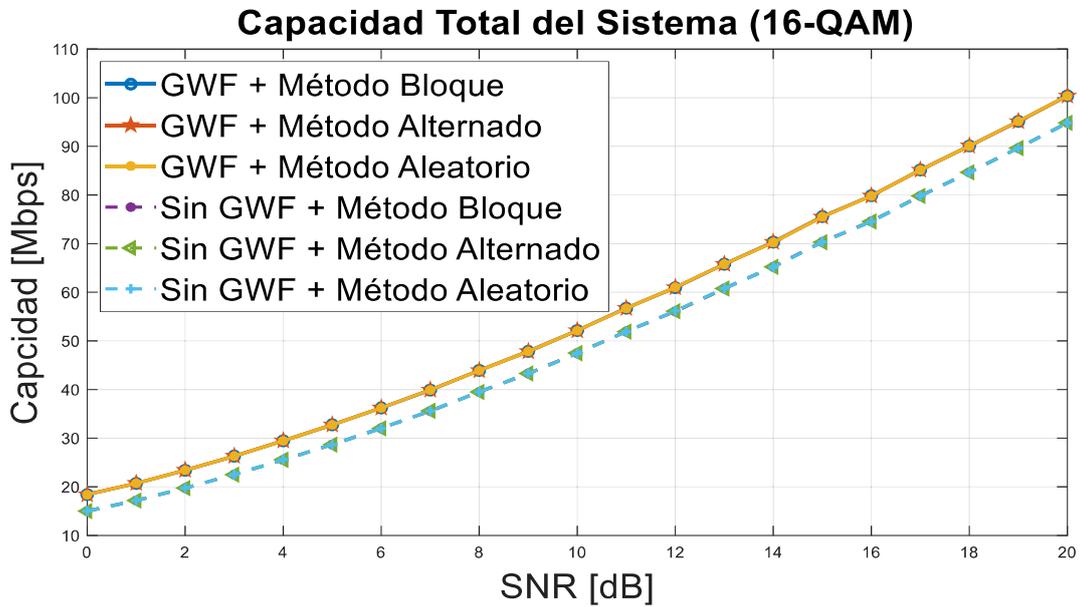
**Figura 3.1.** Capacidad total con modulación QPSK para 2 usuarios sobre un canal de 20 MHz.

En la Figura 3.1 se muestra las curvas de la capacidad total del sistema OFDMA obtenida con la modulación QPSK para los 3 tipos de asignación de subportadoras. En la Figura 3.1 se muestra que el algoritmo GWF aumenta la capacidad de transmisión del canal para niveles bajos de SNR hasta un 14.47% en 0 dB utilizando el método de asignación de subportadoras alternadas, mientras que para valores altos de SNR la capacidad con y sin el uso del algoritmo GWF es similar obteniendo mejoras de la capacidad menores al 1% para los 3 tipos de asignación de subportadoras.

##### 3.1.1.2. Modulación 16-QAM

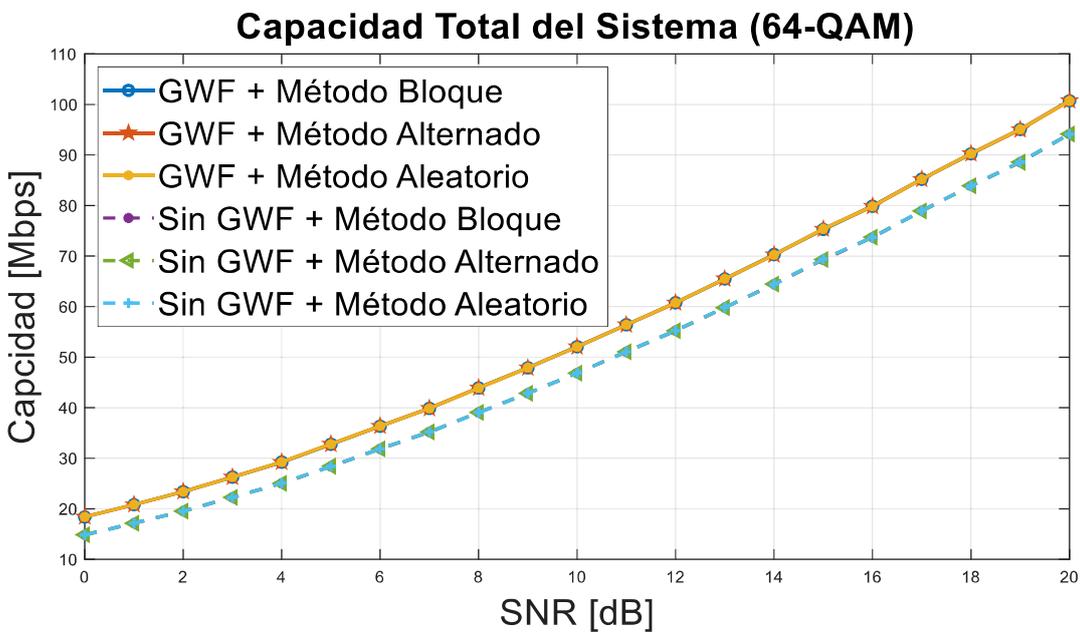
En la Figura 3.2 se muestran las curvas de la capacidad obtenida con la modulación 16-QAM comparando los tres tipos de asignación de subportadoras utilizando el algoritmo GWF. Con el uso de dicho algoritmo se optimiza la capacidad del sistema OFDMA para todos los niveles de SNR, alcanzando una mejora del 22.71% con el método de asignación de subportadoras alternadas para 0 dB mientras que para 16 dB se obtiene una mejora de la capacidad del 7.05%. Dichas mejoras se obtienen utilizando la

modulación 16-QAM, la cual trabaja con diferentes amplitudes y el algoritmo permite distribuir la potencia de las subportadoras de una manera eficiente para de esta forma aprovechar los subcanales en mejor estado.



**Figura 3.2.** Capacidad total con modulación 16-QAM para 2 usuarios sobre un canal de 20 MHz.

### 3.1.1.3. Modulación 64-QAM

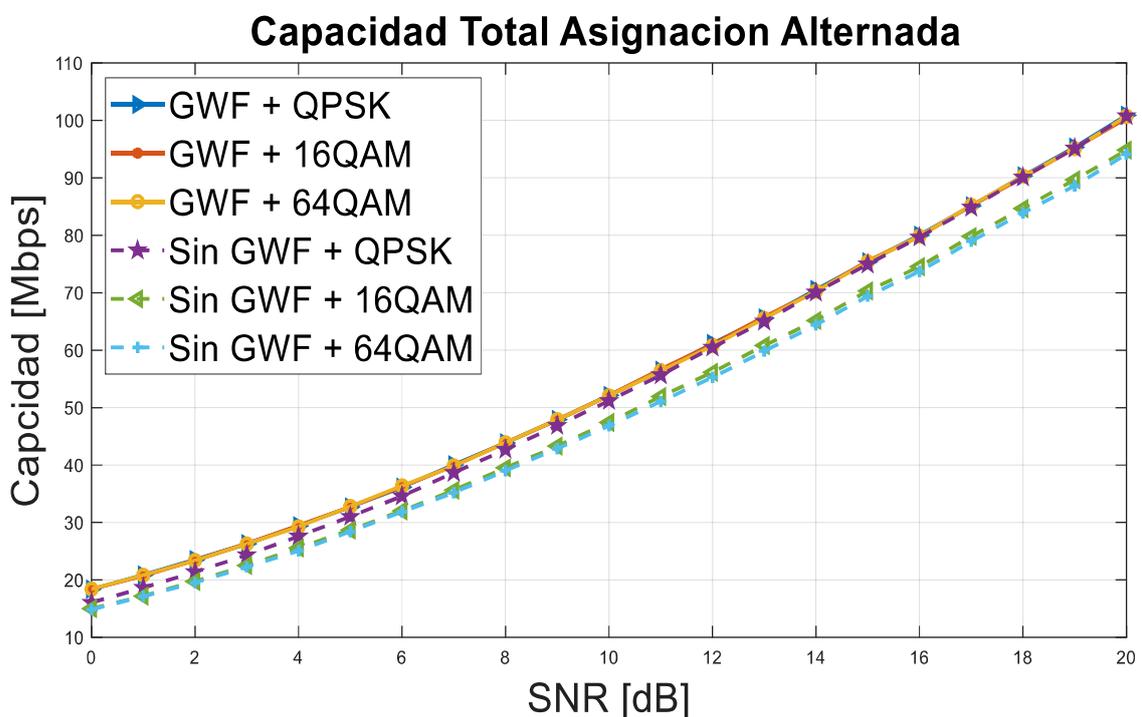


**Figura 3.3.** Capacidad total con modulación 64-QAM para 2 usuarios sobre un canal de 20 MHz.

La modulación 64-QAM permite transmitir una mayor cantidad de información, ya que cada símbolo representa 6 bits. En este esquema de modulación se utiliza símbolos con diferentes amplitudes las cuales se transmiten sobre el canal sin importar su condición, mientras que con el algoritmo GWF se optimiza la distribución de potencia de cada símbolo considerando el estado de canal. En la Figura 3.3 se muestran las curvas de la capacidad total del sistema OFDMA utilizando la modulación 64-QAM para comparar los tres métodos de asignación de subportadoras. La mejora de la capacidad que se obtiene al utilizar el algoritmo GWF para 0 dB es de 23.91% utilizando el método de subportadoras alternadas mientras que en 16 dB se obtiene una mejora del 8.28% para el mismo tipo de asignación de subportadoras.

### 3.1.1.4. Modulaciones QPSK, 16-QAM y 64-QAM

En la Figura 3.4 se muestra la comparación de las curvas de la capacidad total del sistema para los tres esquemas de modulación utilizando la asignación de subportadoras alternadas, la cual presentó mejores resultados para todos los niveles de SNR evaluados. Para cada esquema de modulación el algoritmo GWF ofrece una mejora de la capacidad, la cual se incrementa mientras más compleja sea la modulación utilizada.



**Figura 3.4.** Capacidad total con asignación de subportadoras alternadas para 2 usuarios sobre un canal de 20 MHz.

El algoritmo GWF se desempeña de mejor manera al utilizar la modulación 64-QAM, obteniendo como resultado las mejoras que se muestran en la Tabla 3.3.

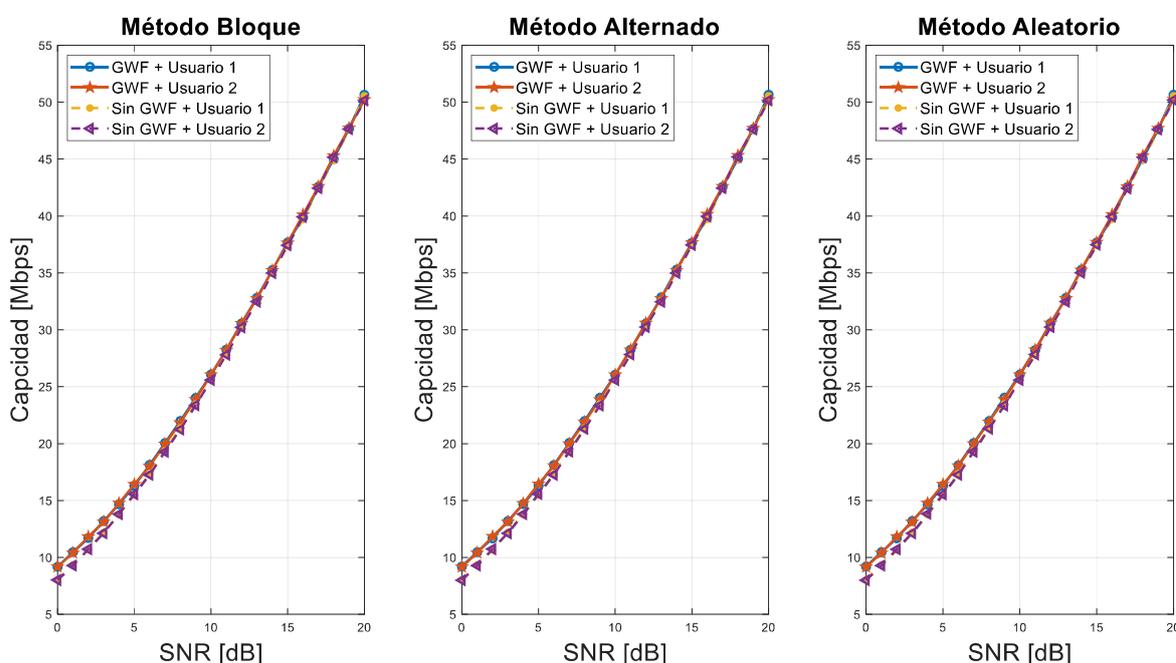
**Tabla 3.3.** Mejora porcentual de la capacidad con las diferentes modulaciones para el método de subportadoras alternadas para 2 usuarios sobre un canal de 20 MHz.

SNR [dB]	0	4	8	12	16	20
Modulación QPSK	14.47%	6.71%	2.84%	1.11%	0.43%	0.14%
Modulación 16-QAM	22.71%	15.36%	11.19%	8.70%	7.05%	5.85%
Modulación 64-QAM	23.91%	16.74%	12.45%	9.94%	8.28%	7.00%

### 3.1.2. CAPACIDAD POR USUARIO

#### 3.1.2.1. Modulación QPSK

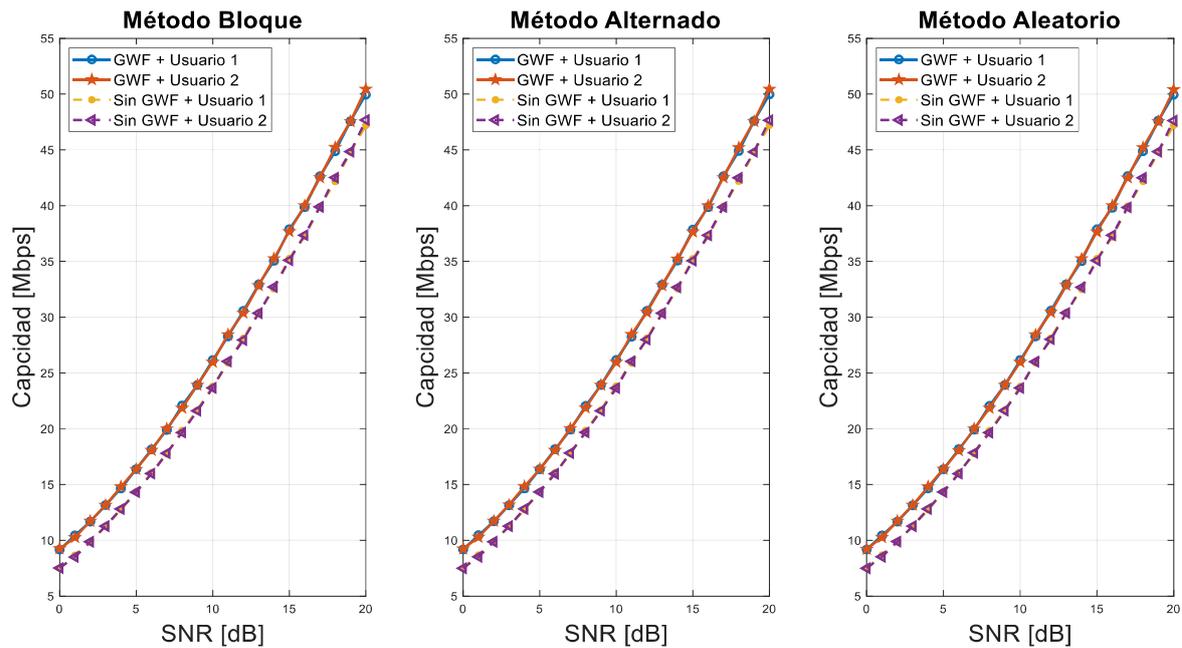
En la Figura 3.5 se muestran tres gráficas que comparan los tres métodos de asignación con la capacidad de cada usuario utilizando la modulación QPSK con y sin el uso del algoritmo *geometric water-filling*. El algoritmo aumenta la capacidad de transmisión del canal de manera equitativa para los 2 usuarios. La mejora promedio de la capacidad de los dos usuarios que se obtuvo en 0 dB es del 14.38% con el método de asignación tipo bloque, 14.47% con el método de asignación alternada y 14.40% para el método de asignación aleatorio, siendo el método de asignación alternado el que mejor se ve beneficiado por el algoritmo GWF.



**Figura 3.5.** Capacidad de 2 usuarios utilizando la modulación QPSK con los métodos bloque, alternado y aleatorio sobre un canal de 20 MHz.

### 3.1.2.2. Modulación 16-QAM

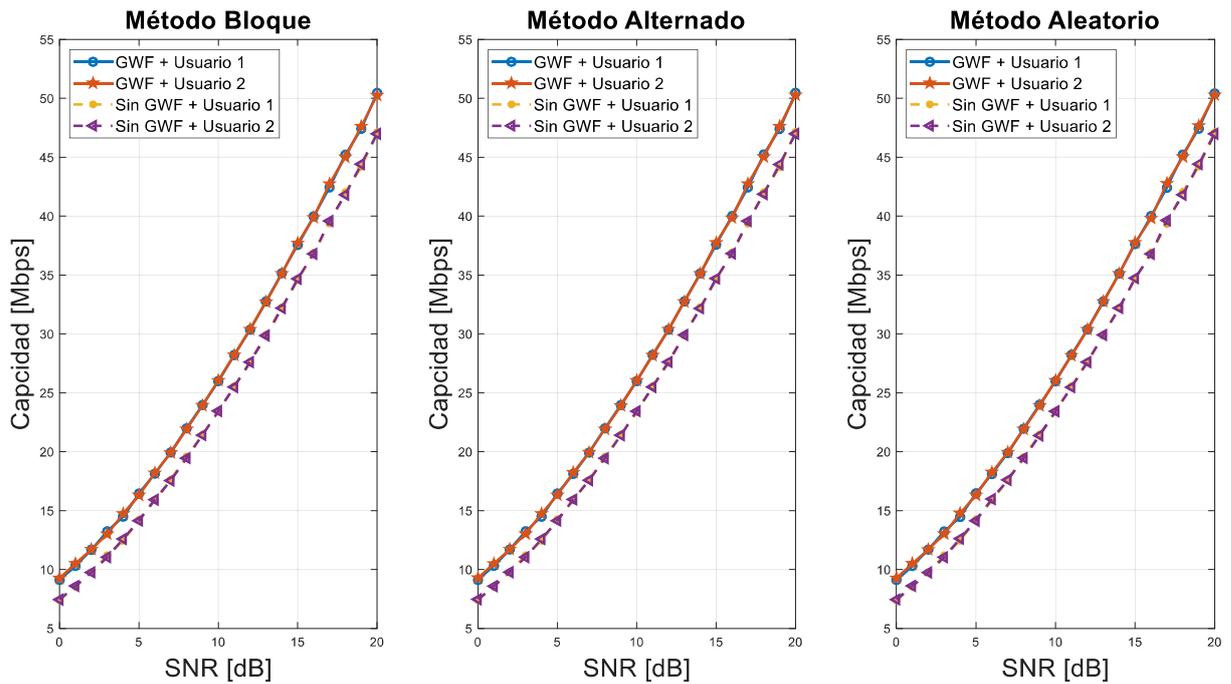
En la Figura 3.6 se muestran las curvas de la capacidad obtenida con la modulación 16-QAM con y sin el uso del algoritmo *geometric water-filling* para los tres métodos de asignación de subportadoras. La capacidad de los 2 usuarios se beneficia por el uso del algoritmo GWF, obteniendo una mejora aproximada del 22% para los tres tipos de asignación de recursos para un valor de SNR de 0 dB.



**Figura 3.6.** Capacidad por usuario con modulación 16-QAM, métodos bloque, alternado y aleatorio sobre un canal de 20 MHz.

### 3.1.2.3. Modulación 64-QAM

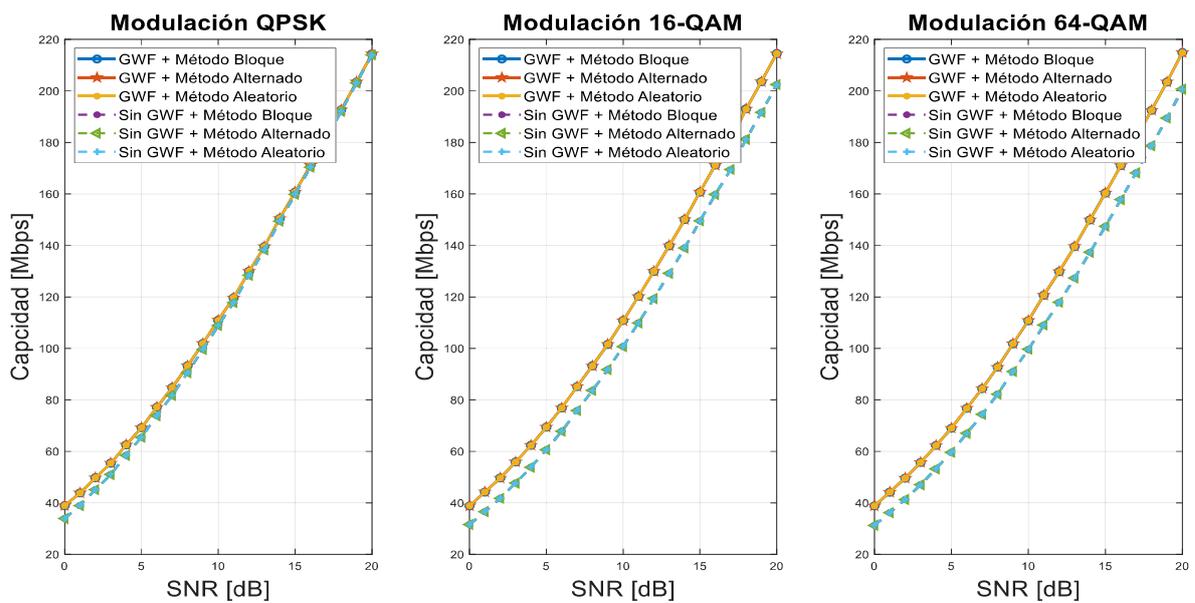
En la Figura 3.7 se muestra la capacidad de los dos usuarios utilizando la modulación 64-QAM con y sin el uso del algoritmo *geometric water-filling* para los tres métodos de asignación de subportadoras. El algoritmo aumenta la capacidad del canal de mejor manera en comparación con los esquemas de modulación previamente analizados, obteniendo una mejora aproximada del 24% para los 2 usuarios en las peores condiciones evaluadas, es decir en un valor de SNR de 0 dB.



**Figura 3.7.** Capacidad por usuario utilizando la modulación 64-QAM para los métodos bloque, alternado y aleatorio sobre un canal de 20 MHz.

## 3.2. CAPACIDAD DEL SISTEMA CON DOS USUARIOS CON UN CANAL DE 40 MHZ

### 3.2.1. CAPACIDAD TOTAL



**Figura 3.8.** Capacidad total con modulación QPSK, 16-QAM y 64-QAM para 2 usuarios sobre un canal de 40 MHz.

Las gráficas de los resultados obtenidos en la simulación con dos usuarios sobre un canal de 40 MHz tienen una tendencia similar a las gráficas previamente detalladas, a diferencia de que la capacidad del canal, la cual se incrementa por un factor cercano a dos, debido a que se aumenta el ancho de banda del canal de 20 a 40 MHz y el número subportadoras disponibles de 52 a 108.

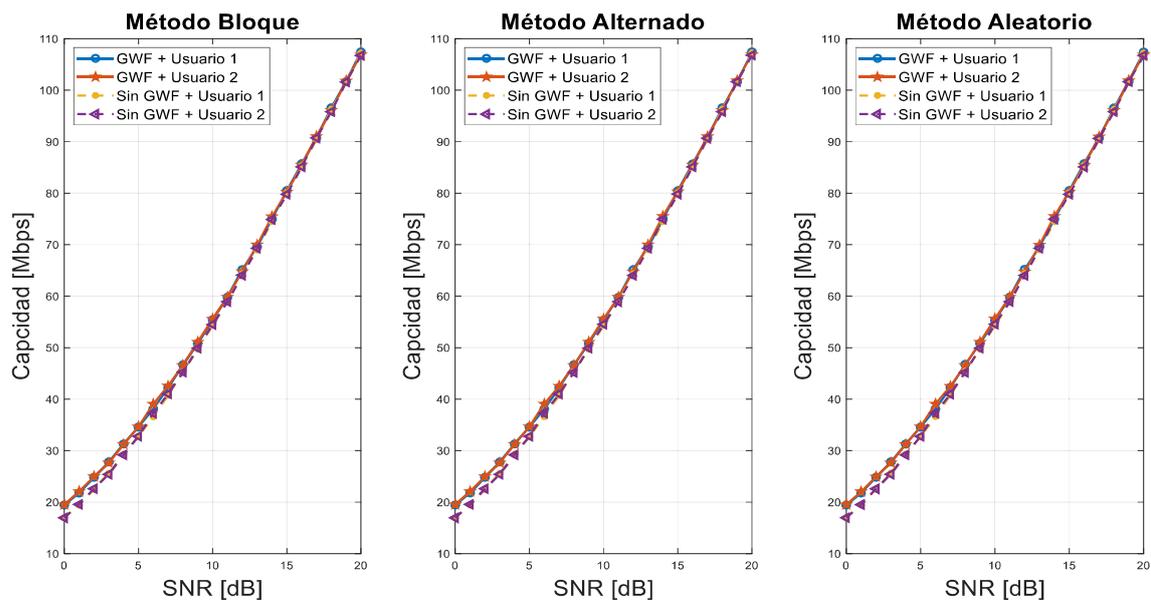
En la Figura 3.8 se muestra de forma general la mejora de la capacidad total del sistema obtenida con y sin el uso del algoritmo GWF para los tres esquemas de modulación propuestos, comparando en cada uno los tres métodos de asignación de subportadoras. El algoritmo GWF trabaja de mejor manera con el método de asignación de subportadoras alternadas, logrando mejoras de hasta 24.25% en la capacidad del canal con un SNR de 0 dB como se muestra en la tabla 3.4.

**Tabla 3.4.** Mejora porcentual de la capacidad con las diferentes modulaciones para el método de subportadoras alternadas para 2 usuarios sobre un canal de 40 MHz.

SNR [dB]	0	4	8	12	16	20
Modulación QPSK	14.88%	6.91%	2.97%	1.17%	0.43%	0.16%
Modulación 16-QAM	23.15%	15.76%	11.38%	8.85%	7.12%	5.93%
Modulación 64-QAM	24.25%	16.93%	12.73%	10.11%	8.38%	7.07%

### 3.2.2. CAPACIDAD POR USUARIO

#### 3.2.2.1. Modulación QPSK

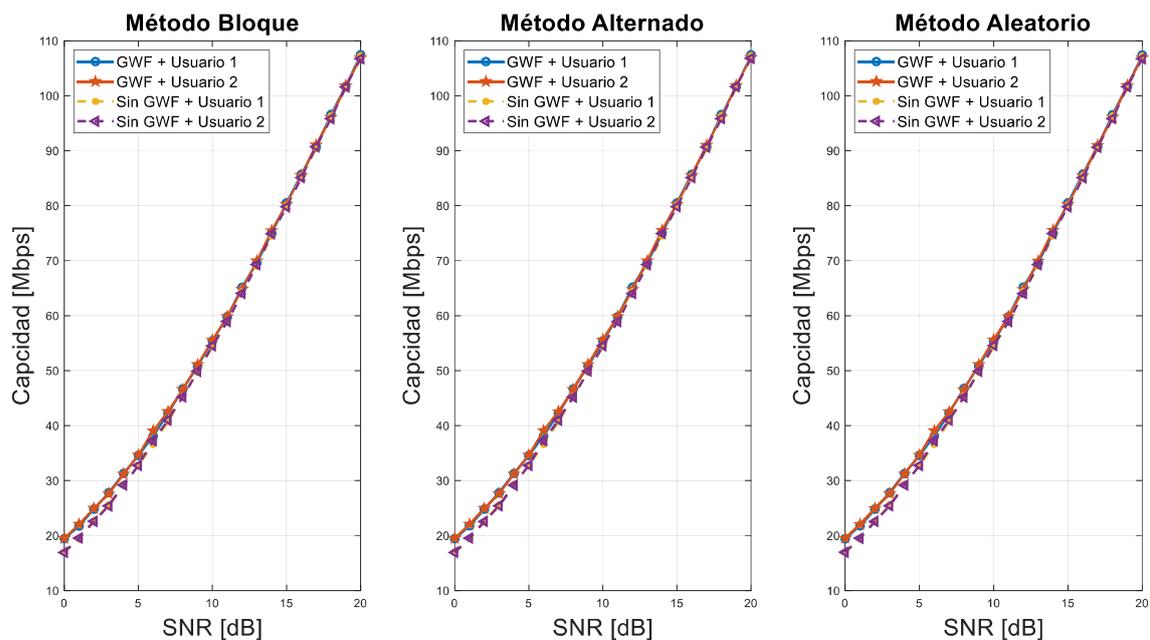


**Figura 3.9.** Capacidad por usuario utilizando la modulación QPSK para los métodos bloque, alternado y aleatorio sobre un canal de 40 MHz.

En la Figura 3.9 se muestran las curvas de la capacidad de cada usuario utilizando la modulación QPSK con y sin el uso del algoritmo *geometric water-filling* con los tres métodos de asignación. El algoritmo aumenta la capacidad de transmisión del canal para valores bajos de SNR hasta un 15.19% con el método de asignación de subportadoras aleatorias.

### 3.2.2.2. Modulación 16-QAM

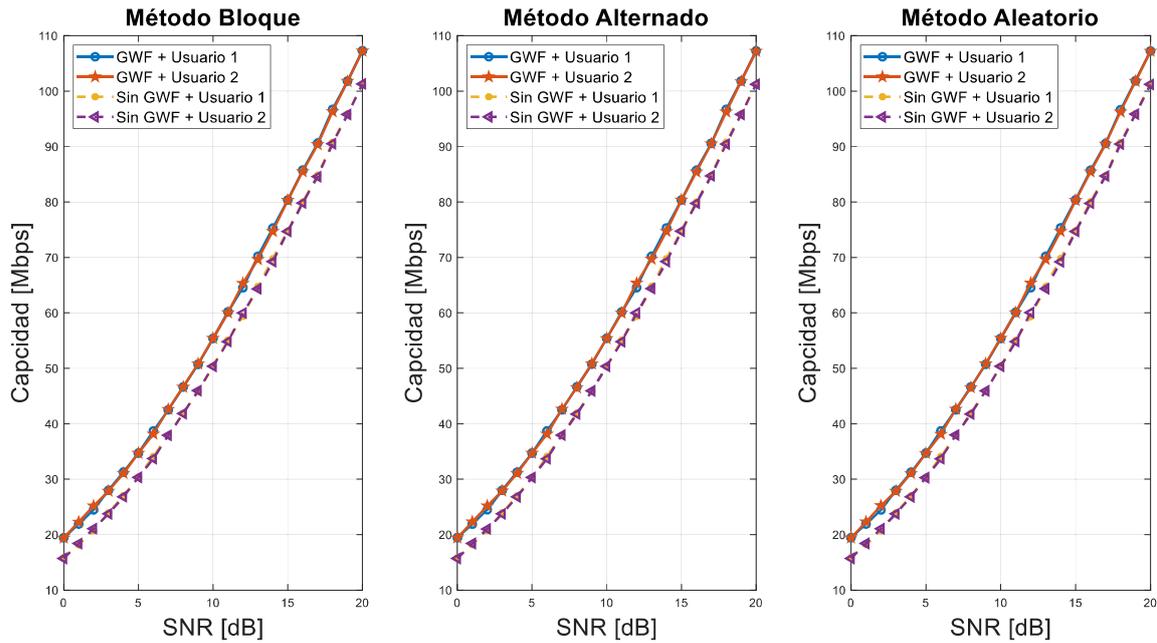
En la Figura 3.10 se muestra la capacidad de cada usuario utilizando la modulación 16-QAM con y sin el uso del algoritmo *geometric water-filling* comparando los tres métodos de asignación de subportadoras. El algoritmo aumenta la capacidad de transmisión del canal de cada usuario de manera similar para los tres métodos de asignación de subportadoras manteniéndose en un promedio de 23.20% para una SNR de 0 dB mientras que en 20 dB obtiene una mejora del 5.91% en la capacidad.



**Figura 3.10.** Capacidad por usuario con modulación 16-QAM, métodos bloque, alternado y aleatorio para 2 usuarios sobre un canal de 40 MHz.

### 3.2.2.3. Modulación 64-QAM

En la Figura 3.11 se muestran las curvas de la capacidad de los usuarios que se obtiene con la modulación 64-QAM con y sin el uso del algoritmo *geometric water-filling*, comparando los tres métodos de asignación de subportadoras. El algoritmo aumenta la capacidad del canal para los tres tipos asignación de subportadoras de manera similar para los dos usuarios ya que para un SNR de 0 dB, un usuario mejora su capacidad un 23.75% y el otro usuario mejora en un 23.75%.



**Figura 3.11.** Capacidad de dos usuarios utilizando la modulación 64-QAM para los métodos bloque, alternado y aleatorio sobre un canal de 40 MHz.

### 3.3. CAPACIDAD DEL SISTEMA CON CUATRO USUARIOS CON UN CANAL DE 20 MHZ

#### 3.3.1. CAPACIDAD TOTAL

La mejora de la capacidad total del sistema se mantiene similar, aumentando la eficiencia del algoritmo mientras mayor sea la complejidad de la modulación y menor sea el valor de SNR ya que el algoritmo puede identificar y aprovechar los subcanales con las mejores condiciones, alcanzando mejoras de hasta el 24.86% de la capacidad para todo el sistema como se muestra en las Tablas 3.5, 3.6 y 3.7 en las que cada una muestra la mejora porcentual de la capacidad para cada tipo de modulación evaluado.

**Tabla 3.5.** Mejora porcentual de la capacidad con la modulación QPSK para 4 usuarios con un canal de 20 MHz.

SNR [dB]	0	4	8	12	16	20
Método bloque	15.15%	6.84%	3.10%	1.31%	0.40%	0.14%
Método alternado	15.46%	7.20%	2.99%	1.17%	0.43%	0.15%
Método aleatorio	15.34%	7.15%	3.01%	1.17%	0.44%	0.15%

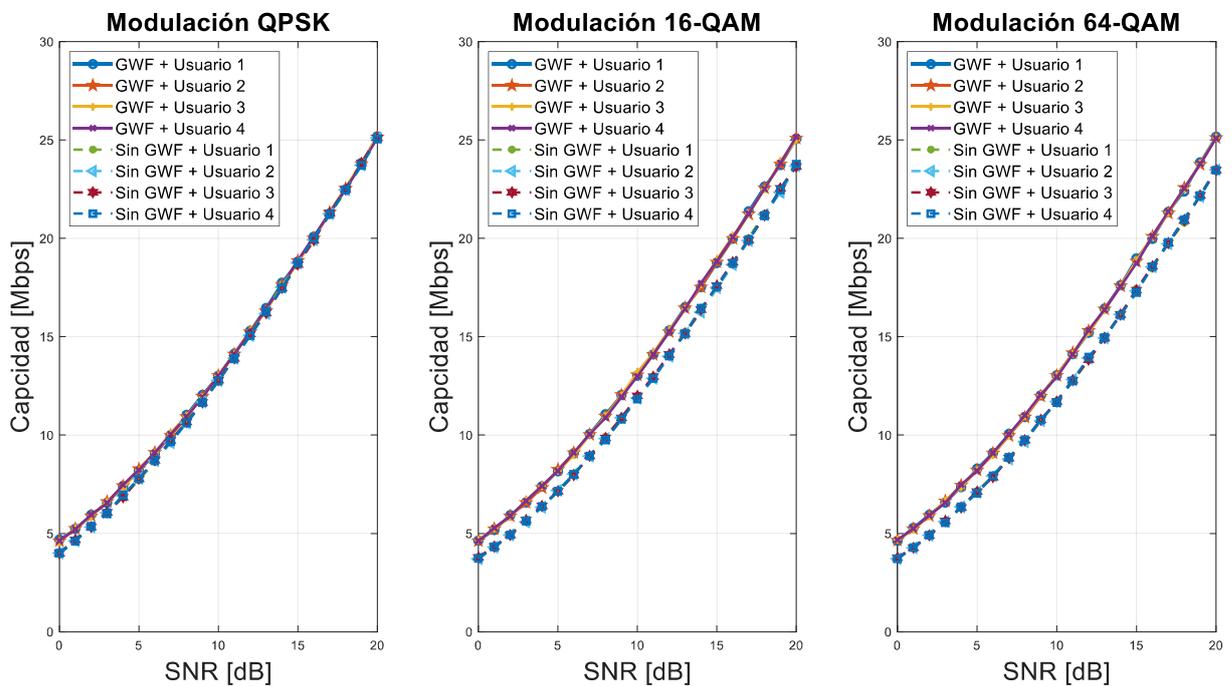
**Tabla 3.6.** Mejora porcentual de la capacidad con la modulación 16-QAM para 4 usuarios con un canal de 20 MHz.

SNR [dB]	0	4	8	12	16	20
Método bloque	24.18%	16.22%	11.56%	8.92%	7.19%	5.77%
Método alternado	23.84%	16.04%	11.48%	8.78%	7.03%	5.82%
Método aleatorio	23.69%	15.91%	11.45%	8.76%	7.03%	5.85%

**Tabla 3.7.** Mejora porcentual de la capacidad con la modulación 64-QAM para 4 usuarios con un canal de 20 MHz.

SNR [dB]	0	4	8	12	16	20
Método bloque	25.01%	16.67%	12.87%	9.86%	8.33%	6.96%
Método alternado	24.93%	16.93%	12.69%	10.00%	8.29%	6.97%
Método aleatorio	24.86%	16.96%	12.64%	9.99%	8.32%	7.00%

### 3.3.2. CAPACIDAD POR USUARIO



**Figura 3.12.** Capacidad por usuario utilizando el método de asignación alternada sobre un canal de 20 MHz.

En la Figura 3.12 se muestra la capacidad de cada usuario utilizando el método de asignación de subportadoras alternadas para cada tipo de modulación. En cada gráfica se compara la capacidad de cada usuario con y sin el uso del algoritmo *geometric water-filling*. Para el caso de la modulación QPSK, el algoritmo aumenta la capacidad de los cuatro usuarios hasta aproximadamente 12 dB a partir de los cuales el aumento deja de ser considerable. Tanto para la modulación 16-QAM Y 64-QAM el algoritmo trabaja de manera eficiente para valores bajos de SNR como se muestra en la tabla 3.8.

**Tabla 3.8.** Mejora porcentual de la capacidad de cuatro usuarios utilizando la asignación de subportadoras alternadas sobre un canal de 20 MHz.

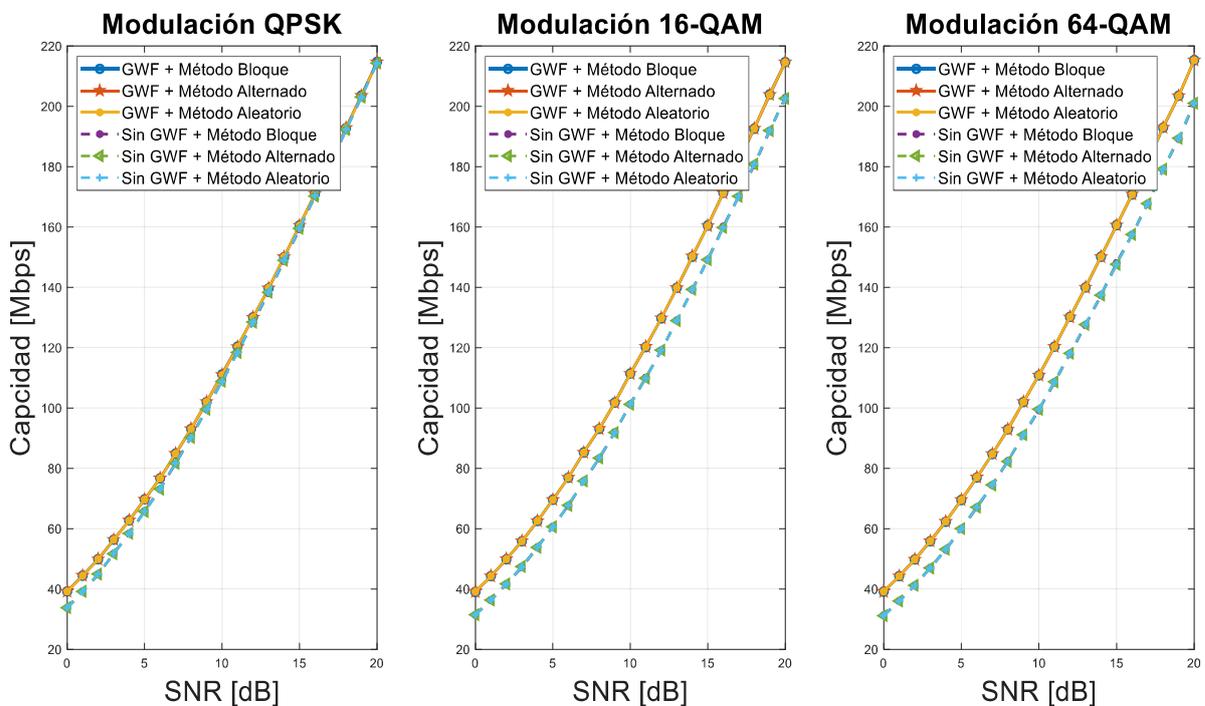
Modulación QPSK						
SNR [dB]	0	4	8	12	16	20
Usuario 1	15.68%	7.02%	3.15%	1.16%	0.44%	0.16%
Usuario 2	15.37%	7.28%	2.99%	1.18%	0.46%	0.16%
Usuario 3	14.84%	6.73%	2.96%	1.18%	0.42%	0.14%
Usuario 4	15.48%	7.56%	2.95%	1.14%	0.42%	0.14%
Modulación 16-QAM						
SNR [dB]	0	4	8	12	16	20
Usuario 1	23.77%	16.10%	11.44%	8.73%	7.02%	5.88%
Usuario 2	24.02%	15.92%	11.50%	8.83%	7.09%	5.94%
Usuario 3	23.40%	15.65%	11.33%	8.89%	7.04%	5.76%
Usuario 4	23.57%	15.95%	11.51%	8.59%	6.96%	5.80%
Modulación 64-QAM						
SNR [dB]	0	4	8	12	16	20
Usuario 1	24.35%	16.50%	12.53%	10.07%	8.18%	7.00%
Usuario 2	25.03%	17.22%	12.91%	10.07%	8.29%	6.94%
Usuario 3	24.63%	16.26%	12.56%	10.02%	8.26%	7.02%
Usuario 4	25.69%	17.74%	12.73%	9.86%	8.41%	6.92%

El método de asignación de subportadoras que tiene el mayor rendimiento es el método de subportadoras alternadas, el cual independientemente de la modulación utilizada se ve mejor beneficiado por el algoritmo GWF.

### 3.4. CAPACIDAD DEL SISTEMA CON CUATRO USUARIOS CON UN CANAL DE 40 MHZ

#### 3.4.1. CAPACIDAD TOTAL

De manera similar que en el caso de 2 usuarios, al aumentar el ancho de banda y las subportadoras disponibles, la capacidad que puede ofrecer el sistema aumenta. En la Figura 3.13 se muestra la comparación de la capacidad total del sistema con los tres métodos de asignación de subportadoras para los tres esquemas de modulación por separado. Las curvas de la capacidad se mantienen similares al caso de dos usuarios, obteniendo mejores resultados para niveles bajos de SNR con el uso de la modulación 64-QAM.



**Figura 3.13.** Capacidad total con modulación QPSK, 16-QAM y 64-QAM para 4 usuarios sobre un canal de 40 MHz.

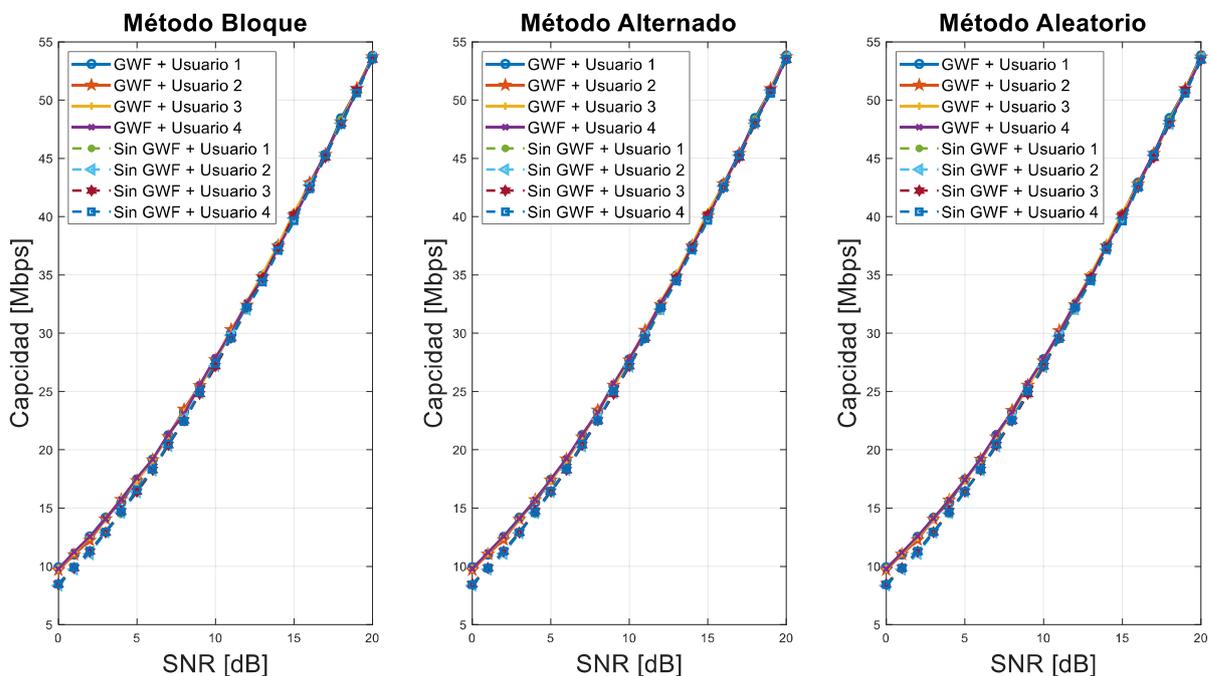
La máxima mejora de la capacidad corresponde al 25.56%, la cual se obtiene utilizando el algoritmo GWF con la modulación 64-QAM y el método de asignación de subportadoras alternadas como se muestra en la Tabla 3.9.

**Tabla 3.9.** Mejora porcentual de la capacidad con las diferentes modulaciones utilizando el método de subportadoras alternadas para 4 usuarios sobre un canal de 40 MHz.

SNR [dB]	0	4	8	12	16	20
Modulación QPSK	16.05%	7.37%	3.17%	1.17%	0.43%	0.15%
Modulación 16-QAM	24.33%	16.22%	11.63%	8.91%	7.14%	5.97%
Modulación 64-QAM	25.56%	17.30%	12.88%	10.17%	8.42%	7.09%

### 3.4.2. CAPACIDAD POR USUARIO

#### 3.4.2.1. Modulación QPSK



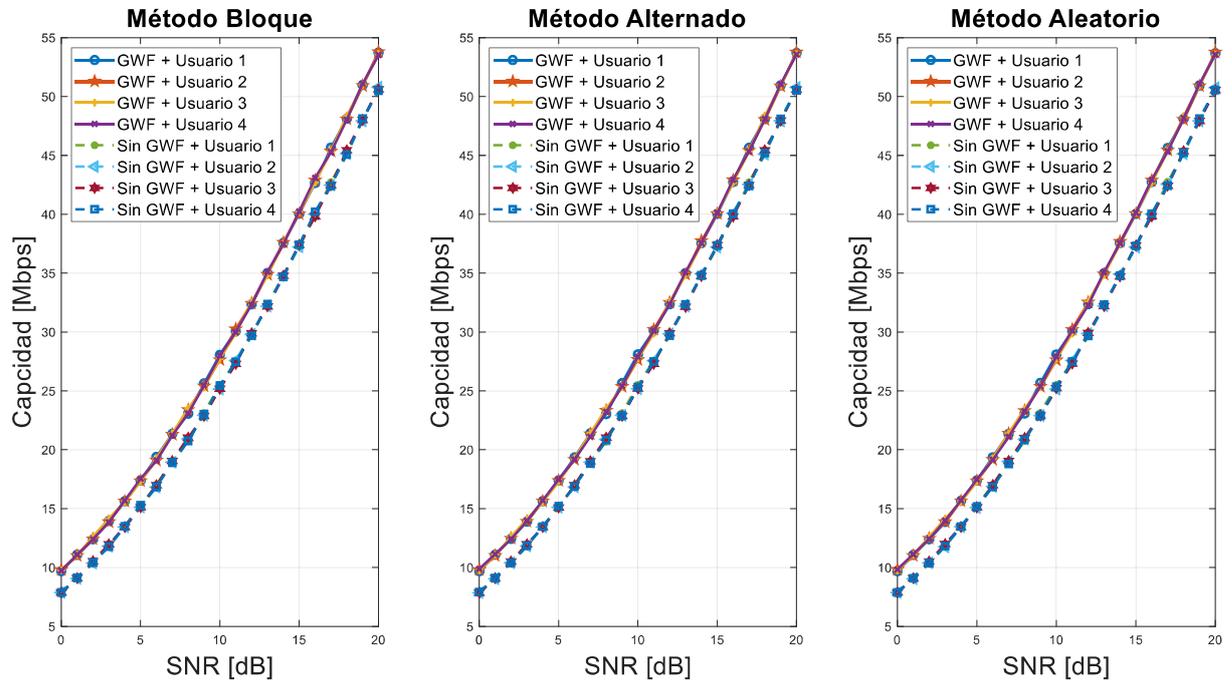
**Figura 3.14.** Capacidad por usuario con modulación QPSK, métodos bloque, alternado y aleatorio para un sistema OFDMA con 4 usuarios sobre un canal de 40 MHz.

En la Figura 3.14 se muestran las gráficas de la capacidad de cada usuario para los tres métodos de asignación de subportadoras utilizando la modulación QPSK. En este caso la mejora de la capacidad al utilizar el algoritmo GWF es similar para los tres métodos de asignación de subportadoras, alcanzando un máximo de 16.54% y un mínimo de 15.63% para el conjunto de usuarios que utilizan la asignación de subportadoras alternadas.

#### 3.4.2.2. Modulación 16-QAM

En la Figura 3.15 se muestran tres gráficas con curvas de la capacidad obtenida con la modulación 16-QAM para los métodos de asignación tipo bloque, aleatorio y alternado. El

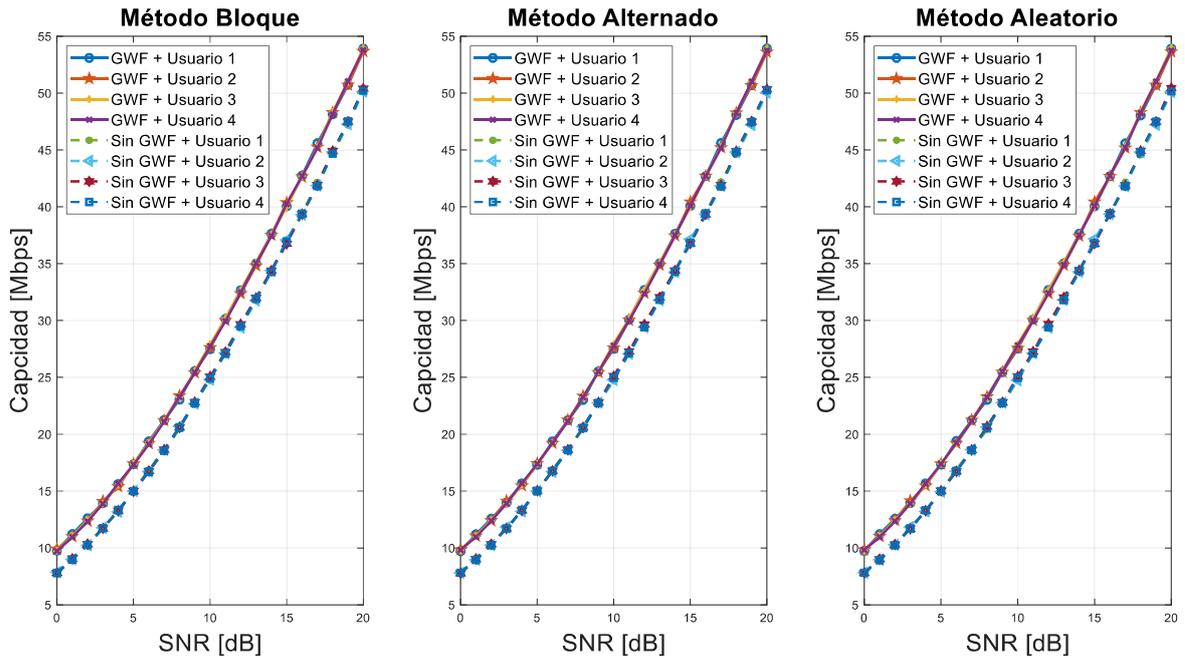
algoritmo GWF mejora la capacidad de los 4 usuarios de manera equitativa alcanzando mejoras cercanas al 24% para cada uno de ellos.



**Figura 3.15.** Capacidad por usuarios con modulación 16-QAM, métodos bloque, alternado y aleatorio para un sistema OFDMA con 4 usuarios sobre un canal de 40 MHz.

### 3.4.2.3. Modulación 64-QAM

En la Figura 3.16 se muestran tres gráficas en la que cada una contiene las curvas de la capacidad obtenida utilizando la modulación 64-QAM para los tres métodos de asignación. En cada gráfica se compara la capacidad de los usuarios con y sin el uso del algoritmo GWF el cual aumenta la capacidad del canal de una mejor manera en comparación con las modulaciones previamente analizadas. En este caso cada usuario que utiliza el algoritmo GWF obtiene un incremento promedio del 25.55% al utilizar el método de asignación de subportadoras alternadas para un valor de SNR de 0 dB, mientras que con el método de asignación tipo bloque se obtiene una mejora promedio de la capacidad de un 25.25% para cada usuario.



**Figura 3.16.** Capacidad por usuario utilizando la modulación 64-QAM para los métodos bloque, alternado y aleatorio para 4 usuarios sobre un canal de 40 MHz.

En el sistema OFDMA propuesto se obtiene una respuesta favorable del uso del algoritmo GWF, el cual consigue aumentar la capacidad de todos los usuarios hasta un 25%. Cada método de asignación de subportadoras trabaja de igual manera con el algoritmo GWF, obteniendo un rendimiento similar con variaciones menores al 1% en lo que respecta a la mejora de la capacidad.

### 3.5. RESULTADOS FINALES

En la Tabla 3.10 se muestra la mejora porcentual de la capacidad total del sistema OFDMA, la cual se obtiene al utilizar el algoritmo GWF con cada tipo de asignación de subportadoras. En promedio para dos y cuatro usuarios, al utilizar el algoritmo GWF con la modulación 64-QAM se obtiene una mejora de la capacidad del 10.30% y 10.40% mientras que con la modulación QPSK se obtiene una mejora del 1.49% y 1.56% respectivamente debido a que dicha modulación es más resistente al ruido y trabaja de mejora manera en ambientes con mucho ruido.

El rango de SNR que se utilizó en la Tabla 3.10 corresponde al nivel medio para establecer y mantener la conexión con un transmisor WLAN que trabaje con el estándar 802.11n [14].

**Tabla 3.10.** Mejora porcentual promedio de la capacidad total del sistema OFDMA de 2 y 4 usuarios para un rango de 8 a 16 dB.

AB	Método de Asignación	2 Usuarios			4 Usuarios		
		QPSK	16-QAM	64-QAM	QPSK	16-QAM	64-QAM
<b>20 MHz</b>	<b>Bloque</b>	1.30%	8.98%	10.23%	1.60%	9.22%	10.35%
	<b>Alternado</b>	1.45%	8.99%	10.22%	1.53%	9.09%	10.32%
	<b>Aleatorio</b>	1.44%	8.96%	10.21%	1.54%	9.08%	10.28%
<b>40 MHz</b>	<b>Bloque</b>	1.51%	9.10%	10.39%	1.55%	9.21%	10.51%
	<b>Alternado</b>	1.52%	9.12%	10.40%	1.59%	9.22%	10.49%
	<b>Aleatorio</b>	1.75%	9.10%	10.37%	1.60%	9.20%	10.45%
<b>Mejora promedio</b>		1.49%	9.03%	10.30%	1.56%	9.17%	10.40%

## 4. CONCLUSIONES Y RECOMENDACIONES

### 3.1. CONCLUSIONES

- Al implementar el algoritmo *geometric water-filling* en un sistema OFDMA se aprovecha el conocimiento del estado del canal obtenido por las subportadoras piloto para realizar la reasignación de potencia a cada subportadora y optimizar la capacidad que puede ofrecer dicho sistema.
- Cada método de asignación de subportadoras obtiene un resultado similar al analizar la capacidad total del sistema OFDMA, pero al comparar la capacidad por usuario, la capacidad alcanzada por cada uno es diferente. Esto se debe a que mientras mayor sea el número de usuarios, aumenta la probabilidad de que les asignen subcanales con diferentes respuestas en frecuencia, alcanzando diferentes capacidades para cada usuario.
- El algoritmo *geometric water-filling* se desempeña de manera eficiente cuando se utiliza con esquemas de modulación como 64-QAM, esto se debe a que en este tipo de modulación se trabaja con una variedad de amplitudes, tanto altas como bajas, siendo las últimas las causantes de un bajo rendimiento sobre canales con una respuesta variante en frecuencia, como es el caso de los canales inalámbricos.
- Para un SNR de 0 dB la mejora promedio de la capacidad de 2 usuarios utilizando el algoritmo GWF con el método de asignación de subportadoras alternadas y la modulación 64-QAM es del 24.26% mientras que para 4 usuarios en las mismas condiciones de SNR, método de asignación de subportadoras y modulación, la mejora promedio de la capacidad es del 25.54%.
- Al aumentar el número de usuarios, se reduce el número de subportadoras asignadas a cada usuario, lo cual reduce la capacidad que se le puede ofrecer. Por este motivo se utiliza el algoritmo *geometric water-filling*, el cual asigna una mayor potencia para los mejores subcanales, aumentando la capacidad de transmisión del usuario.
- Al utilizar el algoritmo GWF con la modulación QPSK se obtiene una mejora máxima del 16.54% para un valor SNR de 0 dB debido a la robustez de la modulación, a diferencia de la modulación 64-QAM en la que se consiguen mejoras de la capacidad de hasta un 26%.

- La eficiencia del algoritmo *geometric water-filling* aumenta mientras menor sea el valor de la SNR del canal, debido a que no todas las subportadoras tienen el mismo valor de SNR. En este caso el algoritmo *geometric water-filling* asigna una mayor potencia a los mejores subcanales, aumentando la capacidad total del sistema.
- El algoritmo GWF aumenta ligeramente su eficiencia al incrementar el ancho de banda del canal y el número de subportadoras ya que dispone de una mayor cantidad de recursos sobre los cuales trabajar.
- Utilizar un canal multitrayecto para cada usuario mejora el desempeño del sistema OFDMA implementado, ya que este parámetro permite simular la posición de cada usuario respecto al transmisor mejorando la confiabilidad de la simulación.
- En las curvas de la capacidad que comparan cada esquema de modulación con los diferentes métodos de asignación de subportadoras, se concluye que mientras menor sea el índice de modulación, el algoritmo *geometric water-filling* reduce su eficiencia, debido a que este tipo de modulación es más robusta en condiciones de bajo SNR.

### **3.2. RECOMENDACIONES**

- Para obtener resultados más confiables se recomienda aumentar el número de iteraciones de la simulación del sistema OFDMA.
- Al presentar los resultados se debe asegurar su comprensión por lo que se recomienda añadir etiquetas a cada figura así como un distintivo para cada curva que se presente en las mismas, así como las unidades y nombre de los ejes con los cuales se está trabajando.
- Como trabajo futuro se puede analizar el comportamiento del algoritmo GWF con una mayor cantidad de usuarios utilizando técnicas de transmisión MIMO.
- En un trabajo futuro se puede realizar la implementación en hardware del algoritmo GWF conjuntamente con la capa física del estándar IEEE 802.11n para verificar las mejoras de la capacidad obtenidas en la simulación.
- Otro trabajo futuro puede ser el análisis de la combinación de modulaciones para la transmisión en un sistema OFDMA utilizando el algoritmo GWF, en el cual los subcanales con SNR bajo transmitan con un esquema de modulación más

robusto, mientras que en valores adecuados de SNR se utilice esquemas de modulación que transmitan una mayor cantidad de información.

## 5. REFERENCIAS BIBLIOGRÁFICAS

- [1] D. A. Zoubi, G. V. Tsoulos y D. I. Kaklamani, "Subcarrier Allocation Algorithms for multicellular OFDMA networks without Channel State Information," de *3rd International Symposium on Wireless Pervasive Computing*, Santorini, Greece, 2008.
- [2] J. Ángeles, "Técnica de modulación OFDM," de *Estudio y simulación de un sistema ACO-OFDM para comunicaciones ópticas inalámbricas*, Sevilla, 2014.
- [3] D. Reinoso y D. Simba, *Análisis Mediante Simulación del Algoritmo de Geometric Water-Filling en Sistemas OFDM*, Quito, Ecuador, 2018.
- [4] S. Sesia, I. Toufik y M. Baker, *LTE – The UMTS Long Term Evolution: From Theory to Practice*, John Wiley & Sons, 2011.
- [5] F. J. Sánchez , "OFDM," de *OFDMA Y SC-FDMA EN LA INTERFAZ RADIO DE LTE*, Sevilla, 2012.
- [6] S. Srikanth, V. Kumaran y C. Manikandan, *Orthogonal Frequency Division Multiple Access: Is it the Multiple Access System of the Future?*, India, 2006.
- [7] V. Hinojosa, A. Mendoza y R. Ibarra, "Evaluación del Funcionamiento de la Capa Física y Capa MAC de las Redes Inalámbricas IEEE 802.11," *Pistas Educativas*, vol. 35, nº 108, 2014.
- [8] Y. S. Cho, K. Jaekwon, Y. Won Young y C.-G. Kang, *MIMO-OFDM Wireless Communications with MATLAB*, John Wiley & Sons, 2010.
- [9] F. J. Sánchez, "Técnicas Acceso Múltiple. OFDMA y SC-FDMA," de *OFDMA Y SC-FDMA EN LA INTERFAZ RADIO DE LTE*, Sevilla, 2012.
- [10] M. A. Alsinella, *Implementación de técnicas de estimación y sincronización*, Ballaterra, 2008.
- [11] KdTerri, "Telecom Source," 03 Octubre 2012. [En línea]. Available: <http://www.telecomsource.net/showthread.php?1435-What-are-fast-fading-and-slow-fading/page2>.
- [12] P. He, L. Zhao, S. Zhou y Z. Niu, *Radio Resource Management Using Geometric Water-Filling*, New York: Springer, 2014.
- [13] Y. Otani, S. Ohno y T. Hinamoto, "Subcarrier Allocation for multi-user OFDM system," de *Asia-Pacific Conference on Communications*, Perth, Western Australia,, 2005.
- [14] J. Geier, "Wireless-Nets," Diciembre 2013. [En línea]. Available: [http://www.wireless-nets.com/resources/tutorials/define\\_SNR\\_values.html](http://www.wireless-nets.com/resources/tutorials/define_SNR_values.html). [Último acceso: Diciembre 2019].

# ANEXOS

## ANEXO A. Código del programa principal

```
1.  %-----PROGRAMA PRINCIPAL-----
2.  clc
3.  clear all
4.  close all
5.  %-----Datos para el Transmisor-----
6.  usuarios=2;      % Numero de usuarios  2 o 4
7.  AB=20e6;        % Ancho de Banda 20MHz o 40 MHz
8.  mod=[2 4 6];    % Bits por simbolo para cada modulaci3n
9.                  % Modulacion QPSK(2), 16QAM(4) O 64QAM(6)
10. norm=[sqrt(2) sqrt(10) sqrt(42)];
11. rep_snr=5000;   % Repeticiones para cada SNR
12. ig=1/4;        % Intervalo de Guarda
13. SNR=0:1:20;    % SNR
14. if AB==20e6
15.     Nsubt=64;
16.     Nsubd=52;
17.     AB_Sub=AB/Nsubt;
18. elseif AB==40e6
19.     Nsubt=128;
20.     Nsubd=108;
21.     AB_Sub=AB/Nsubt;
22. end
23. %Seleccion de modulacion
24. for modlc=1:3
25.     m=mod(modlc);
26.     M=2^m;
27.     n=norm(modlc);
28.     %Seleccion de SNR
29.     for iter=1:length(SNR)
30.         Snr=SNR(iter);
31.         snr_veces=10^(Snr/10);
32.         for rep=1:rep_snr
33.             %-----Transmisor-----
34.                 [Sym_OFDMA_tx_1, Sym_D_1] = TX(Nsubt, n, m, M,
35.                 ig,Nsubd);
36.                 [Sym_OFDMA_tx_2, Sym_D_2] = TX(Nsubt, n, m, M,
37.                 ig,Nsubd);
38.                 if usuarios==4
39.                     [Sym_OFDMA_tx_3, Sym_D_3] = TX(Nsubt, n, m, M,
40.                     ig,Nsubd);
41.                     [Sym_OFDMA_tx_4, Sym_D_4] = TX(Nsubt, n, m, M,
42.                     ig,Nsubd);
43.                 end
44.             %-----Canal-----
45.                 H_D_1 = CANAL_1(Nsubt, Sym_OFDMA_tx_1, Snr);
46.                 H_D_2 = CANAL_2(Nsubt, Sym_OFDMA_tx_2, Snr);
47.                 if usuarios==4
48.                     H_D_3 = CANAL_3(Nsubt, Sym_OFDMA_tx_3, Snr);
49.                     H_D_4 = CANAL_4(Nsubt, Sym_OFDMA_tx_4, Snr);
50.                 end
51.             %-----Asignacion de recursos-----
52.                 DSM= randperm(Nsubd,Nsubd);
```

```

49.         H=zeros(3,Nsubd);
50.         Sym_D=zeros(3,Nsubd);
51.         if usuarios == 2
52.             as_1=[1:Nsubd/2; Nsubd/2+1:Nsubd];
53.             as_2=[1:2:Nsubd; 2:2:Nsubd];
54.             as_3=[DSM(as_1(1,:)); DSM(as_1(2,:))];
55.             %H(asignacion,:)
56.             H(1,:)=[H_D_1(as_1(1,:)) H_D_2(as_1(2,:))];
57.             H(2,:)=[H_D_1(as_2(1,:)) H_D_2(as_2(2,:))];
58.             H(3,:)=[H_D_1(as_3(1,:)) H_D_2(as_3(2,:))];
59.             Sym_D(1,:)=[Sym_D_1(as_1(1,:))
Sym_D_2(as_1(2,:))];
60.             Sym_D(2,:)=[Sym_D_1(as_2(1,:))
Sym_D_2(as_2(2,:))];
61.             Sym_D(3,:)=[Sym_D_1(as_3(1,:))
Sym_D_2(as_3(2,:))];
62.         elseif usuarios == 4
63.             as_1=[1:Nsubd/4; Nsubd/4+1:Nsubd/2;
Nsubd/2+1:Nsubd*3/4; Nsubd*3/4+1:Nsubd];
64.             as_2=[1:4:Nsubd; 2:4:Nsubd; 3:4:Nsubd;
4:4:Nsubd];
65.             as_3=[DSM(as_1(1,:)); DSM(as_1(2,:));
DSM(as_1(3,:)); DSM(as_1(4,:))];
66.             %H(asignacion,:)
67.             H(1,:)=[H_D_1(as_1(1,:)) H_D_2(as_1(2,:))
H_D_3(as_1(3,:)) H_D_4(as_1(4,:))];
68.             H(2,:)=[H_D_1(as_2(1,:)) H_D_2(as_2(2,:))
H_D_3(as_2(3,:)) H_D_4(as_2(4,:))];
69.             H(3,:)=[H_D_1(as_3(1,:)) H_D_2(as_3(2,:))
H_D_3(as_3(3,:)) H_D_4(as_3(4,:))];
70.             Sym_D(1,:)=[Sym_D_1(as_1(1,:)) Sym_D_2(as_1(2,:))
Sym_D_3(as_1(3,:)) Sym_D_4(as_1(4,:))];
71.             Sym_D(2,:)=[Sym_D_1(as_2(1,:)) Sym_D_2(as_2(2,:))
Sym_D_3(as_2(3,:)) Sym_D_4(as_2(4,:))];
72.             Sym_D(3,:)=[Sym_D_1(as_3(1,:)) Sym_D_2(as_3(2,:))
Sym_D_3(as_3(3,:)) Sym_D_4(as_3(4 ,:))];
73.         end
74.         %-----GWF-----
75.         for t=1:3
76.             Ptx_D(t,:)=sum(abs(Sym_D(t,:).^2));
77.             ruido(t,:)=(Ptx_D(t,:)/snr_veces)/Nsubt;
78.             [PA(t,:)] = GWF (Nsubd, Ptx_D(t,:), H(t,:),
ruido(t,:));
79.         end
80.         %-----Capacidad Para Multiples Usuarios-----
81.         [cap, cap_gwf] = CAPACIDAD(usuarios, H, Sym_D,
ruido, PA, as_1);
82.         CAP_GWF(modlc,rep,:)=cap_gwf(:, :, :);
83.         CAP(modlc,rep,:)=cap(:, :, :);
84.         end
85.         capGWF(modlc,iter,:)=AB_Sub*sum(CAP_GWF(modlc, :, :))/
(1e6*rep_snr);
86.         capM(modlc,iter,:)=AB_Sub*sum(CAP(modlc, :, :))/
(1e6*rep_snr);
87.         end

```

```
88. %-----Respaldo de las variables-----
89.     if usuarios==2
90.         if Nsubt==64
91.             save('Capacidades_2_64', 'SNR', 'capM', 'capGWF')
92.         elseif Nsubt==128
93.             save('Capacidades_2_128', 'SNR', 'capM', 'capGWF')
94.         end
95.     elseif usuarios==4
96.         if Nsubt==64
97.             save('Capacidades_4_64', 'SNR', 'capM', 'capGWF')
98.         elseif Nsubt==128
99.             save('Capacidades_4_128', 'SNR', 'capM', 'capGWF')
100.        end
101.    end
102. end
```

## ANEXO B. Función TX

```
1. function [Sym_OFDMA_tx, Sym_D] = TX(Nsubt, n, m, M, ig,
   Nsubd)
2. %Datos
3. Datos_tx=randi([0 1], 1, Nsubd*m);
4. %Conversion serie paralelo
5. datos_tx_par=reshape(Datos_tx,Nsubd*m,1);
6. %Modulacion y normalizacion
7. datos_tx_mod_nn=qammod(datos_tx_par, M,'InputType','bit');
8. datos_tx_mod=datos_tx_mod_nn./n;
9. Sym_D=datos_tx_mod';
10. %Simbolo OFDM en base al estandar 802.11n
11. Sym=zeros(Nsubt,1);
12. if Nsubt==64
13.     %Señales piloto en las subportadoras -21,-7, 7 y 21
14.     Sym(12)=1;
15.     Sym(26)=1;
16.     Sym(39)=1;
17.     Sym(53)=1;
18.     %Datos en las subportadoras de la -28 a la -1 y de la 1 a
   la 28
19.     Sym(5:11)=datos_tx_mod(27:33);
20.     Sym(13:25)=datos_tx_mod(34:46);
21.     Sym(27:32)=datos_tx_mod(47:52);
22.     Sym(33:38)=datos_tx_mod(1:6);
23.     Sym(40:52)=datos_tx_mod(7:19);
24.     Sym(54:60)=datos_tx_mod(20:26);
25. elseif Nsubt==128
26.     %Señales piloto en las subportadoras -53, -25, -11, 11,
   25, 53
27.     Sym(12)=1;
28.     Sym(40)=1;
29.     Sym(54)=1;
30.     Sym(75)=1;
31.     Sym(89)=1;
32.     Sym(117)=1;
33.     %Datos en las subportadoras de la -58 a la -2 y de la 2 a
   la 58
34.     Sym(7:11)=datos_tx_mod(55:59);
35.     Sym(13:39)=datos_tx_mod(60:86);
36.     Sym(41:53)=datos_tx_mod(87:99);
37.     Sym(55:63)=datos_tx_mod(100:108);
38.     Sym(66:74)=datos_tx_mod(1:9);
39.     Sym(76:88)=datos_tx_mod(10:22);
40.     Sym(90:116)=datos_tx_mod(23:49);
41.     Sym(118:122)=datos_tx_mod(50:54);
42. end
43. %IFFT
44. Sym_ifft=ifft(Sym,Nsubt);
45. %Insercion del prefijo ciclico en el intervalo de guarda
46. Sym_OFDMA=[Sym_ifft(Nsubt-Nsubt*ig+1:end); Sym_ifft];
47. Sym_OFDMA_tx=reshape(Sym_OFDMA,1,length(Sym_OFDMA));
48. end
```

## ANEXO C. Función CANAL\_1

```
1. function [H_D] = CANAL_1(Nsubt, Sym_OFDMA_tx, Snr)
2. %Parametros del canal multitrayecto (PDP)
3. p_rayos = [0 -3 -6]; % Potencia de los rayos
   en dB
4. retardos= [0 6 12]; % Retardos de los rayos
5. N_rayos=length(retardos); % Numero de rayos
6. p_rayos_veces = 10.^(p_rayos/10); % Potencia de los rayos
   en veces
7. p_pm=p_rayos_veces/sum(p_rayos_veces); % Potencia promedio de
   los rayos
8. p_pm_n=sqrt(p_pm); % Potencia promedio de
   los rayos normalizada
9. %Canal Rayleigh
10. canal=(randn(1,N_rayos)+i*randn(1,N_rayos))./sqrt(2);
11. h_t=canal.*p_pm_n;
12. h=zeros(1,Nsubt);
13. h(1,retardos+1)=h_t; % Canal
14. Sym_OFDMA_h=conv(Sym_OFDMA_tx,h); % Respuesta del Simbolo
   OFDMA en el canal
15. H=fft(h,Nsubt); % Respuesta en frecuencia
   del canal
16. if Nsubt==64
17.     H_D(27:33)= H(5:11);
18.     H_D(34:46)= H(13:25);
19.     H_D(47:52)= H(27:32);
20.     H_D(1:6)= H(33:38);
21.     H_D(7:19)= H(40:52);
22.     H_D(20:26)= H(54:60);
23. elseif Nsubt==128
24.     H_D(55:59)=H(7:11);
25.     H_D(60:86)=H(13:39);
26.     H_D(87:99)=H(41:53);
27.     H_D(100:108)=H(55:63);
28.     H_D(1:9)=H(66:74);
29.     H_D(10:22)=H(76:88);
30.     H_D(23:49)=H(90:116);
31.     H_D(50:54)=H(118:122);
32. end
33. %Senal con ruido AWGN con una relacion de Eb/No
34. Sym_OFDMA_N=awgn(Sym_OFDMA_h,Snr,'measured');
35. end
```

## ANEXO D. Función CANAL\_2

```
1. function [H_D] = CANAL_2(Nsubt, Sym_OFDMA_tx, Snr)
2. %Parametros del canal de desvanecimiento
3. p_rayos = [0 -3 -6 -9] ; % Potencia de los rayos en dB
4. retardos= [0 3 6 9]; % Retardos de los rayos
5. N_rayos=length(retardos); % Numero de rayos
6. p_rayos_veces = 10.^(p_rayos/10); % Potencia de los rayos
   en veces
7. p_pm=p_rayos_veces/sum(p_rayos_veces);% Potencia promedio de
   los rayos
8. p_pm_n=sqrt(p_pm); % Potencia promedio de
   los rayos normalizada
9. %Canal Rayleigh
10. canal=(randn(1,N_rayos)+i*randn(1,N_rayos))./sqrt(2);
11. h_t=canal.*p_pm_n;
12. h=zeros(1,Nsubt);
13. h(1,retardos+1)=h_t; % Canal
14. Sym_OFDMA_h=conv(Sym_OFDMA_tx,h); % Respuesta del Simbolo
   OFDMA en el canal
15. H=fft(h,Nsubt); % Respuesta en
   frecuencia del canal
16. if Nsubt==64
17.     H_D(27:33)= H(5:11);
18.     H_D(34:46)= H(13:25);
19.     H_D(47:52)= H(27:32);
20.     H_D(1:6)= H(33:38);
21.     H_D(7:19)= H(40:52);
22.     H_D(20:26)= H(54:60);
23. elseif Nsubt==128
24.     H_D(55:59)=H(7:11);
25.     H_D(60:86)=H(13:39);
26.     H_D(87:99)=H(41:53);
27.     H_D(100:108)=H(55:63);
28.     H_D(1:9)=H(66:74);
29.     H_D(10:22)=H(76:88);
30.     H_D(23:49)=H(90:116);
31.     H_D(50:54)=H(118:122);
32. end
33. %Senal con ruido AWGN con una relacion de Eb/No
34. Sym_OFDMA_N=awgn(Sym_OFDMA_h,Snr,'measured');
35. end
```

## ANEXO E. Función CANAL\_3

```
1. function [H_D] = CANAL_3(Nsubt, Sym_OFDMA_tx, Snr)
2. %Parametros del canal de desvanecimiento
3. p_rayos = [0 0]; % Potencia de los rayos en dB
4. retardos= [0 5]; % Retardos de los rayos
5. N_rayos=length(retardos); % Numero de rayos
6. p_rayos_veces = 10.^(p_rayos/10); % Potencia de los rayos
   en veces
7. p_pm=p_rayos_veces/sum(p_rayos_veces);% Potencia promedio de
   los rayos
8. p_pm_n=sqrt(p_pm); % Potencia promedio de
   los rayos normalizada
9. %Canal Rayleigh
10. canal=(randn(1,N_rayos)+i*randn(1,N_rayos))./sqrt(2);
11. h_t=canal.*p_pm_n;
12. h=zeros(1,Nsubt);
13. h(1,retardos+1)=h_t; % Canal
14. Sym_OFDMA_h=conv(Sym_OFDMA_tx,h); % Respuesta del Simbolo
   OFDMA en el canal
15. H=fft(h,Nsubt); % Respuesta en
   frecuencia del canal
16. if Nsubt==64
17.     H_D(27:33)= H(5:11);
18.     H_D(34:46)= H(13:25);
19.     H_D(47:52)= H(27:32);
20.     H_D(1:6)= H(33:38);
21.     H_D(7:19)= H(40:52);
22.     H_D(20:26)= H(54:60);
23. elseif Nsubt==128
24.     H_D(55:59)=H(7:11);
25.     H_D(60:86)=H(13:39);
26.     H_D(87:99)=H(41:53);
27.     H_D(100:108)=H(55:63);
28.     H_D(1:9)=H(66:74);
29.     H_D(10:22)=H(76:88);
30.     H_D(23:49)=H(90:116);
31.     H_D(50:54)=H(118:122);
32. end
33. %Senal con ruido AWGN con una relacion de Eb/No
34. Sym_OFDMA_N=awgn(Sym_OFDMA_h,Snr, 'measured');
35. end
```

## ANEXO F. Función CANAL\_4

```
1. function [H_D] = CANAL_4(Nsubt, Sym_OFDMA_tx, Snr)
2. %Parametros del canal de desvanecimiento
3. p_rayos = [0 -3 -6 -9]; % Potencia de los rayos en dB
4. retardos= [0 2 4 6]; % Retardos de los rayos
5. N_rayos=length(retardos); % Numero de rayos
6. p_rayos_veces = 10.^(p_rayos/10); % Potencia de los rayos
   en veces
7. p_pm=p_rayos_veces/sum(p_rayos_veces);% Potencia promedio de
   los rayos
8. p_pm_n=sqrt(p_pm); % Potencia promedio de
   los rayos normalizada
9. %Canal Rayleigh
10. canal=(randn(1,N_rayos)+i*randn(1,N_rayos))./sqrt(2);
11. h_t=canal.*p_pm_n;
12. h=zeros(1,Nsubt);
13. h(1,retardos+1)=h_t; % Canal
14. Sym_OFDMA_h=conv(Sym_OFDMA_tx,h); % Respuesta del Simbolo
   OFDMA en el canal
15. H=fft(h,Nsubt); % Respuesta en
   frecuencia del canal
16. if Nsubt==64
17.     H_D(27:33)= H(5:11);
18.     H_D(34:46)= H(13:25);
19.     H_D(47:52)= H(27:32);
20.     H_D(1:6)= H(33:38);
21.     H_D(7:19)= H(40:52);
22.     H_D(20:26)= H(54:60);
23. elseif Nsubt==128
24.     H_D(55:59)=H(7:11);
25.     H_D(60:86)=H(13:39);
26.     H_D(87:99)=H(41:53);
27.     H_D(100:108)=H(55:63);
28.     H_D(1:9)=H(66:74);
29.     H_D(10:22)=H(76:88);
30.     H_D(23:49)=H(90:116);
31.     H_D(50:54)=H(118:122);
32. end
33. %Senal con ruido AWGN con una relacion de Eb/No
34. Sym_OFDMA_N=awgn(Sym_OFDMA_h,Snr, 'measured');
35. end
```

## ANEXO G. Función GWF

```
1. function [PA] = GWF(Nsubd, Ptx, H_D, ruido)
2. H=abs(H_D);
3. canal_datos=H;
4. B=abs(canal_datos).^2/ruido; %ganancia del canal
5. [a,I]=sort(B,'descend');
6. d=1./a;
7. P2=Ptx;
8. Ptxvc=Ptx;
9. k=2;
10. while (P2>0) && (k<=Nsubd)
11.     suma=0;
12.     for i=1:k
13.         suma=suma+(d(k)-d(i));
14.     end
15.     P2=Ptxvc-suma;
16.     if (P2<0)
17.         P2=0;
18.     else
19.         P2k_ast=P2;
20.         k_ast=k;
21.         k=k+1;
22.     end
23. end
24. Sk_ast=P2k_ast/k_ast;
25. S=zeros(1,Nsubd);

26. for i=1:k_ast
27.     S(i)=Sk_ast+d(k_ast)-d(i); %potencia del escalon
    determinada
28. end
29. Pasig(I)=S; %Restaura el orden original de
    las potencias
30. PA=zeros(1,Nsubd);
31. PA= Pasig; %Nueva potencia asignada
32. end
```

## ANEXO H. Función CAPACIDAD

```
1. function [cap, cap_gwf] = CAPACIDAD(usuarios, H_D, Sym_D,
   ruido, PA, as_1)
2. if usuarios == 2
3.     %Método de subportadoras adyacentes
4.
   cap(1,:,1)=sum(log2(1+abs(H_D(1,as_1(1,:)).^2).*abs(Sym_D(1,as_
   1(1,:)).^2)/ruido(1,:)));
5.
   cap(1,:,4)=sum(log2(1+abs(H_D(1,as_1(2,:)).^2).*abs(Sym_D(1,as_
   1(2,:)).^2)/ruido(1,:)));
6.
   cap_gwf(1,:,1)=sum(log2(1+abs(H_D(1,as_1(1,:)).^2).*PA(1,as_1(1
   ,:))/ruido(1,:)));
7.
   cap_gwf(1,:,4)=sum(log2(1+abs(H_D(1,as_1(2,:)).^2).*PA(1,as_1(2
   ,:))/ruido(1,:)));
8.     %Método de subportadoras alternadas
9.
   cap(1,:,2)=sum(log2(1+abs(H_D(2,as_1(1,:)).^2).*abs(Sym_D(2,as_
   1(1,:)).^2)/ruido(2,:)));
10.
   cap(1,:,5)=sum(log2(1+abs(H_D(2,as_1(2,:)).^2).*abs(Sym_D(2,as_
   1(2,:)).^2)/ruido(2,:)));
11.
   cap_gwf(1,:,2)=sum(log2(1+abs(H_D(2,as_1(1,:)).^2).*PA(2,as_1(1
   ,:))/ruido(2,:)));
12.
   cap_gwf(1,:,5)=sum(log2(1+abs(H_D(2,as_1(2,:)).^2).*PA(2,as_1(2
   ,:))/ruido(2,:)));
13.     %Método de diversidad de subportadoras
14.
   cap(1,:,3)=sum(log2(1+abs(H_D(3,as_1(1,:)).^2).*abs(Sym_D(3,as_
   1(1,:)).^2)/ruido(3,:)));
15.
   cap(1,:,6)=sum(log2(1+abs(H_D(3,as_1(2,:)).^2).*abs(Sym_D(3,as_
   1(2,:)).^2)/ruido(3,:)));
16.
   cap_gwf(1,:,3)=sum(log2(1+abs(H_D(3,as_1(1,:)).^2).*PA(3,as_1(1
   ,:))/ruido(3,:)));
17.
   cap_gwf(1,:,6)=sum(log2(1+abs(H_D(3,as_1(2,:)).^2).*PA(3,as_1(2
   ,:))/ruido(3,:)));
18. elseif usuarios == 4
19.     %Método de subportadoras adyacentes
20.
   cap(1,:,1)=sum(log2(1+abs(H_D(1,as_1(1,:)).^2).*abs(Sym_D(1,as_
   1(1,:)).^2)/ruido(1,:)));
21.
   cap(1,:,4)=sum(log2(1+abs(H_D(1,as_1(2,:)).^2).*abs(Sym_D(1,as_
   1(2,:)).^2)/ruido(1,:)));
```

```

22.
    cap(1,:,7)=sum(log2(1+abs(H_D(1,as_1(3,:)).^2).*abs(Sym_D(1,as_
    1(3,:)).^2)/ruido(1,:)));
23.
    cap(1,:,10)=sum(log2(1+abs(H_D(1,as_1(4,:)).^2).*abs(Sym_D(1,as
    _1(4,:)).^2)/ruido(1,:)));
24.
    cap_gwf(1,:,1)=sum(log2(1+abs(H_D(1,as_1(1,:)).^2).*PA(1,as_1(1
    ,:))/ruido(1,:)));
25.
    cap_gwf(1,:,4)=sum(log2(1+abs(H_D(1,as_1(2,:)).^2).*PA(1,as_1(2
    ,:))/ruido(1,:)));
26.
    cap_gwf(1,:,7)=sum(log2(1+abs(H_D(1,as_1(3,:)).^2).*PA(1,as_1(3
    ,:))/ruido(1,:)));
27.
    cap_gwf(1,:,10)=sum(log2(1+abs(H_D(1,as_1(4,:)).^2).*PA(1,as_1(
    4,:))/ruido(1,:)));
28.
    %Método de subportadoras alternadas
29.
    cap(1,:,2)=sum(log2(1+abs(H_D(2,as_1(1,:)).^2).*abs(Sym_D(2,as_
    1(1,:)).^2)/ruido(2,:)));
30.
    cap(1,:,5)=sum(log2(1+abs(H_D(2,as_1(2,:)).^2).*abs(Sym_D(2,as_
    1(2,:)).^2)/ruido(2,:)));
31.
    cap(1,:,8)=sum(log2(1+abs(H_D(2,as_1(3,:)).^2).*abs(Sym_D(2,as_
    1(3,:)).^2)/ruido(2,:)));
32.
    cap(1,:,11)=sum(log2(1+abs(H_D(2,as_1(4,:)).^2).*abs(Sym_D(2,as
    _1(4,:)).^2)/ruido(2,:)));
33.
    cap_gwf(1,:,2)=sum(log2(1+abs(H_D(2,as_1(1,:)).^2).*PA(2,as_1(1
    ,:))/ruido(2,:)));
34.
    cap_gwf(1,:,5)=sum(log2(1+abs(H_D(2,as_1(2,:)).^2).*PA(2,as_1(2
    ,:))/ruido(2,:)));
35.
    cap_gwf(1,:,8)=sum(log2(1+abs(H_D(2,as_1(3,:)).^2).*PA(2,as_1(3
    ,:))/ruido(2,:)));
36.
    cap_gwf(1,:,11)=sum(log2(1+abs(H_D(2,as_1(4,:)).^2).*PA(2,as_1(
    4,:))/ruido(2,:)));
37.
    %Método de diversidad de subportadoras
38.
    cap(1,:,3)=sum(log2(1+abs(H_D(3,as_1(1,:)).^2).*abs(Sym_D(3,as_
    1(1,:)).^2)/ruido(3,:)));
39.
    cap(1,:,6)=sum(log2(1+abs(H_D(3,as_1(2,:)).^2).*abs(Sym_D(3,as_
    1(2,:)).^2)/ruido(3,:)));
40.
    cap(1,:,9)=sum(log2(1+abs(H_D(3,as_1(3,:)).^2).*abs(Sym_D(3,as_
    1(3,:)).^2)/ruido(3,:)));

```

```

41.
    cap(1, :, 12) = sum(log2(1 + abs(H_D(3, as_1(4, :)).^2) .* abs(Sym_D(3, as_
    _1(4, :)).^2) / ruido(3, :)));
42.
    cap_gwf(1, :, 3) = sum(log2(1 + abs(H_D(3, as_1(1, :)).^2) .* PA(3, as_1(1
    , :)) / ruido(3, :)));
43.
    cap_gwf(1, :, 6) = sum(log2(1 + abs(H_D(3, as_1(2, :)).^2) .* PA(3, as_1(2
    , :)) / ruido(3, :)));
44.
    cap_gwf(1, :, 9) = sum(log2(1 + abs(H_D(3, as_1(3, :)).^2) .* PA(3, as_1(3
    , :)) / ruido(3, :)));
45.
    cap_gwf(1, :, 12) = sum(log2(1 + abs(H_D(3, as_1(4, :)).^2) .* PA(3, as_1(
    4, :)) / ruido(3, :)));
46. end
47. end

```

## ANEXO I. Función GRAFICA\_2\_Usuarios

```
1. function [ ] = GRAFICA_2_Usuarios()
2. close all
3. clear all
4. clc
5. Nsubt=128;
6. if Nsubt==64
7.     load Capacidades_2_64.mat
8. elseif Nsubt==128
9.     load Capacidades_2_128.mat
10. end
11. CapT(:,:,1)=capM(:,:,1)+capM(:,:,4);
12. CapT(:,:,2)=capM(:,:,2)+capM(:,:,5);
13. CapT(:,:,3)=capM(:,:,3)+capM(:,:,6);
14. CapT_GWF(:,:,1)=capGWF(:,:,1)+capGWF(:,:,4);
15. CapT_GWF(:,:,2)=capGWF(:,:,2)+capGWF(:,:,5);
16. CapT_GWF(:,:,3)=capGWF(:,:,3)+capGWF(:,:,6);
17. %Capacidad (Modulacion , , Usuario)
18. %Modulacion: 1 QPSK, 2 16-QAM y 3 para 64-QAM
19. %Usuarios:   Usuario 1: 1,2,3
20. %             Usuario 2: 4,5,6
21. %% 16QAM total con y sin GWF
22. figure(1)
23. plot(SNR,CapT_GWF(2,:,1),'-o', SNR,CapT_GWF(2,:,2),'-
    p',SNR,CapT_GWF(2,:,3),'-*')
24. hold on
25. plot(SNR,CapT(2,:,1),'--*',SNR,CapT(2,:,2),'--<',
    SNR,CapT(2,:,3),'--+')
26. grid on
27. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
28. legend('GWF + Metodo Bloque','GWF + Metodo Alternado','GWF +
    Metodo Aleatorio','Sin GWF + Metodo Bloque','Sin GWF + Metodo
    Alternado','Sin GWF + Metodo Aleatorio',
    'Location','northwest')
29. title('Capacidad Total del Sistema (16-QAM)')

30. %% 64QAM total con y sin GWF
31. figure(2)
32. plot(SNR,CapT_GWF(3,:,1),'-o', SNR,CapT_GWF(3,:,2),'-
    p',SNR,CapT_GWF(3,:,3),'-*')
33. hold on
34. plot(SNR,CapT(3,:,1),'--*',SNR,CapT(3,:,2),'--<',
    SNR,CapT(3,:,3),'--+')
35. grid on
36. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
37. legend('GWF + Metodo Bloque','GWF + Metodo Alternado','GWF +
    Metodo Aleatorio','Sin GWF + Metodo Bloque','Sin GWF + Metodo
    Alternado','Sin GWF + Metodo Aleatorio',
    'Location','northwest')
38. title('Capacidad Total del Sistema (64-QAM)')

39. %% 16QAM usuario 1/2 con y sin GWF
40. figure(3)
```

```

41. subplot(1,3,1)
42. plot(SNR,capGWF(2,:,1),'-o',SNR,capGWF(2,:,4),'-p')
43. hold on
44. plot(SNR,capM(2,:,1),'--*',SNR,capM(2,:,4),'--<')
45. grid on
46. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
47. legend('GWF + Usuario 1','GWF + Usuario 2','Sin GWF + Usuario
  1','Sin GWF + Usuario 2','Location','northwest')
48. title('Capacidad por Usuario Metodo Bloque(16-QAM)')

49. subplot(1,3,2)
50. plot(SNR,capGWF(2,:,2),'-o',SNR,capGWF(2,:,5),'-p')
51. hold on
52. plot(SNR,capM(2,:,2),'--*',SNR,capM(2,:,5),'--<')
53. grid on
54. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
55. legend('GWF + Usuario 1','GWF + Usuario 2','Sin GWF + Usuario
  1','Sin GWF + Usuario 2','Location','northwest')
56. title('Capacidad por Usuario Metodo Alternado(16-QAM)')

57. subplot(1,3,3)
58. plot(SNR,capGWF(2,:,3),'-o',SNR,capGWF(2,:,6),'-p')
59. hold on
60. plot(SNR,capM(2,:,3),'--*',SNR,capM(2,:,6),'--<')
61. grid on
62. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
63. legend('GWF + Usuario 1','GWF + Usuario 2','Sin GWF + Usuario
  1','Sin GWF + Usuario 2', title('Capacidad por Usuario Metodo
  Aleatorio(16-QAM)')

64. %% 64QAM usuario 1/2 con y sin GWF
65. figure(4)
66. subplot(1,3,1)
67. plot(SNR,capGWF(3,:,1),'-o',SNR,capGWF(3,:,4),'-p')
68. hold on
69. plot(SNR,capM(3,:,1),'--*',SNR,capM(3,:,4),'--<')
70. grid on
71. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
72. legend('GWF + Usuario 1','GWF + Usuario 2','Sin GWF + Usuario
  1','Sin GWF + Usuario 2','Location','northwest')
73. title('Capacidad por Usuario Metodo Bloque(64-QAM)')

74. subplot(1,3,2)
75. plot(SNR,capGWF(3,:,2),'-o',SNR,capGWF(3,:,5),'-p')
76. hold on
77. plot(SNR,capM(3,:,2),'--*',SNR,capM(3,:,5),'--<')
78. grid on
79. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
80. legend('GWF + Usuario 1','GWF + Usuario 2','Sin GWF + Usuario
  1','Sin GWF + Usuario 2','Location','northwest')
81. title('Capacidad por Usuario Metodo Alternado(64-QAM)')

82. subplot(1,3,3)
83. plot(SNR,capGWF(3,:,3),'-o',SNR,capGWF(3,:,6),'-p')
84. hold on

```

```

85. plot(SNR, capM(3, :, 3), '--*', SNR, capM(3, :, 6), '--<')
86. grid on
87. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
88. legend('GWF + Usuario 1', 'GWF + Usuario 2', 'Sin GWF + Usuario
1', 'Sin GWF + Usuario 2', 'Location', 'northwest')
89. title('Capacidad por Usuario Metodo Aleatorio(64-QAM)')

90. %% Total con y sin GWF
91. figure(5)
92. subplot(1,3,1)
93. plot(SNR, CapT_GWF(1, :, 1), '-o', SNR, CapT_GWF(1, :, 2), '-
p', SNR, CapT_GWF(1, :, 3), '-*')
94. hold on
95. plot(SNR, CapT(1, :, 1), '--*', SNR, CapT(1, :, 2), '--<',
SNR, CapT(1, :, 3), '--+')
96. grid on
97. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
98. legend('GWF + Metodo Bloque', 'GWF + Metodo Alternado', 'GWF +
Metodo Aleatorio', 'Sin GWF + Metodo Bloque', 'Sin GWF + Metodo
Alternado', 'Sin GWF + Metodo Aleatorio', 'Location', 'northwest')
99. title('Capacidad Total del Sistema (QPSK)')

100. subplot(1,3,2)
101. plot(SNR, CapT_GWF(2, :, 1), '-o', SNR, CapT_GWF(2, :, 2), '-
p', SNR, CapT_GWF(2, :, 3), '-*')
102. hold on
103. plot(SNR, CapT(2, :, 1), '--*', SNR, CapT(2, :, 2), '--<',
SNR, CapT(2, :, 3), '--+')
104. grid on
105. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
106. legend('GWF + Metodo Bloque', 'GWF + Metodo Alternado', 'GWF +
Metodo Aleatorio', 'Sin GWF + Metodo Bloque', 'Sin GWF + Metodo
Alternado', 'Sin GWF + Metodo Aleatorio',
'Location', 'northwest')
107. title('Capacidad Total del Sistema (16-QAM)')

108. subplot(1,3,3)
109. plot(SNR, CapT_GWF(3, :, 1), '-o', SNR, CapT_GWF(3, :, 2), '-
p', SNR, CapT_GWF(3, :, 3), '-*')
110. hold on
111. plot(SNR, CapT(3, :, 1), '--*', SNR, CapT(3, :, 2), '--<',
SNR, CapT(3, :, 3), '--+')
112. grid on
113. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
114. legend('GWF + Metodo Bloque', 'GWF + Metodo Alternado', 'GWF +
Metodo Aleatorio', 'Sin GWF + Metodo Bloque', 'Sin GWF + Metodo
Alternado', 'Sin GWF + Metodo Aleatorio',
'Location', 'northwest')
115. title('Capacidad Total del Sistema (64-QAM)')

116. %% Total con y sin GWF 1 metodo, todas las modulaciones
117. figure(6)
118. plot(SNR, CapT_GWF(1, :, 3), '->', SNR, CapT_GWF(2, :, 3), '-
*', SNR, CapT_GWF(3, :, 3), '-o')
119. hold on

```

```
120. plot(SNR,CapT(1,:,3),'--p',SNR,CapT(2,:,3),'--<',
        SNR,CapT(3,:,3),'--+')
121. grid on
122. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
123. legend('GWF + QPSK','GWF + 16QAM','GWF + 64QAM','Sin GWF +
        QPSK','Sin GWF + 16QAM', 'Sin GWF +
        64QAM','Location','northwest')
124. title('Capacidad Total Asignacion Alternada')
```

## ANEXO J. Función GRAFICA\_4\_Usuarios

```
1. function [ ] = GRAFICA_4_Usuarios()
2. close all
3. clear all
4. clc
5. Nsubt=128;
6. if Nsubt==64
7.     load Capacidades_4_64.mat
8. elseif Nsubt==128
9.     load Capacidades_4_128.mat
10. end
11. %Capacidad Total
12. CapT(:,:,1)=capM(:,:,1)+capM(:,:,4)+capM(:,:,5)+capM(:,:,10);
13. CapT(:,:,2)=capM(:,:,2)+capM(:,:,5)+capM(:,:,8)+capM(:,:,11);
14. CapT(:,:,3)=capM(:,:,3)+capM(:,:,6)+capM(:,:,9)+capM(:,:,12);
15. CapT_GWF(:,:,1)=capGWF(:,:,1)+capGWF(:,:,4)+capGWF(:,:,7)+cap
    GWF(:,:,10);
16. CapT_GWF(:,:,2)=capGWF(:,:,2)+capGWF(:,:,5)+capGWF(:,:,8)+cap
    GWF(:,:,11);
17. CapT_GWF(:,:,3)=capGWF(:,:,3)+capGWF(:,:,6)+capGWF(:,:,9)+cap
    GWF(:,:,12);
18. %Capacidad (Modulacion , , Usuario)
19. %Modulacion: 1 QPSK, 2 16-QAM y 3 para 64-QAM
20. %Usuarios: Usuario 1: 1,2,3
21. % Usuario 2: 4,5,6
22. % Usuario 3: 7,8,9
23. % Usuario 4: 10,11,12
24. %% 16QAM total con y sin GWF
25. figure(1)
26. plot(SNR,CapT_GWF(2,:,1),'-o', SNR,CapT_GWF(2,:,2),'-
    p',SNR,CapT_GWF(2,:,3),'-*')
27. hold on
28. plot(SNR,CapT(2,:,1),'--*',SNR,CapT(2,:,2),'--<',
    SNR,CapT(2,:,3),'--+')
29. grid on
30. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
31. legend('GWF + Metodo Bloque','GWF + Metodo Alternado','GWF +
    Metodo Aleatorio','Sin GWF + Metodo Bloque','Sin GWF + Metodo
    Alternado','Sin GWF + Metodo Aleatorio','Location','northwest')
32. title('Capacidad Total del Sistema (16-QAM)')

33. %% 64QAM total con y sin GWF
34. figure(2)
35. plot(SNR,CapT_GWF(3,:,1),'-o', SNR,CapT_GWF(3,:,2),'-
    p',SNR,CapT_GWF(3,:,3),'-*')
36. hold on
37. plot(SNR,CapT(3,:,1),'--*',SNR,CapT(3,:,2),'--<',
    SNR,CapT(3,:,3),'--+')
38. grid on
39. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
40. legend('GWF + Metodo Bloque','GWF + Metodo Alternado','GWF +
    Metodo Aleatorio','Sin GWF + Metodo Bloque','Sin GWF + Metodo
    Alternado','Sin GWF + Metodo Aleatorio','Location','northwest')
```

```

41. title('Capacidad Total del Sistema (64-QAM)')

42. %% 16QAM usuario 1/2/3/4 con y sin GWF
43. figure(3)
44. subplot(1,3,1)
45. plot(SNR,capGWF(2,:,1),'-o',SNR,capGWF(2,:,4),'-
    p',SNR,capGWF(2,:,7),'-+',SNR,capGWF(2,:,10),'-x')
46. hold on
47. plot(SNR,capM(2,:,1),'--*',SNR,capM(2,:,4),'--
    <',SNR,capM(2,:,7),'--h',SNR,capM(2,:,10),'--s')
48. grid on
49. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
50. legend('GWF + Usuario 1','GWF + Usuario 2', 'GWF + Usuario
    3','GWF + Usuario 4','Sin GWF + Usuario 1','Sin GWF + Usuario
    2', 'Sin GWF + Usuario 3','Sin GWF + Usuario
    4','Location','northwest')
51. title('Capacidad por Usuario Metodo Bloque(16-QAM)')

52. subplot(1,3,2)
53. plot(SNR,capGWF(2,:,2),'-o',SNR,capGWF(2,:,5),'-
    p',SNR,capGWF(2,:,8),'-+',SNR,capGWF(2,:,11),'-x')
54. hold on
55. plot(SNR,capM(2,:,2),'--*',SNR,capM(2,:,5),'--
    <',SNR,capM(2,:,8),'--h',SNR,capM(2,:,11),'--s')
56. grid on
57. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
58. legend('GWF + Usuario 1','GWF + Usuario 2', 'GWF + Usuario
    3','GWF + Usuario 4','Sin GWF + Usuario 1','Sin GWF + Usuario
    2', 'Sin GWF + Usuario 3','Sin GWF + Usuario
    4','Location','northwest')
59. title('Capacidad por Usuario Metodo Alternado(16-QAM)')

60. subplot(1,3,3)
61. plot(SNR,capGWF(2,:,3),'-o',SNR,capGWF(2,:,6),'-
    p',SNR,capGWF(2,:,9),'-+',SNR,capGWF(2,:,12),'-x')
62. hold on
63. plot(SNR,capM(2,:,3),'--*',SNR,capM(2,:,6),'--
    <',SNR,capM(2,:,9),'--h',SNR,capM(2,:,12),'--s')
64. grid on
65. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
66. legend('GWF + Usuario 1','GWF + Usuario 2', 'GWF + Usuario
    3','GWF + Usuario 4', 'Sin GWF + Usuario 1','Sin GWF + Usuario
    2', 'Sin GWF + Usuario 3','Sin GWF + Usuario
    4','Location','northwest')
67. title('Capacidad por Usuario Metodo Aleatorio(16-QAM)')

68. %% 64QAM usuario 1/2/3/4 con y sin GWF
69. subplot(1,3,1)
70. plot(SNR,capGWF(3,:,1),'-o',SNR,capGWF(3,:,4),'-
    p',SNR,capGWF(3,:,7),'-+',SNR,capGWF(3,:,10),'-x')
71. hold on
72. plot(SNR,capM(3,:,1),'--*',SNR,capM(3,:,4),'--
    <',SNR,capM(3,:,7),'--h',SNR,capM(3,:,10),'--s')
73. grid on

```

```

74. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
75. legend('GWF + Usuario 1', 'GWF + Usuario 2', 'GWF + Usuario
3', 'GWF + Usuario 4', 'Sin GWF + Usuario 1', 'Sin GWF + Usuario
2', 'Sin GWF + Usuario 3', 'Sin GWF + Usuario 4',
'Location', 'northwest')
76. title('Capacidad por Usuario Metodo Bloque(16-QAM)')

77. subplot(1,3,2)
78. plot(SNR, capGWF(3, :, 2), '-o', SNR, capGWF(3, :, 5), '-
p', SNR, capGWF(3, :, 8), '-+', SNR, capGWF(3, :, 11), '-x')
79. hold on
80. plot(SNR, capM(3, :, 2), '--*', SNR, capM(3, :, 5), '--
<', SNR, capM(3, :, 8), '--h', SNR, capM(3, :, 11), '--s')
81. grid on
82. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
83. legend('GWF + Usuario 1', 'GWF + Usuario 2', 'GWF + Usuario
3', 'GWF + Usuario 4', 'Sin GWF + Usuario 1', 'Sin GWF + Usuario
2', 'Sin GWF + Usuario 3', 'Sin GWF + Usuario
4', 'Location', 'northwest')
84. title('Capacidad por Usuario Metodo Alternado(16-QAM)')

85. subplot(1,3,3)
86. plot(SNR, capGWF(3, :, 3), '-o', SNR, capGWF(3, :, 6), '-
p', SNR, capGWF(3, :, 9), '-+', SNR, capGWF(3, :, 12), '-x')
87. hold on
88. plot(SNR, capM(3, :, 3), '--*', SNR, capM(3, :, 6), '--
<', SNR, capM(3, :, 9), '--h', SNR, capM(3, :, 12), '--s')
89. grid on
90. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
91. legend('GWF + Usuario 1', 'GWF + Usuario 2', 'GWF + Usuario
3', 'GWF + Usuario 4', 'Sin GWF + Usuario 1', 'Sin GWF + Usuario
2', 'Sin GWF + Usuario 3', 'Sin GWF + Usuario
4', 'Location', 'northwest')
92. title('Capacidad por Usuario Metodo Aleatorio(16-QAM)')

93. %% Total con y sin GWF
94. figure(5)
95. subplot(1,3,1)
96. plot(SNR, CapT_GWF(1, :, 1), '-o', SNR, CapT_GWF(1, :, 2), '-
p', SNR, CapT_GWF(1, :, 3), '-*')
97. hold on
98. plot(SNR, CapT(1, :, 1), '--*', SNR, CapT(1, :, 2), '--<',
SNR, CapT(1, :, 3), '--+')
99. grid on
100. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
101. legend('GWF + Metodo Bloque', 'GWF + Metodo Alternado', 'GWF +
Metodo Aleatorio', 'Sin GWF + Metodo Bloque', 'Sin GWF + Metodo
Alternado', 'Sin GWF + Metodo Aleatorio', 'Location', 'northwest')
102. title('Capacidad Total del Sistema (QPSK)')

103. subplot(1,3,2)
104. plot(SNR, CapT_GWF(2, :, 1), '-o', SNR, CapT_GWF(2, :, 2), '-
p', SNR, CapT_GWF(2, :, 3), '-*')
105. hold on

```

```

106. plot(SNR,CapT(2,:,1),'--*',SNR,CapT(2,:,2),'--< ',
        SNR,CapT(2,:,3),'--+')
107. grid on
108. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
109. legend('GWF + Metodo Bloque','GWF + Metodo Alternado','GWF +
        Metodo Aleatorio', 'Sin GWF + Metodo Bloque','Sin GWF + Metodo
        Alternado','Sin GWF + Metodo Aleatorio',
        'Location','northwest')
110. title('Capacidad Total del Sistema (16-QAM)')

111. subplot(1,3,3)
112. plot(SNR,CapT_GWF(3,:,1),'-o', SNR,CapT_GWF(3,:,2),'-
        p',SNR,CapT_GWF(3,:,3),'-*')
113. hold on
114. plot(SNR,CapT(3,:,1),'--*',SNR,CapT(3,:,2),'--< ',
        SNR,CapT(3,:,3),'--+')
115. grid on
116. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
117. legend('GWF + Metodo Bloque','GWF + Metodo Alternado','GWF +
        Metodo Aleatorio','Sin GWF + Metodo Bloque','Sin GWF + Metodo
        Alternado','Sin GWF + Metodo Aleatorio',
        'Location','northwest')
118. title('Capacidad Total del Sistema (64-QAM)')

119. %% Total con y sin GWF 1 metodo, todas las modulaciones
120. figure(6)
121. plot(SNR,CapT_GWF(1,:,3),'->',SNR,CapT_GWF(2,:,3),'-
        *',SNR,CapT_GWF(3,:,3),'-o')
122. hold on
123. plot(SNR,CapT(1,:,3),'--p',SNR,CapT(2,:,3),'--< ',
        SNR,CapT(3,:,3),'--+')
124. grid on
125. xlabel('SNR [dB]'), ylabel('Capacidad [Mbps]');
126. legend('GWF + QPSK','GWF + 16QAM','GWF + 64QAM','Sin GWF +
        QPSK','Sin GWF + 16QAM', 'Sin GWF +
        64QAM','Location','northwest')
127. title('Capacidad Total Asignacion Alternada')
128. end

```

## **ORDEN DE EMPASTADO**