



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**Análisis y simulación de los parámetros de enlaces satelitales mediante
Matlab/Simulink para fines didácticos.**

AUTOR:

Mendoza Sánchez, Xavier Alexander

Trabajo de Titulación previo a la obtención del título de
INGENIERO EN TELECOMUNICACIONES

TUTOR:

ZAMORA CEDEÑO, NÉSTOR ARMANDO

Guayaquil, Ecuador

7 de marzo del 2018



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr.
Mendoza Sánchez, Xavier Alexander como requerimiento para la
obtención del título de **INGENIERO EN TELECOMUNICACIONES**.

TUTOR

Zamora Cedeño, Néstor Armando

DIRECTOR DE CARRERA

Heras Sánchez, Miguel Armando

Guayaquil, a los 7 días del mes de marzo del año 2018



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Mendoza Sánchez, Xavier Alexander**

DECLARÓ QUE:

El trabajo de titulación **“Análisis y simulación de los parámetros de enlaces satelitales mediante Matlab/Simulink para fines didácticos”** previo a la obtención del Título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 7 días del mes de Marzo del año 2018

EL AUTOR

MENDOZA SANCHEZ, XAVIER ALEXANDER



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Mendoza Sánchez, Xavier Alexander**

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: “**Análisis y simulación de los parámetros de enlaces satelitales mediante Matlab/Simulink para fines didácticos**”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

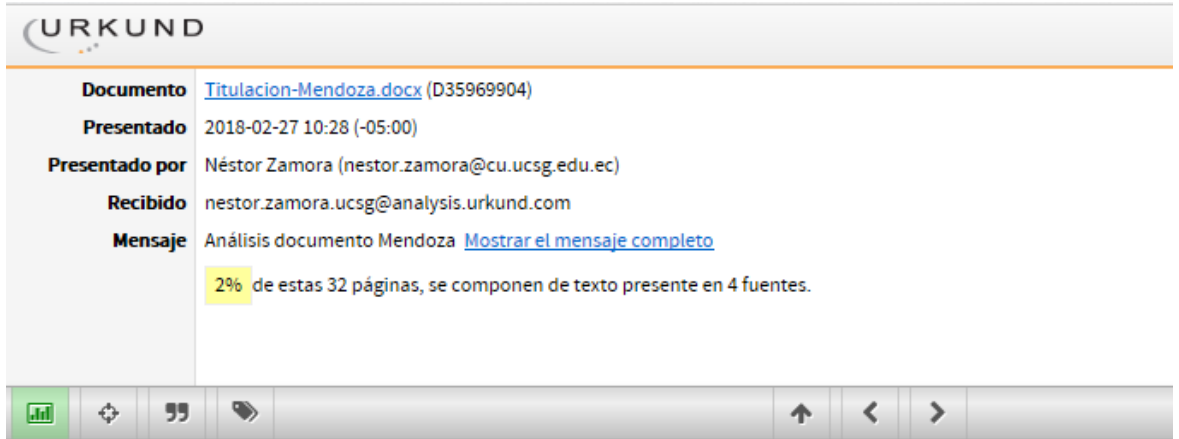
Guayaquil, a los 7 días del mes de Marzo del año 2018

EL AUTOR

MENDOZA SANCHEZ, XAVIER ALEXANDER

REPORTE URKUND

Informe del Trabajo de Titulación de la Carrera de Ingeniería en Telecomunicaciones, con 2% de coincidencias perteneciente al estudiante, **MENDOZA SÁNCHEZ, XAVIER ALEXANDER**.



The screenshot displays the URKUND interface with the following details:

Documento	Titulacion-Mendoza.docx (D35969904)
Presentado	2018-02-27 10:28 (-05:00)
Presentado por	Néstor Zamora (nestor.zamora@cu.ucsg.edu.ec)
Recibido	nestor.zamora.ucsg@analysis.orkund.com
Mensaje	Análisis documento Mendoza Mostrar el mensaje completo 2% de estas 32 páginas, se componen de texto presente en 4 fuentes.

At the bottom of the interface, there is a navigation bar with icons for home, search, quote, and document, along with navigation arrows.

Atte.

M. Sc. Néstor Armando Zamora Cedeño

TUTOR TRABAJO DE TITULACIÓN

Profesor Titular Auxiliar

DEDICATORIA

Dedico este trabajo a mis padres que fueron los pilares fundamentales durante toda mi vida y siempre me han dado su apoyo incondicional en las decisiones que he tomado, a Diego Sánchez quien con sus consejos me ayudo en la elección de mi carrera y durante mi formación profesional, a Mayra y Analith Sánchez fue alguien importante en el desarrollo del proyecto.

EL AUTOR

MENDOZA SANCHEZ, XAVIER ALEXANDER

AGRADECIMIENTO

Agradezco primero que todo a Dios, a mis padres, a mis abuelos y a mis tíos que con sus consejos me ayudaron en mi formación personal y profesional. A mi tutor el Ing. Néstor Zamora que con el aporte de sus conocimientos supo guiarme en el desarrollo de mi proyecto de tesis para conseguir mi título profesional. También agradezco a mis amigos que estuvieron ayudándome en el desarrollo de este proyecto.

EL AUTOR

MENDOZA SANCHEZ, XAVIER ALEXANDER



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

f. _____

ROMERO PAZ, MANUEL DE JESUS
DECANO

f. _____

PALACIOS MELÉDEZ, EDWIN FERNANDO
COORDINADOR DE ÁREA

f. _____

MONTENEGRO, MARCOS
OPONENTE

INDICE GENERAL

Índice de figuras	XI
Índice de Tablas	XIII
Capítulo 1: Introducción	15
1.1. Introducción.....	15
1.2. Antecedentes	16
1.3. Planteamiento del problema	16
1.4. Objetivos del problema de investigación.	16
1.4.1. Objetivo general.....	16
1.4.2. Objetivos específicos.	17
1.5. Hipótesis	17
1.6. Metodología de Investigación	17
Capítulo 2: Fundamentación Teórica	18
2.1. Satélites	18
2.1.1. Definición	18
2.1.2. Arquitectura	18
2.2. Enlace Satelital	19
2.2.1. Definición	19
2.2.2. Parámetros de un enlace satelital	19
2.2.3. Ecuaciones de un enlace satelital	24
2.2.4. Formas de acceso al satélite	29
2.3. MATLAB.....	39
2.3.1. Definición	39
2.3.2. Características de MATLAB.....	39
2.3.3. Uso interactivo de MATLAB	40
2.3.4. Principales comandos.....	41
2.3.5. Introducción de datos en MATLAB.....	42

2.3.6. Expresiones en MATLAB	43
2.3.7. Matrices	45
2.3.8. Uso de archivos -M en una simulación.....	46
Capítulo 3: Simulación y Resultados obtenidos	48
3.1. Creación de interfaces con guides.	48
3.2. Diseño de la aplicación.....	51
3.2.1. Interfaz 1	51
3.2.2. Interfaz 2 (PIRE)	53
3.2.3. Interfaz 3 (Temperatura Equivalente de Ruido).....	55
3.2.4. Interfaz 4 (Densidad de Ruido)	56
3.2.5. Interfaz 5 (Relación de Potencia de Portadora a Densidad de Ruido)	57
3.2.6. Interfaz 6 (Relación de Densidad de Energía de Bit a Densidad de Ruido)	58
3.2.7. Interfaz 7 (Relación de Ganancia a Temperatura de Ruido)	59
3.2.8. Interfaz 8 (Perdidas del Espacio Libre)	61
3.2.9. Interfaz 9 (Ecuación del Enlace de Subida).....	62
3.2.10. Interfaz 10 (Ecuación del Enlace de Bajada)	63
3.2.11. Interfaz 11 (Punto de Operación del Transponder).....	64
3.3. Demostración de la aplicación	65
Capítulo 4: Conclusiones y Recomendaciones	71
4.1. Conclusiones.....	71
4.2. Recomendaciones.....	71
Referencias Bibliográficas	73

Índice de figuras

Capítulo 2

Figura 2. 1: Modelo de Enlace Satelital.....	19
Figura 2. 2: Unidad de Transmisión del canal.	29
Figura 2. 3: División de las frecuencias en FDMA.....	30
Figura 2. 4: Multiplexación por división de tiempo.....	32
Figura 2. 5: Combinación de TDMA y FDMA.	33
Figura 2 6 Entorno de MATLAB.....	41

Capítulo 3

Figura 3. 1: Ingresar a una guide.....	48
Figura 3. 2: Selección de guide.....	49
Figura 3. 3: Ventana de edición de la guide,	49
Figura 3. 4: Paleta de componentes de la guide.	50
Figura 3. 5: Opción de guardar y play de la guide.....	50
Figura 3. 6: Archivo con extensión .m.	51
Figura 3. 7: Interfaz 1.....	51
Figura 3. 8: Interfaz 2.....	53
Figura 3. 9: Explicación de los botones de la interfaz.	54
Figura 3. 10: Interfaz 3.....	55
Figura 3. 11: Interfaz 4.....	56
Figura 3. 12: Interfaz 5.....	57
Figura 3. 13: Interfaz 6.....	58
Figura 3. 14: Interfaz 7.....	59
Figura 3. 15: Interfaz 8.....	61
Figura 3. 16: Interfaz 9.....	62
Figura 3. 17: Interfaz 10.....	63
Figura 3. 18: Interfaz 11.....	64
Figura 3. 19: Demostración de PIRE.....	66
Figura 3. 20: Demostración de la T_e	66
Figura 3. 21: Demostración de Temperatura equivalente de ruido.	66
Figura 3. 22: Demostración 2 de la N_o	67
Figura 3. 23: Demostración de Densidad de ruido.	67

Figura 3. 24: Demostración de Potencia de portadora a densidad de ruido.	68
Figura 3. 25: Demostración de la Relación de densidad de energía de bit a densidad de ruido.	68
Figura 3. 26: Demostración de la relación G/T_e	69
Figura 3. 27: Demostración de la Ecuación del enlace de subida.	69
Figura 3. 28: Demostración de la ecuación del enlace de bajada.	70

Índice de Tablas

Capítulo 2

Tabla 2. 1: Lista de comando más usados en MATLAB.....	41
Tabla 2. 2: Símbolos y constantes de MATLAB.	44
Tabla 2. 3: Operadores más comunes en MATLAB.	45
Tabla 2. 4: Comandos para genera matrices.	45

RESUMEN

El presente trabajo se realizó con intenciones didácticas para reforzar los conocimientos recibidos en el aula de clase sobre el cálculo de los parámetros de un enlace satelital, creando una aplicación mediante MATLAB teniendo como base los conocimientos previos adquiridos durante mi formación como profesional usando la herramienta GUI con la que se creó interfaces que facilitan el cálculo de los parámetros y sirve de ayuda para la comprensión de la teoría aplicada a la práctica. Además de servir como una herramienta de apoyo para los docentes durante la formación de profesionales en la carrera de Ingeniería en Telecomunicaciones y afines a esta. Se debe considerar que en la parte didáctica de los sistemas satelitales resulta favorable utilizar este tipo de programas de manera de ejemplo para complementar el aprendizaje. Durante el desarrollo del proyecto se presentó retos que fueron superados porque se puso en práctica conocimientos adquiridos en el salón de clase.

PALABRAS CLAVES: SATELITAL, MATLAB, INTERFAZ, PARÁMETROS, ENLACES, PROPAGACIÓN

ABSTRACT

The present work was carried out with didactic intentions to reinforce the knowledge received in the classroom on the calculation of the parameters of a satellite link, creating an application through MATLAB based on the previous knowledge acquired during my professional training using the GUI tool with which interfaces were created that facilitate the calculation of the parameters and helps to understand the theory applied to the practice. In addition to serving as a support tool for teachers during the training of professionals in the career of Telecommunications Engineering and related to it. It should be considered that in the didactic part of the satellite systems it is favorable to use this type of programs as an example to complement the learning. During the development of the project, challenges were presented that were overcome because knowledge acquired in the classroom was put into practice.

Keywords: SATELLITE, MATLAB, INTERFACE, PARAMETERS, LINKS, PROPAGATION

Capítulo 1: Introducción

1.1. Introducción

En el trabajo que se va a desarrollar a continuación, presentaré unas breves definiciones de lo que son satélites, su arquitectura, su clasificación (la cual varía mucho según su utilización y la órbita en la que esté ubicado), después de tener claro los conceptos pasaré a explicar rápida y de manera concisa cuales son los parámetros más importantes que debe de cumplir un enlace satelital para que opere sin problemas.

Una vez que se deje claro los conceptos y las ecuaciones que se deben de usar para calcular dichos parámetros, pasaré a simularlos mediante el programa MATLAB, utilizando el programa como material de apoyo, esto se va a realizar con un fin de que el estudiante pueda reforzar lo aprendido en el salón de clase.

El programa MATLAB (“Matrix Laboratory”) es una herramienta que nos permite realizar cálculos matemáticos, tanto con vectores como con matrices, una de las grandes ventajas es que también permite trabajar con cantidades escalares. Tiene muchas opciones al momento de realizar gráficos ya que se puede realizar tanto en 2D como en 3D, ya que es un programa que maneja su propio lenguaje de programación. Lo que para este trabajo contribuirá de gran manera al momento de realizar la simulación.

Como ya lo mencioné los satélites se clasifican de muchas maneras y uno de los satélites más importantes son los satélites meteorológicos, con los cuales comenzó todo el avance tecnológico que nos ha ayudado en muchas ocasiones a evitar grandes desgracias en catástrofes naturales. Después de estos avances los satélites fueron implementados para realizar transmisiones televisivas en los años 50 y 60.

En pocas palabras se puede decir que un enlace satelital está formado por tres fases, de las cuales dos se encuentran ubicadas en las estaciones terrestres, las cuales toman el nombre de enlace de subida o bajada, y la

última está ubicada en el espacio, donde el transpondedor recibirá la señal y la devuelve con una frecuencia más baja. La función de un satélite es recibir una señal, amplificarla, para luego volver a transmitirla.

1.2. Antecedentes

En términos generales, un satélite es un objeto pequeño que gira alrededor de un objeto grande en el espacio. Por ejemplo, la luna que es un satélite natural que gira alrededor de la tierra. Ya se sabe que las comunicaciones hacen referencia al intercambio o compartimento de información entre dos o más entidades, a través de cualquier medio o canal. En otras palabras, esto no es más que enviar, recibir y procesamiento de información.

Si la comunicación tiene lugar entre dos estaciones terrenas a través de un satélite, se llama comunicación satelital. En esta comunicación, las ondas electromagnéticas se utilizan como señales portadoras. Estas señales llevan la información como voz, audio, video o cualquier otro dato entre el suelo y el espacio y viceversa.

La Unión Soviética había lanzado el primer satélite artificial del mundo llamado Sputnik 1 en 1957. Casi 18 años después, India también lanzó el satélite artificial llamado, Aryabhata en 1975.

1.3. Planteamiento del problema

La falta de conocimiento y la poca utilización de simuladores como material de apoyo ha creado la necesidad de incluir una parte práctica para el cálculo de los parámetros de un enlace satelital para entender de mejor manera la teoría.

1.4. Objetivos del problema de investigación.

1.4.1. Objetivo general

Diseñar una aplicación en MATLAB para el cálculo de los parámetros de un enlace satelital con el fin de que los estudiantes puedan comprender

mejor la resolución de este tipo de problemas propuestos en el salón de clase.

1.4.2. Objetivos específicos.

- Describir los parámetros que se utilizan en los enlaces satelitales.
- Describir como el programa MATLAB se utiliza en la simulación de sistemas.
- Calcular los parámetros de los enlaces satelitales utilizando MATLAB.
- Realizar las prácticas de simulación en el programa MATLAB.

1.5. Hipótesis

Con el uso de MATLAB como herramienta de apoyo los estudiantes de la carrera de ingeniería en Telecomunicaciones comprenden de una manera más práctica los problemas propuestos por los catedráticos en clase.

1.6. Metodología de Investigación

En el trabajo de investigación previo a la obtención del título de Ingeniero en Telecomunicaciones, las metodologías que se utilizaron para desarrollarlo fueron descriptivas, empíricas, cuantitativas y experimentales. El método Descriptivo debido a que se estudió y se revisó cada parámetro satelital para poder saber que se va a calcular.

Capítulo 2: Fundamentación Teórica

2.1. Satélites

2.1.1. Definición

Los sistemas satelitales son en pocas palabras un sistema repetidor, el cual es capaz de recibir y volver a enviar una señal por medio de un dispositivo receptor-transmisor al cual se lo conoce como transponder, dicho dispositivo recibe una parte del espectro, la amplifica y luego lo envía a otra frecuencia para no tener inconvenientes con la interferencia de la señal. (Hernandez, Corredor, & Pedraza, 2010)

En lo que respecta a satélites como ya lo mencionaron existen dos tipos de satélites ya sean naturales o artificiales, la tierra cuenta con un solo satélite natural que es conocido por todos, la Luna, para los satélites artificiales son todos los que han sido creados por el hombre y enviados al espacio a orbitar la tierra con diferentes fines.

2.1.2. Arquitectura

Los satélites artificiales están constituidos tanto por componentes comunes hasta otros componentes específicos que varían según la misión, estos pueden ser:

- a) Sistema de suministro de energía:** es el encargado de asegurar el buen funcionamiento de todos los sistemas, comúnmente está constituido por paneles solares. (Jami & Vega, 2012)
- b) Sistema de control:** es el ordenador principal del satélite, es el que procesa las instrucciones que ya fueron almacenadas y a su vez recibe las nuevas instrucciones enviadas desde la tierra. (Jami & Vega, 2012)
- c) Sistema de comunicaciones:** es un conjunto de antenas y transmisores que sirven para poder comunicarse con las estaciones de seguimiento, para enviar la información y recibir instrucciones. (Jami & Vega, 2012)
- d) Sistema de posicionamiento:** sirve para enviar al satélite a una posición determinada y para poderlo guiar hacia un nuevo objetivo. (Jami & Vega, 2012)

- e) **Blindaje térmico:** son los instrumentos que protegen al satélite de los cambios bruscos de temperatura a los que se exponen en el espacio, dependiendo si están de frente al sol o de espaldas a este, este tipo de materiales es el que le suele dar una coloración dorada al satélite. (Jami & Vega, 2012)
- f) **Carga útil:** son todos los instrumentos que fueron adaptados al satélite según para la misión que haya sido construido. (Jami & Vega, 2012)

2.2. Enlace Satelital

2.2.1. Definición

Un enlace satelital básicamente es el canal por el cual van a ser enviada y recibidas las señales, está conformado de tres partes: un enlace de subida, un transpondedor y un enlace de bajada.

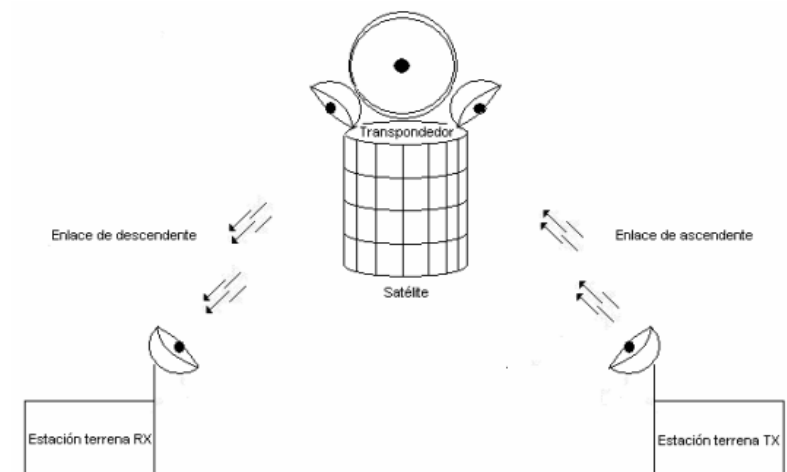


Figura 2. 1: Modelo de Enlace Satelital.

Fuente: (Peña, 2018).

2.2.2. Parámetros de un enlace satelital

Para que un parámetro satelital esté en óptimas condiciones tiene que cumplir con todos los parámetros establecidos, según el enlace con el que se trabaje.

2.2.2.1. Potencia Radiada Isotrópica Efectiva (PIRE)

Según (Hernandez et al., 2010), se la conoce como una potencia de transmisión equivalente, y viene definida por la siguiente ecuación:

$$PIRE = P_r * G_t$$

Ecuación 2. 1

Fuente: (Hernandez et al., 2010)

donde:

P_r = potencia total radiada de una antena [watts],

G_t = ganancia de la antena transmisora [adimensional],

Para describirlo en términos logarítmicos (dB), la formula anterior se expresa de la siguiente manera:

$$PIRE_{dBW} = P_{r(dBW)} + G_{t(dB)}$$

Ecuación 2. 2

Fuente: (Hernandez et al., 2010)

Ahora hablando de la salida del transmisor, P_r esta definida de la siguiente manera:

$$P_r = P_t - L_{bo} - L_{bf}$$

Ecuación 2. 3

Fuente: (Hernandez et al., 2010)

Donde:

P_t = potencia de salida real del transmisor [dBw],

L_{bo} = perdidas de respaldo por HPA [dB],

L_{bf} = ramificación total y perdida del alimentador [dB], por lo que:

$$PIRE = P_t - L_{bo} - L_{bf} + G_t$$

Ecuación 2. 4

Fuente: (Hernandez et al., 2010)

2.2.2.2. Temperatura equivalente de ruido (Te)

Este parámetro es un valor supuesto, pero no puede medirse; Te usualmente se lo utiliza para reemplazar a la figura de ruido, debido a que es una táctica más precisa para revelar el ruido que presenta un dispositivo o un receptor al momento de calcular su rendimiento, la potencia de ruido viene a ser expresada de la siguiente manera en la Ecuación 2.5. (Hernandez et al., 2010)

$$N = KTB$$

Ecuación 2. 5

Fuente: (Hernandez et al., 2010)

Por lo que,

$$T = \frac{N}{KB}$$

Ecuación 2. 6

Fuente: (Hernandez et al., 2010)

Donde:

N= potencia total de ruido [watts],

K= la constante de Boltzmann $\left(1.38 * 10^{-23} \frac{J}{^{\circ}K}\right)$,

B= ancho de banda [Hz],

T= temperatura ambiente [$^{\circ}K$]; para expresar el factor ruido en términos adimensionales se utiliza la siguiente formula:

$$NF = 1 + \frac{T_e}{T}$$

Ecuación 2. 7

Fuente: (Hernandez et al., 2010)

Entonces, despejando T_e se tiene:

$$T_e = T(NF - 1)$$

Ecuación 2. 8

Fuente: (Hernandez et al., 2010)

De esta manera se encuentra T_e , siendo esta la temperatura equivalente de ruido.

2.2.2.3. Densidad de ruido

Según (Hernandez et al., 2010) , la densidad de ruido viene a ser, la potencia de ruido total regularizada a un ancho de banda de 1Hz y viene definida por la siguiente expresión matemática:

$$N_0 = \frac{N}{B} = KT_e$$

Ecuación 2. 9

Fuente: (Hernandez et al., 2010)

Donde N_0 = es la densidad de potencia de ruido [W/Hz]. Para expresar la Ecuación 2.9 en términos logarítmicos se tiene la siguiente expresión:

$$N_{0\frac{dBW}{Hz}} = 10\log K + 10\log T_e$$

Ecuación 2. 10

Fuente: (Tomasi, 2010)

2.2.2.4. Relación de potencia de portadora a densidad de ruido (C/No)

Para hacer este cálculo se tiene que hacer relación a la Ecuación 2.9, en la cual se define como obtener la densidad de ruido (No), por lo que se tiene la siguiente expresión matemática:

$$\frac{C}{N_0} = \frac{C}{KT_E}$$

Ecuación 2. 11

Fuente: (Hernandez et al., 2010)

Donde C = la potencia de la portadora de banda ancha.

Para expresarlo en términos logarítmicos se tiene la siguiente expresión:

$$\frac{C}{N_0} (dB) = C_{dBW} - N_{0(dBW)}$$

Ecuación 2. 12

Fuente: (Tomasi, 2010)

2.2.2.5. Relación de densidad de energía de bit a densidad de ruido (Eb/No)

Según (Hernandez et al., 2010) , este es uno de los parámetros más relevantes y más empleado en las comunicaciones satelitales al momento de valorar un sistema de radio digital, es un método idóneo al momento de examinar los sistemas digitales que emplean distintas tasas de transmisión, diferentes esquemas de modulación.

$$\frac{Eb}{N_0} = \frac{\frac{C}{fb}}{\frac{N}{B}} = \frac{CB}{Nfb}$$

Ecuación 2. 13

Fuente: (Hernandez et al., 2010)

Para tener la Ecuación 2.13 en términos logarítmicos se tiene que hacer lo siguiente:

$$\frac{E_b}{N_0} (dB) = \frac{C}{N} (dB) + \frac{B}{f_b} (dB)$$

Ecuación 2. 14

Fuente: (Tomasi, 2010)

2.2.2.6. Relación de ganancia a temperatura equivalente de ruido (G/Te)

Según (Hernandez et al., 2010), es una forma de valorar, que se utiliza para interpretar la calidad de un satélite es un receptor de una estación terrena, ya que las potencias excesivamente pequeñas de la portadora de recepción que usualmente se perciben en los sistemas satelitales, repetidamente un LNA está ubicado en punto de alimentación de la antena; este parámetro es una explicación de la ganancia de la antena receptora más la ganancia del LNA, a la temperatura de ruido equivalente, teniendo así la siguiente expresión matemática:

$$\frac{G}{T_e} = \frac{G_r + G_{LNA}}{T_e}$$

Ecuación 2. 15

Fuente: (Hernandez et al., 2010)

$\frac{G}{T_e}$ es un parámetro muy beneficioso para describir las relaciones $\frac{E_b}{N_0}$ y $\frac{C}{N}$ en el transponder del satélite y receptores de la estación terrena. (Hernandez et al., 2010). Para expresarla en términos logarítmicos se tiene la siguiente expresión:

$$\frac{G}{T_e} (dBK^{-1}) = A_{r(dB)} + A_{(LNA)(dB)} - T_{e(dBK)}$$

Ecuación 2. 16

Fuente: (Tomasi, 2010)

2.2.2.7. Perdidas por espacio libre (PEL o Lo)

Según (Hernandez et al., 2010), el espacio libre es un entorno uniforme exento de corrientes y cargas eléctricas, por lo que, está libre de pérdidas por el efecto de Joule, en la que las ondas de radio se mueven en línea recta y sin debilitarse.

$$PEL = L_0 = \left(\frac{4\pi D}{\lambda}\right)^2$$

Ecuación 2. 17

Fuente: (Hernandez et al., 2010)

2.2.3. Ecuaciones de un enlace satelital

En las siguientes ecuaciones se detallan los modelos de subida y de bajada de un satélite, en estos cálculos solo se toman en cuenta las ganancias y pérdidas ideales, así como los efectos del ruido térmico. (Hernandez, Corredor C., & Pedraza, 2010)

2.2.3.1. Ecuación del enlace de subida

$$\frac{C}{N_0} = \frac{G_t P_r (L_0 L_u) G_r}{K T_e} = \frac{G_t P_r (L_0 L_u)}{K} * \frac{G}{T_e}$$

Ecuación 2. 18

Fuente: (Hernandez et al., 2010)

L_0 son las perdidas del espacio libre,

L_u son las perdidas atmosféricas de subida adicionales.

Las señales de subida y bajada tienen que atravesar por la atmosfera de la tierra, donde son absorbidas de a poco por la humedad, oxígeno y partículas en el aire. Reconociendo el ángulo de elevación, la distancia de viaje de la señal RF por la atmosfera cambia de una estación terrena a otra, debido a que L_p y L_u simbolizan perdidas, estos son valores decimales menores a 1. G/T_e es la ganancia de la antena receptora del transponder sumado a la ganancia del LNA dividida para la temperatura de ruido equivalente de entrada. (Hernandez et al., 2010)

Pasándolo a una expresión logarítmica, se tiene la siguiente ecuación:

$$\frac{C}{N_0} = 10 \log(G_t P_r) - 20 \log\left(\frac{4\pi D}{\lambda}\right) + 10 \log\left(\frac{G}{T_e}\right) - 10 \log(L_u) - 10 \log(k)$$

Ecuación 2. 19

Fuente: (Hernandez et al., 2010)

Donde que:

$10 \log(G_t P_r)$ viene a ser la PIRE de la estación terrena, (Hernandez et al., 2010)

$20 \log\left(\frac{4\pi D}{\lambda}\right)$ es la representación de las pérdidas por el espacio libre, (Hernandez et al., 2010)

$10 \log\left(\frac{G}{T_e}\right)$ es una relación entre la ganancia a temperatura equivalente de ruido del satélite, (Hernandez et al., 2010)

$10 \log(L_u)$ esto representa a las pérdidas atmosféricas adicionales, (Hernandez et al., 2010)

$10 \log(k)$ esto representa la constante de Boltzman. (Hernandez et al., 2010)

Lo que nos da la siguiente expresión:

$$\frac{C}{N_0} = PIRE(dBW) - L_p(dB) + \frac{G}{T_e}(dBK^{-1}) - L_u(dB) - K(dBWK)$$

Ecuación 2. 20

Fuente: (Tomasi, 2010)

2.2.3.2. Ecuaciones del enlace de bajada

$$\frac{C}{N_0} = \frac{G_t P_r (L_o L_d) G_r}{K T_e} = \frac{G_t P_r (L_o L_d)}{K} * \frac{G}{T_e}$$

Ecuación 2. 21

Fuente: (Hernandez et al., 2010)

Donde se tiene que:

L_d , representan las pérdidas atmosféricas de bajada adicionales, a lo que se pasa la Ecuación 2.16 a términos logarítmicos, se tiene lo siguiente:

$$\frac{C}{N_0} = 10 \log(G_t P_r) - 2 \log\left(\frac{4\pi D}{\lambda}\right) + 10 \log\left(\frac{G}{T_e}\right) - 10 \log(L_d) - 10 \log(k)$$

Ecuación 2. 22

Fuente: (Hernandez et al., 2010)

Donde:

$10 \log(G_t P_r)$, representa la PIRE del satélite, (Hernandez et al., 2010)

$10 \log(L_d)$, representan las pérdidas atmosféricas adicionales. (Hernandez et al., 2010)

A lo que se tiene la siguiente expresión:

$$\frac{C}{N_0} = PIRE(dBW) - L_p(dB) + \frac{G}{T_e} (dBK^{-1}) - L_a(dB) - K(dBWK)$$

Ecuación 2. 23

Fuente: (Tomasi, 2010)

2.2.3.3. Diferentes apreciaciones importantes que hay que tener en cuenta en el cálculo de enlaces satelitales.

Se debe de tener muchos aspectos en consideración a la hora de calcular los parámetros de un enlace satelital, y como mencionaron anteriormente cuales son los cálculos principales que hay que tener en cuenta, no se puede dejar a un lado las siguientes apreciaciones importantes a la hora de calcular dichos parámetros:

$$W = \frac{P_t}{4\pi d^2} \left[\frac{\text{watts}}{m^2} \right]$$

Ecuación 2. 24

Fuente: (Hernandez et al., 2010)

Si la antena representa una ganancia, entonces se tiene que:

$$W = \frac{P_t G_t}{4\pi d^2} \left[\frac{\text{watts}}{m^2} \right]$$

Ecuación 2. 25

Fuente: (Hernandez et al., 2010)

Como ya se sabe que el PIRE= $P_t G_t$, entonces se tiene lo siguiente:

$$W_{\frac{dBW}{m}} = PIRE_{dBW} - 20 \log d_{km} - 71$$

Ecuación 2. 26

Fuente: (Hernandez et al., 2010)

Según (Hernandez et al., 2010) investigo que: *“una antena receptora “recoge” la señal y la cantidad de la señal “recogida” depende del tamaño (área efectiva) de la antena”*.

Por lo que se tiene que la potencia que recibe la antena estará dada por la siguiente ecuación:

$$P_r = W * A_e [\text{watts}]$$

Ecuación 2. 27

Fuente: (Hernandez et al., 2010)

Donde se tiene que,

$$A_e = (\lambda^2/4\pi) G_r$$

Ecuación 2. 28

Fuente: (Hernandez et al., 2010)

Por lo que se tiene que la Ecuación 2.27 queda de la siguiente manera:

$$P_{r_{dBW}} = PIRE_{dBW} - L_{o_{dB}} - G_{rdB}$$

Ecuación 2. 29

Fuente: (Hernandez et al., 2010)

Según (Hernandez et al., 2010), si se tiene en cuenta que la Ecuación 2.29 se considera que G_r es la ganancia de una antena de $1m^2$ con una eficiencia del 100%, P_r vendría a ser el nivel de iluminación por unidad de superficie, expresada en (dBW/m^2) y por lo que se puede decir que el nivel de iluminación en la Ecuación 2.21 se podrá expresar de la siguiente manera:

$$W_{dBW/m^2} = PIRE_{dBW} - L_{o_{dB}} - G_{1m^2dB}$$

Ecuación 2. 30

Fuente: (Hernandez et al., 2010)

2.2.3.3.1. Punto de operación del transponder

Debido a que el amplificador de potencia de salida de este dispositivo no es lineal está obligado a funcionar por debajo del punto de saturación para poder evitar las distorsiones lineales. Para cumplir lo que mencionaron anteriormente se debe de limitar la potencia de entrada, por lo que un TWTA (amplificador en base de tubos) típico genera un gran despilfarro de potencia; para hacer que este despilfarro sea mínimo existen dos maneras de hacerlo, una de estas se basa en emplear un SSPA (amplificador de estado sólido) y se lo emplearía como amplificador de la potencia de salida; el otro método consiste en el empleo de los TWTA con un linealizador; los dos métodos son útiles y ayudan a enriquecer las características de intermodulación de los transpondedores. (Hernandez et al., 2010)

Según (Hernandez et al., 2010), la disminución de la potencia de entrada (IBO) viene determinada como la relación entre la densidad de flujo de saturación y la densidad de flujo de operación de una portadora y la

disminución de la potencia de salida (OBO), y tiene la siguiente expresión matemática:

$$OBO_{dB} = IBO_{dB} - X_{dB}$$

Ecuación 2. 31

Fuente: (Hernandez et al., 2010)

Donde se tiene que X está determinada por la relación de la ganancia de compresión entre la reducción de potencia de entrada y salida, este valor varia en la ocasión de que exista una o varias portadoras. (Hernandez, Corredor C., & Pedraza, 2010)

Con lo mencionado anteriormente se puede pasar a determinar el PIRE de funcionamiento del transponder:

$$PIRE_{updB} = PIRE_{satdB} - OBO_{dB}$$

Ecuación 2. 32

Fuente: (Hernandez et al., 2010)

2.2.3.3.2. Densidad de flujo de potencia del satélite en la superficie terrestre

Según lo investigado por (Hernandez et al., 2010) dice que: *“las posibilidades de interferencia provenientes del satélite transmisor se limitan reduciendo en la superficie terrestre la densidad de flujo de potencia máxima producida por un satélite, dichos limites varían en función del ángulo de llegada”*.

Para hacer un cálculo de cada caso se puede tomar como referencia la siguiente ecuación:

$$PAD_{4KHZ} = W - 10 \log \left(\frac{B}{4KHZ} \right)$$

Ecuación 2. 33

Fuente: (Hernandez et al., 2010)

en donde se tiene que W es un nivel de iluminación el cual se obtuvo en la Ecuación 2. 20 y calculado para el enlace descendente; B es el ancho de banda por el cual pasa la portadora e IDR es la tasa de bits de datos de intermedia. (Hernandez et al., 2010)

2.2.3.3.3. Unidad de canal QPSK/IDR

Según lo que investigaron (Hernandez et al., 2010): “*a: velocidad de información (IR); b, c: velocidad compuesta (CR=IR+OH), siendo OH la cantidad de bits de información del encabezamiento; d: velocidad de transmisión (R=CR/FEC); e: la velocidad de símbolos (SR=R/2) y B= ancho de banda ocupado (B=0.6R)*”.



Figura 2. 2: Unidad de Transmisión del canal.

Fuente: (Hernandez et al., 2010)

2.2.4. Formas de acceso al satélite

2.2.4.1. Acceso múltiple

La manera de definir el acceso múltiple consiste en involucrar a varios sistemas que generan posibilidades en las que múltiples estaciones terrenas acoplen sus enlaces de comunicación por medio de un simple transponder. (Vega, Maza, & López, 2007)

2.2.4.2. Clasificación según la asignación del canal

Según el canal que se le ha asignado a un sistema satelital, ya sea de una manera fija o por medio de un sistema de control, el cual consiste en ir asignando dinámicamente según los requerimientos que se vayan presentando (a esto también se le conoce como asignación por demanda). (Vega et al., 2007)

Por lo que se tiene la siguiente clasificación:

- a) **PAMA (Permanently Assigned Multiple Access):** para cada uno de los canales de cada estación está asignado permanentemente una porción de la capacidad en frecuencia del transponder tanto para FDMA o TDMA. (Vega et al., 2007)

- b) **DAMA (Demand Assigned Multiple Access):** ninguna estación tiene asignado de manera permanente una frecuencia para FDMA o una ráfaga para TDMA, todo se va asignando según la demanda. (Vega et al., 2007)
- c) **RMA (Random Multiple Access):** este tipo de asignación consiste en que en el momento que una estación necesite comunicarse utilizara un intervalo de tiempo de transmisión cualquiera en una portadora. (Vega et al., 2007)

2.2.4.3. Técnicas de multiplexación

2.2.4.3.1. Acceso múltiple por división de frecuencia (FDMA-Frequency Division Multiple Access)

Este tipo de fundamentos es aplicado en la telefonía de larga distancia en FDMA, a la cual se le ofrece a cada usuario una frecuencia de transmisión de tal manera que no se van a superponer, la demodulación de esta se aplica mediante filtros pasa banda. (Amado, 2008)

En FDMA, el volumen del espectro libre o el ancho de banda es dividido en un conjunto de canalizaciones de banda estrecha también llamadas como radiocanales o portadoras, cada una de estas con una frecuencia central y apartado una de las otras, las cuales van a ser asignadas una para cada usuario, este tipo de técnica es usada de manera especial en modulaciones analógicas, sobre todo en FM. (Amado, 2008)

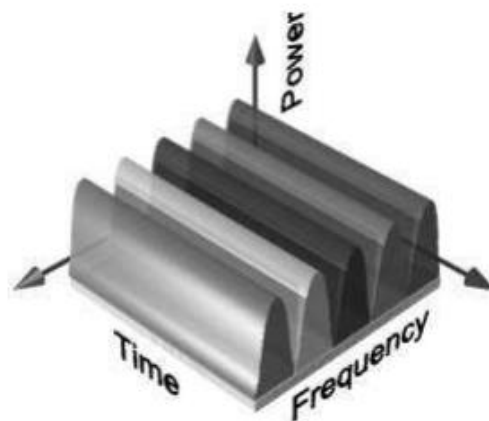


Figura 2. 3: División de las frecuencias en FDMA.
Fuente: (Amado, 2008)

El enlace ascendente se da cuando un el usuario pone su información en un canal determinado, el cual fue fijado por el sistema por medio de la selección en su transmisor de la frecuencia correcta; a diferencia del enlace descendente, la red es la que se ocupa de distribuir una serie de canales para que los receptores para que se armonicen por medio de un filtro pasa banda la frecuencia de interés. (Amado, 2008).

Las características primordiales de FDMA son:

- a) Los usuarios tienen todo el tiempo para entablar su comunicación, (Amado, 2008)
- b) Es usado tanto en modulaciones digitales como analógicas, (Amado, 2008)
- c) En su estado más natural muestra inconvenientes a las modulaciones analógicas debido a que no les deja acoplar aplicaciones de distinta naturaleza, (Amado, 2008)
- d) Tiene restricciones al momento de incluir las señales y la información para mejorar la llamada, (Amado, 2008)
- e) Tiene una gran trayectoria que avala su tecnología, (Amado, 2008)

En la actualidad las técnicas de FDMA han quedado prácticamente en el olvido, lo único que la mantiene en uso en ciertas partes, es que combina con la técnica TDMA, en sistemas como TETRA, DECT y GSM. (Amado, 2008)

2.2.4.3.2. Acceso múltiple por división de tiempo (TDMA - Time Division Multiple Access).

En este tipo de técnica, se fragmenta un periodo en intervalos establecidos a cada cadena digital entrante de manera repetitiva, para realizar la demultiplexación de esta técnica se debe intercambiar la señal con la tasa adecuada, para poder realizar esto, se le da acceso al usuario 1 por un corto periodo de tiempo, después de esto el acceso es concedido al usuario 2 lo cual se repite hasta volver al primer usuario. (Amado, 2008)

Usuario 1	Usuario 2	Usuario 3	Usuario 1	Usuario 2	Usuario 3	Usuario 1
-----------	-----------	-----------	-----------	-----------	-----------	-----------

Figura 2. 4: Multiplexación por división de tiempo.

Fuente: (Amado, 2008)

El ancho de banda que se le da a un usuario de la tecnología GPS una vez que usa la multiplexación TDMA es de aproximadamente de 200kHz, la cantidad de usuarios que pueden estar en un mismo canal es de 8, esto nos da a entender que cada 4.6 ms un usuario podrá transmitir por el canal durante un periodo de 577 us. (Amado, 2008)

Al momento de que el usuario este usando el canal va a pensar que siempre tiene disponibilidad del mismo, pero la realidad es que se hace una alternación a una velocidad increíblemente rápida y que para el usuario es realmente imperceptible; aquí nace la necesidad de una memoria para que los sistemas sean capaces de guardar las comunicaciones entre los usuarios hasta que sean transmitidos, se necesitara un dispositivo de direccionamiento el cual ayude a diferenciar los usuarios y también será necesario un sistema de sincronización para que este indique el momento se puede acceder al canal. (Amado, 2008)

Este tipo de transmisiones se realizan en espacios de duración T_1 , para los que vienen en sucesiones de N intervalos designados en el terminal, por lo que cada intervalo dura $T = T_1/N$, esta duración de tiempo es el tiempo en que el terminal tiene permitido el acceso al medio para poder realizar la transmisión de la información que tenía almacenada para el envío, como un tren de bits denominado ráfaga o burst. (Amado, 2008)

Al momento de hablar de TDMA, se debe conocer algunos conceptos básicos para comprender de que se trata:

- a) **Trama:** según (Amado, 2008) dijo que: “es un ciclo periódico de acceso de los N usuarios a los canales disponibles para compartir. Su duración en intervalos de valor T , tiende a ser menor que $N*T$ ”.
- b) **Periodo de trama:** es la duración de una trama. (Amado, 2008)

- c) **Intervalo de tiempo o time slot:** es el tiempo que tiene de duración una ventana. (Amado, 2008)
- d) **Ráfaga o burst:** esto hace referencia a la secuencia de bits que se van a transmitir o a recibir en un intervalo de tiempo determinado. (Amado, 2008)
- e) **Velocidades del sistema:** existen dos tipos de velocidades, una es la velocidad media de escritura, es el número de bits por usuario en la trama, y el otro tipo de velocidad es la velocidad instantánea de escritura, que es con la que se transmite la información real. (Amado, 2008)

El mayor problema que presenta TDMA es la ampliación de las redes, ya que se tiene que T_1 es un valor fijo, si en un momento determinado el número de usuarios incrementa desmesuradamente, el tiempo que esta designado para la transmisión sería muy corto y la velocidad instantánea para realizarla sería demasiado alta. (Borrell, 2007)

Una vez que se presentó esta gran desventaja de TDMA, es que surgió la idea de mezclarla FDMA, de esta manera es que se pueden incorporar estructuras iguales en otro tipo de frecuencias (FDM). (Amado, 2008)

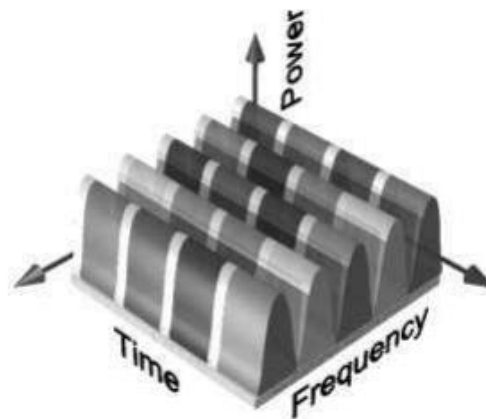


Figura 2. 5: Combinación de TDMA y FDMA.
Fuente: (Amado, 2008).

Esta técnica tiene algunas singularidades, como las siguientes:

- a) Debe de existir una sincronización más rigurosa, todo esto con fin de evitar choques entre ráfagas. (Amado, 2008)

- b) Deben de crearse mecanismos para dar el acceso a las estaciones para emitir y recibir sus intervalos temporales de manera independiente a la ubicación con respecto a los transmisores/receptores. (Amado, 2008)
- c) Toda la información debe de encontrarse de manera digital. (Amado, 2008)

Teóricamente, en TDMA y FDMA los usuarios usan canales que no se interfieren entre sí, las no idealidades son guiadas por medio de tiempos que guarda (TDMA) y las bandas espectrales guarda (FDMA) con la intención de no tener interferencias con otro canal. (Amado, 2008)

2.2.4.3.3. Acceso múltiple por división de código (CDMA-Code Division Multiple Access)

El CDMA tiene como base el principio de ortogonalidad de las señales, lo cual hace que exista una libre recepción sin que haya una interferencia entre los canales. Todo esto es posible haciendo que las señales se superpongan tanto el dominio del tiempo como de la frecuencia. (Amado, 2008)

Según (Amado, 2008), dijo que: *“se dice que dos formas de onda “x” y “y” son ortogonales si su correlación cruzada $R_{xy}(0)$ a lo largo de T es cero, esto es:”*

$$R_{xy}(0) = \int_0^T x(t) \cdot y(t) dt$$

Ecuación 2. 34

Fuente: (Amado, 2008)

Y para el caso de señales discretas, se tiene que:

$$R_{xy}(0) = x^T * y = \sum_{i=1}^I x_i * y_i$$

Ecuación 2. 35

Fuente: (Amado, 2008)

En donde:

$$x^T = [x_1 \ x_2 \ \dots \ x_I]$$

Ecuación 2. 36

Fuente: (Amado, 2008)

$$y^T = [y_1 \ y_2 \ \dots \ y_I]$$

Ecuación 2. 37

Fuente: (Amado, 2008)

A parte de la ortogonalidad deben de cumplir otros requisitos también: las señales a transmitir deben de ir como vectores y sus entradas solo pueden tomar dos valores, los cuales son 1 y -1, esto es conocido como modulación antidopal. La cantidad de 1s y -1s deben de ser iguales, y de la misma manera otro requisito es que el producto punto de cada vector por sí mismo escalado al orden del código, debe ser igual a 1, esto quiere decir que una vez obtenido el producto punto del vector transpuesto por sí mismo se lo divide por el orden del vector. (Jami & Vega, 2012)

$$R_{xx}(0) = x^T * x = \sum_{i=1}^I x_i * x_i$$

Ecuación 2 38

Fuente: (Jami & Vega, 2012)

Lo que se tiene como idea principal es sumar las señales, lo que provoca que se genere un ruido aditivo, el cual puede ser fácilmente filtrado para poder obtener las señales originales, y gracias a la ortogonalidad el bit error rate va a ser el mismo al que se tuviera si se transmitiera por canales distintos. (Rodriguez, 2008)

Según lo investigado por (Amado, 2008), si se presenta el caso de que las dos señales no fueran mutuamente ortogonales, ya sea por un retraso o por cualquier otra adversidad esto haría que se presente un fenómeno denominado crosstalk, lo que nos obliga a poner una condición:

$$\int_0^{T-\tau} x(t) * y(t + \tau) dt = 0$$

Ecuación 2 39

Fuente: (Amado, 2008)

$$\int_{T-\tau}^T x(t) * y(t + \tau - T) dt = 0$$

Ecuación 2 40

Fuente: (Amado, 2008)

Las dos ecuaciones tienen que cumplirse o por lo menos acercarse lo más que se pueda a 0 para cualquier tipo de valor de τ en el sistema. (Rodríguez, 2008)

Teniendo en cuenta el principio de ortogonalidad, en CDMA también hace uso de otras técnicas, una de estas es la técnica del espectro expandido, que se rige bajo el estándar IS-95. A cada usuario se le asignara un código digital único y así mismo se le brindara la total libertad de transmitir en todo el espectro asignado durante el tiempo disponible. (Rodríguez, 2008)

2.2.4.3.4. Espectro expandido

El espectro expandido es una técnica que ha existido por mucho tiempo, como la mayoría de técnica comenzaron con fines militares y la creciente demanda de los usuarios de comunicaciones móviles ha hecho que sea utilizada con otros fines. Este consiste en una manera de transmitir una señal que ocupa un ancho de banda excesivo para enviar la información, la prolongación de la banda es llevada a cabo mediante un código que independiente de los datos. Para la recepción sincronizada con el código del receptor, este se usa para encoger la banda y después poder recuperar la información. (Vega et al., 2007)

Según (Amado, 2008) dijo que: *“en este punto es esencial preguntarse ¿Cómo lograr obtener diferentes señales que sean mutuamente ortogonales con un periodo T y un ancho de banda B? Verdu propone que a menos que el producto TB sea pequeño, la respuesta es 2TB.”*

Según (Amado, 2008), en evidencia a lo que antes se mencionó cuando se tiene un sistema CDMA que cumpla con todas las condiciones

para ser ortogonal de k usuarios en el cual se emplee una modulación antidopar a una tasa de R bits por segundo se necesitara que el ancho de banda se lo más próximo posible a:

$$B = \frac{1}{2} * R * K$$

Ecuación 2. 41

Fuente: (Amado, 2008)

De manera que se pueda obtener un ancho de banda mínimo, el producto de la duración y el ancho de banda de cada forma de onda tendrá cumplirse que sea igual a K/2 para un sistema CDMA, este tipo de condiciones que cumplen traen consigo beneficios, entre esos se tiene que una fuerte resistencia a las distorsiones que se presentan en canales desconocidos y también nos ofrece una capacidad antijamming, lo que esto significa es que ayuda a evitar interferencias. A este tipo de señales es que se les conoce como señales de espectro expandido, las cuales también pueden ser empleadas en sistemas que contengan TDMA y FDMA a un coste de una baja eficiencia espectral. (Amado, 2008)

Hay que tener en cuenta que la ortogonalidad no es un campo obligatorio que deben de cumplir este tipo de sistemas, siempre y cuando se mantengan los niveles de co-interferencia lo suficientemente bajos como que los errores que se introduzcan puedan ser fácilmente eliminados, gracias a esto se pudo lograr que los usuarios de CDMA sean asíncronos ya que sus divisiones de tiempo no deben de estar en fase; por lo que la cantidad de usuarios no se va a ver limitada al doble del producto de la duración y el ancho de banda de la forma de la onda; finalmente el ancho de banda que esté disponible para cada usuario dependerá de la cantidad de usuarios simultáneos y ya no dependerá de la cantidad de usuarios potenciales, por lo que comúnmente se le dice que es una canal inherentemente dinámico. (Amado, 2008)

En este tipo de tecnología como lo es la CDMA el número de usuarios está limitado por varios motivos, entre los cuales se tiene lo siguiente: la

razón del ruido y la señal, la correlación cruzada de las formas de onda, las redundancias de datos y el tipo de receptor. El ancho de banda dividido para la tasa de datos que serán transmitidas va a delimitar la dimensión del espacio en el cual la señal va a trasladarse. (Amado, 2008)

Se podría hacer un cálculo estimado de la máxima cantidad de usuarios que un sistema podría llegar a soportar de la siguiente manera:

Se debe de calcular la interferencia en el demodulador de un receptor genérico, p_r es la potencia que recibe por un sistema de N equipos. (Amado, 2008)

$$i = (N - 1) * p_r$$

Ecuación 2. 42

Fuente: (Amado, 2008)

La densidad de interferencia va a estar definida de la siguiente manera:

$$i_0 = \frac{i}{B} = \frac{(N - 1) * p_r}{B}$$

Ecuación 2. 43

Fuente: (Amado, 2008)

Según (Rodriguez, 2008), después de esto, se calcula la relación R, la cual va a determinar la velocidad binaria neta de los celulares, su unidad está en bits por segundo y se calcula por medio de la siguiente ecuación:

$$R = \frac{e_b}{i_0} = \frac{\frac{p_r}{R}}{(N - 1) * \frac{p_r}{B}} = \frac{B}{R * (N - 1)}$$

Ecuación 2. 44

Fuente: (Rodriguez, 2008)

Con esto se puede hallar el valor de N en función de la ganancia procesada (B/R):

$$N \approx \frac{B}{R * \left(\frac{e_b}{i_0}\right)}$$

Ecuación 2, 45

Fuente: (Rodriguez, 2008)

Este último valor puede llegar a tener valores muy elevados lo que provocara que la cantidad de usuarios de una red CDMA sea alta, a diferencia que en TDMA y FDMA, por lo que fue escogida como técnica de acceso múltiple para las redes de tercera generación. (Rodriguez, 2008)

2.3. MATLAB

2.3.1. Definición

MATLAB es un programa el cual maneja un lenguaje de un alto desempeño que este hecho para ejecutar cálculos técnicos. Este programa incluye varias funciones como las siguientes: calculo, visualización y programación en un área de trabajo muy cómodo en el que se desarrollan los problemas para mostrar las soluciones en una notación matemática. (Borrell, 2007)

Al introducir valores en MATLAB se lo hace una manera interactiva por medio de instrucciones, que para el programa los denomina comando, de la misma manera se puede introducir funciones y programas por un lenguaje organizado y sistematizado. Los elementos fundamentales con los que trabaja MATLAB son las matrices. MATLAB designa la memoria para cada variable de manera rápida y eficiente, por lo que es forzoso que se declaren las variables previo su uso. (Borrell, 2007)

2.3.2. Características de MATLAB

Siendo un programa con tanas aplicaciones, se va a mencionar las características más importantes que posee:

- a)** Realiza cálculos matemáticos rápidamente y con gran exactitud. (Borrell, 2007)
- b)** Capaz de entender muy bien los símbolos matemáticos.(Borrell, 2007)
- c)** Posee varias funciones para gráficos y visualizaciones avanzadas. (Borrell, 2007)
- d)** La programación se lleva a cabo a través de un lenguaje de alto nivel. (Borrell, 2007)

- e) Base para una programación organizada y dirigida a objetos. (Borrell, 2007)
- f) Posee una extensa biblioteca de funciones. (Borrell, 2007)
- g) Paquetes que se especializan en diferentes ramas de ciencia e ingeniería. (Borrell, 2007)

2.3.3. Uso interactivo de MATLAB

Según lo investigado por (Giner, 2008), el ambiente que posee MATLAB está distribuido en ventanas. Las ventanas donde se encuentran la mayoría de comando son las siguientes:

- a) **Command Window:** esta es la primera ventana que se abre al ejecutar MATLAB. Es la ventana principal, en la cual el usuario introduce los comandos para interactuar con MATLAB. (Giner, 2008)
- b) **Command History:** esta ventana es la que guarda el registro de todos los comandos que han sido ingresados. (Giner, 2008)
- c) **Workspace:** conocida también como el área de trabajo, por su traducción al español, en esta ventana está la descripción de cada variable que ha sido usada en cada sesión. (Giner, 2008)

Para las personas que son nuevas en MATLAB es recomendado que utilicen solo la ventana de comandos y, a medida que se familiaricen más con MATLAB puedan manejar las 3 ventanas al mismo tiempo. En la figura 2.12 se muestra el entorno completo de MATLAB.

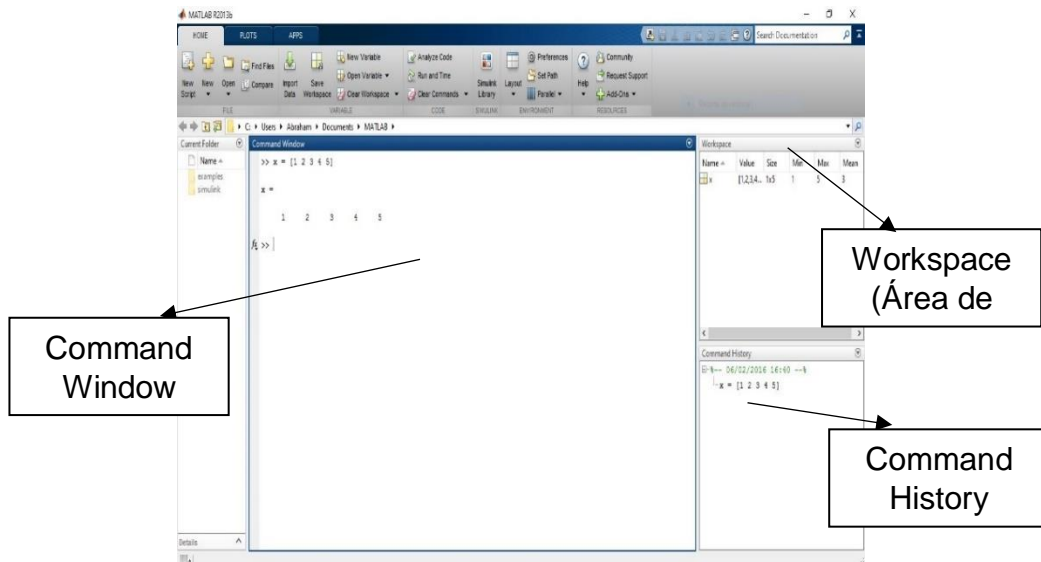


Figura 2.6 Entorno de MATLAB
Elaborado por: Autor

2.3.4. Principales comandos

En la Tabla 2.1 se muestra una lista en que están los comandos más comunes de MATLAB. Dentro de estos comandos se puede diferenciar comandos de búsqueda, ayuda en línea, desplegar las variables introducidas en el espacio de trabajo, entre otros. (Giner, 2008)

Tabla 2. 1: Lista de comando más usados en MATLAB.

Comando	Función
help	Ayuda en línea. Despliega líneas de texto en la ventana de comando conteniendo la descripción sobre un comando específico
helpwin	Despliega una ventana con la descripción específica de un comando y permite ver información sobre otros temas relacionados
lookfor	Busca en la ayuda de todos los comandos la clave especificada
helpdesk	Realiza una búsqueda en hipertexto en un buscador Web proporcionando un acceso directo a toda la documentación: PDFs, información sobre la solución de problemas, etc
doc	Despliega en un buscador Web la página de referencia para el comando especificado, proporciona una descripción, referencias adicionales y ejemplos del comando especificado
figure	Crea una nueva gráfica
close	Cierra una gráfica
who	Despliega las variables presentes en el espacio de trabajo
whos	Despliega las variables presentes en el espacio de trabajo en extenso.
which	Indica la ruta en donde se encuentra la función especificada
cd	Cambia la ruta al subdirectorio superior
pwd	Despliega la ruta en donde se encuentra el directorio de trabajo actual

Fuente: (Giner, 2008).

2.3.5. Introducción de datos en MATLAB

Para poder ingresar datos en MATLAB, se debe aprender sobre las matrices. Para MATLAB, una matriz consiste en un arreglo rectangular de números. Para diferenciar las matrices de 1x1 son denominadas como escalares, las matrices de una columna son conocidas como vectores. Las matrices pueden estar compuestas tanto por valores numéricos como por valores no numéricos. (Borrell, 2007)

Según (Borrell, 2007), cuando se desea asignar un escalar a una variable, se ingresa los valores de la siguiente manera:

Ejemplo 1: dar a la variable dato el valor de 1037

```
>> dato = 1037
```

A lo que MATLAB nos responde de la siguiente manera:

```
dato=
```

```
    1037
```

Según (Borrell, 2007), si se necesita introducir una matriz o un vector, se debe de hacerlo de la siguiente manera:

- a) Se separan los elementos de una columna por un espacio en blanco. (Borrell, 2007)
- b) Se usa un punto y coma (;) para señalar el fin de una columna o vector. (Borrell, 2007)
- c) Se deben de encerrar entre paréntesis rectangulares [] todos los elementos que vaya a contener el vector o la matriz. (Borrell, 2007)

Ejemplos 2: introducir el siguiente vector

```
x = {0.365 -2.564 9.463 -3.574}
```

en la línea de comando se debe ingresar de la siguiente manera:

```
>> x = [0.365 -2.564 9.463 -3.574]
```

y MATLAB desplegara lo siguiente:

```
0.365 -2.564 9.463 -3.574
```

Ejemplo 3: introducir la siguiente matriz

$$X = \begin{bmatrix} 22 & 17 \\ 9 & 28 \end{bmatrix}$$

en la línea de comando se debe ingresar de la siguiente manera:

```
>>X= [22 17; 9 28]
```

y MATLAB responde:

X=

```
22 17  
9 28
```

Cuando se ingrese una matriz o un vector, este se quedará en el área de trabajo (Workspace), y cuando se la necesite llamar a esta matriz para alguna operación, se digita X para hacer referencia a la matriz antes ingresada. (Zarzar, 2009)

2.3.6. Expresiones en MATLAB

MATLAB tiene un lenguaje de programación que se caracteriza por proporcionar expresiones en matemáticas que involucra matrices completas. (Villarreal, 2015)

Las expresiones son las siguientes:

- Variables
- Números
- Funciones
- Operadores

2.3.6.1. Variables

A diferencia otros lenguajes de programación, en MATLAB no es necesario declarar que tipo de datos se va a ingresar (sea entero, punto flotante, complejos, etc) ni dimensiones. Al momento que MATLAB identifica una variable nueva, de manera automática se crea la variable y se le reserva una cierta cantidad de localidades de memoria que el programa cree necesarias, en caso de que se necesite más localidades de memoria para la variable se le agrega manualmente. (Villarreal, 2015)

2.3.6.2. Números

En MATLAB se usa una notación decimal común con un punto decimal que es de manera opcional y el signo *menos* (-) para señalar a los números negativos. Para poner una notación científica se debe usar la letra *e* y especificar la potencia deseada con base de 10. A los números imaginarios se los identifica porque se los debe representar con la letra *i* o la *j* como sufijo. (Zarzar, 2009)

2.3.6.3. Funciones

Matlab posee una extensa variedad de funciones matemáticas simples y avanzadas, en su mayoría las funciones son están programadas para aceptar argumentos complejos. Dos de las funciones más comunes son *sqrt* y *sin* son ya parte del núcleo de MATLAB por lo que no es posible acceder a su código. Para el resto de las funciones están programadas en archivos M y se puede acceder a su código para una simple revisión o para modificarlas. (Zarzar, 2009)

Algunas de las funciones especiales nos facilitan y otras necesitan valores constantes, MATLAB tiene ya incorporado constantes matemáticas y símbolos, los cuales se van a mostrar en la Tabla 2.2. (Villarreal, 2015)

Tabla 2. 2: Símbolos y constantes de MATLAB.

Constante	Significado
pi	3.14159265...
i	Unidad imaginaria, $\sqrt{-1}$
j	Igual que i
eps	Precisión relativa en punto flotante, 2^{-52}
realmin	Número más pequeño representable en punto flotante, 2^{-1022}
realmax	Número más grande representable en punto flotante, $(2-e) 2^{1023}$
Inf	Infinito
NaN	No es un número

Fuente: (Villarreal, 2015)

2.3.6.4. Operadores

En las operaciones que se realizan en MATLAB se utilizan los operadores aritméticos más comunes, son los mismos operadores que se

utilizan en otros lenguajes de programación y tienen un orden parecido a los que se usan en otros lenguajes de programación. En la Tabla 2.3 están los operadores más utilizados en MATLAB. (Zarzar, 2009)

Tabla 2. 3: Operadores más comunes en MATLAB.

Operador	Operación matemática
+	Suma
-	Resta
*	Multiplicación
/	División
^	Potencia
'	Transpuesta compleja conjugada
()	Especifica el orden de evaluación

Fuente: (Zarzar, 2009)

2.3.7. Matrices

Como ya se mencionó antes MATLAB es un programa que se maneja con matrices, por eso es necesario mencionar algunos de los comandos que son para generación de matrices. En la Tabla 2.4 se presentan unos cuantos comandos para generar matrices. (Villarreal, 2015)

Tabla 2. 4: Comandos para genera matrices.

Comando	Función
Zeros	Todos los elementos de la matriz son ceros
Ones	Todos los elementos de la matriz son unos
Rand	Genera una matriz con de elementos con distribución uniforme
Randn	Genera una matriz con elementos con distribución normal

Fuente: (Villarreal, 2015)

Ejemplo 1: se debe de generar una matriz de ceros de 3 reglones por 4 columnas.

```
>>X=zeros (3,4)
```

X=

```
0 0 0 0
0 0 0 0
0 0 0 0
```

Ejemplo 2: generar una matriz de 3x3 que todos sus elementos sean igual a 7.

```
>>A =7*ones (3x3)
```

A =

```
7 7 7
7 7 7
7 7 7
```

Ejemplo 3: crear un vector de 10 valores aleatorios entre 1 y 8 con distribución normal.

```
>>M = fix(10*rand(1,10))
```

M =

```
8 5 3 7 4 2 8 6 7 5
```

El comando *fix* se utilizó para redondear el valor entero inferior inmediato.

2.3.8. Uso de archivos -M en una simulación

Todos los archivos que contengan el código de simulación de MATLAB se denominan archivos -M. Estos pueden ser archivos que permitan argumentos de entrada y generen una salida, o más sencillos que puedes ser *script* en los cuales se muestren una serie de comando de MATLAB. Para que MATLAB reconozca este tipo de archivos deben de tener la extensión .m; este tipo de archivos se los puede hacer en cualquier tipo de editor de texto. (Zarzar, 2009)

2.3.8.1. Scripts

Son el tipo más sencillo de archivos -M que tiene MATLAB, no poseen argumentos de salida. Los *scripts* son de gran utilidad cuando se necesita automatizar un comando de MATLAB y se nos complica hacerlo desde la línea de comando. Estos *scripts* se ejecutan en el área de trabajo o también crean nuevas variables; por lo que pueden utilizar cualquier variable que se haya introducido en el área de trabajo o pueden crear variables nuevas las cuales quedaran disponibles para usarse una vez que se haya terminado de ejecutar el *script*. (Rapallini, 2009)

2.3.8.2. Funciones

Este es otro tipo de archivos -M, las funciones aceptan argumentos de entrada y salida. Las funciones se ejecutan dentro de su área de trabajo propia, el cual se crea al ejecutarse, al cual no se puede ingresar desde el espacio de trabajo de MATLAB. (Rapallini, 2009)

Capítulo 3: Simulación y Resultados obtenidos

Una vez que ya se tiene claro y se ha reforzado la teoría, revisando las fórmulas que se va a necesitar para calcular los parámetros de un enlace satelital, en este capítulo se presentará la aplicación que fue diseñada en el programa MATLAB para simular el cálculo de los parámetros. A lo largo del capítulo se detallará las simulaciones mediante las interfaces que se crearon y se enlazaron para que se pueda acceder de manera directa al parámetro que se necesite calcular.

En la aplicación que se diseñó en MATLAB se crearon once interfaces por medio de “guides”, las cuales estuvieron conformados por: pushbutton, static text y edit string.

3.1. Creación de interfaces con guides.

Se utilizó las guides para la creación de las interfaces como ya se mencionó antes, a continuación, se detallará los pasos para crear una interfaz sencilla con guides:

Paso 1: En la ventana de Command Window se escribe la palabra guide o también se puede ingresar directamente desde su icono en el entorno de desarrollo de MATLAB como se muestra en la Figura 3.1.

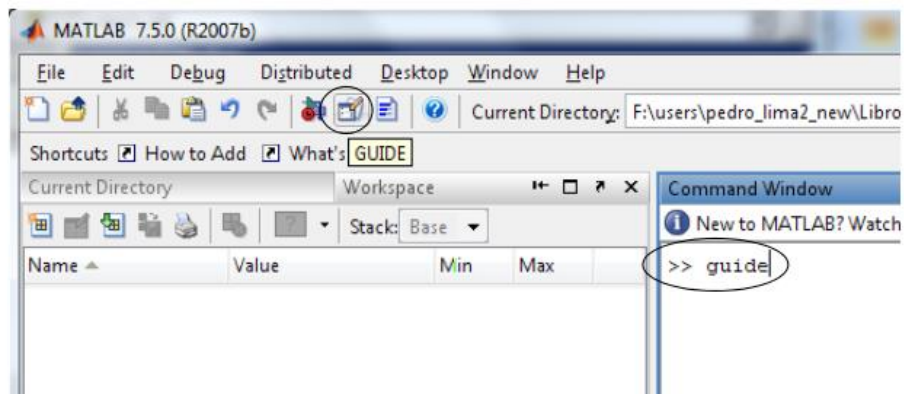


Figura 3. 1: Ingresar a una guide.
Elaborado por: Autor.

Paso 2: se nos abrirá una pequeña ventana en la que se selecciona la opción Blank GUI y se da clic en el botón OK, como lo se ve a continuación en la Figura 3.2.

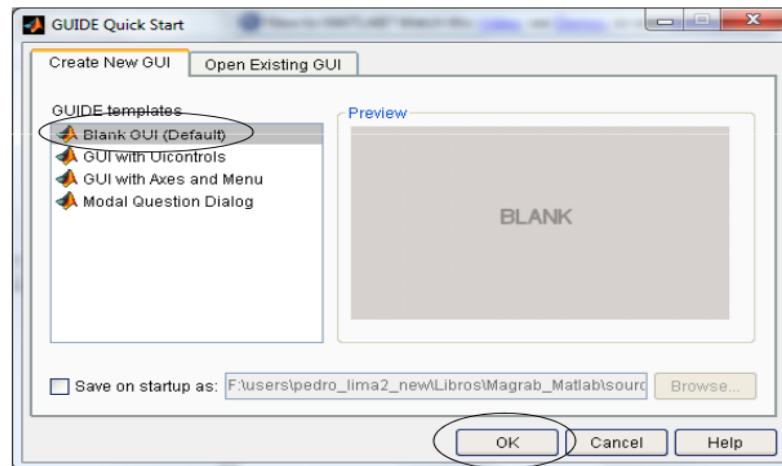


Figura 3. 2: Selección de guide.
Elaborado por: Autor.

Paso 3: luego aparecerá una ventana en cual se puede editar la guide de acuerdo con lo que se necesite, a continuación, la Figura 3.3 se muestra la ventana en la que se puede editar la guide.

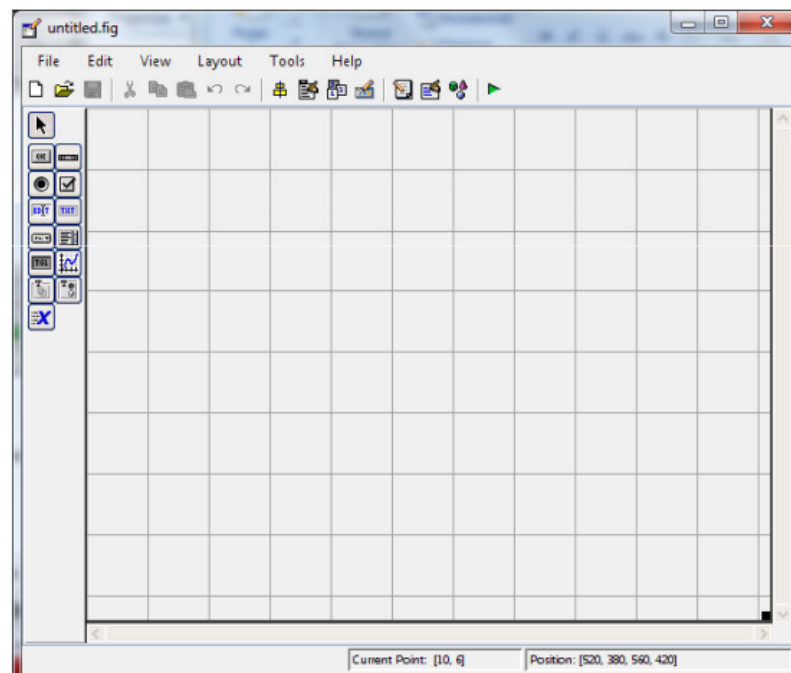


Figura 3. 3: Ventana de edición de la guide,
Elaborado por: Autor,

Paso 4: para modificar la guide se tendrá una paleta al lado izquierdo de esta, en la cual se tiene elementos tales como panel, static text, pop-up menú, push button, axes, etc.

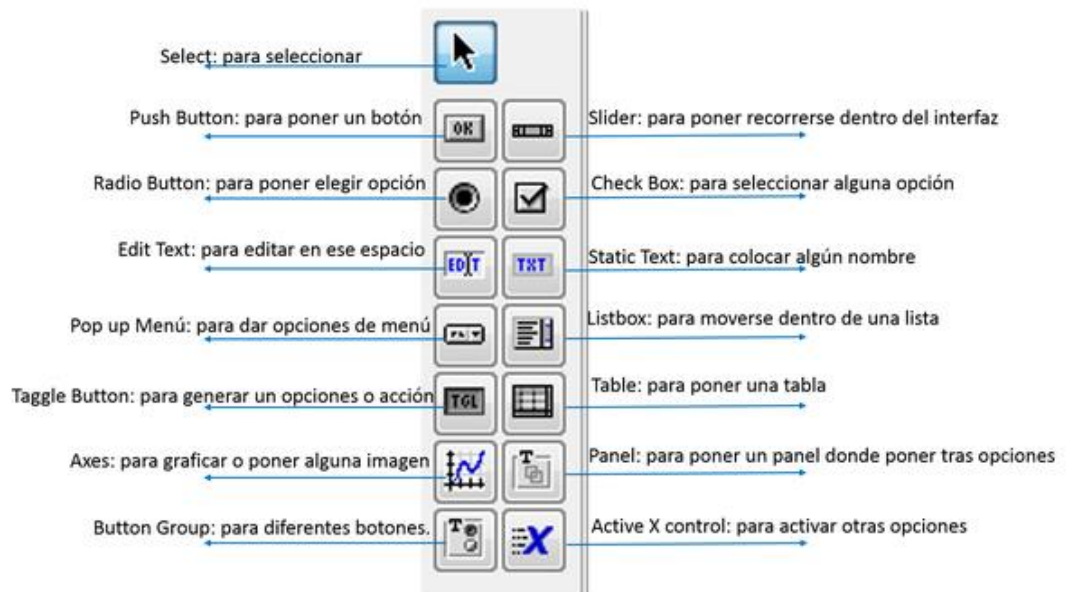


Figura 3. 4: Paleta de componentes de la guide.
Elaborado por: Autor.

Paso 5: se da clic en la opción play que se encuentra en la parte superior de la guide a lo que MATLAB nos mostrara un cuadro de dialogo que nos pregunta si se quiere guardar los cambios realizados (lo que se puede hacer dando clic en el botón save), se da clic en sí y se nombra al archivo como se desee, esto creara dos archivos, uno con extensión .m (este tendrá el código de programación de la guide) y el otro con extensión .fig (el cual contiene el diseño de la guide en binario).

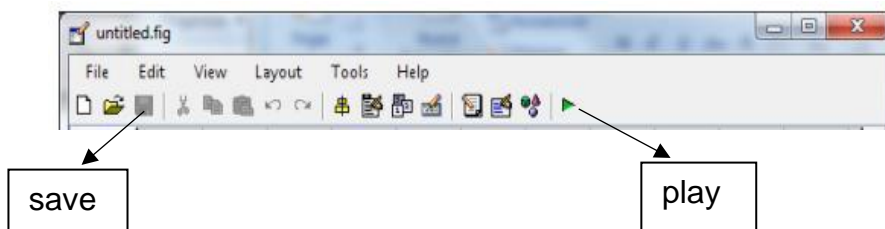


Figura 3. 5: Opción de guardar y play de la guide
Elaborado por: Autor,

Paso 6: se ingresa los elementos que se necesite para crear nuestra guide, en los *pushbuttons* se puede declarar la operación a realizar, dando clic derecho en el *pushbutton* y buscando la opción *Callback*.

Paso 7: se nos abrirá la ventana con el nombre que se le haya asignado con extensión .m y aquí se podrá modificar el código de programación de la guide.

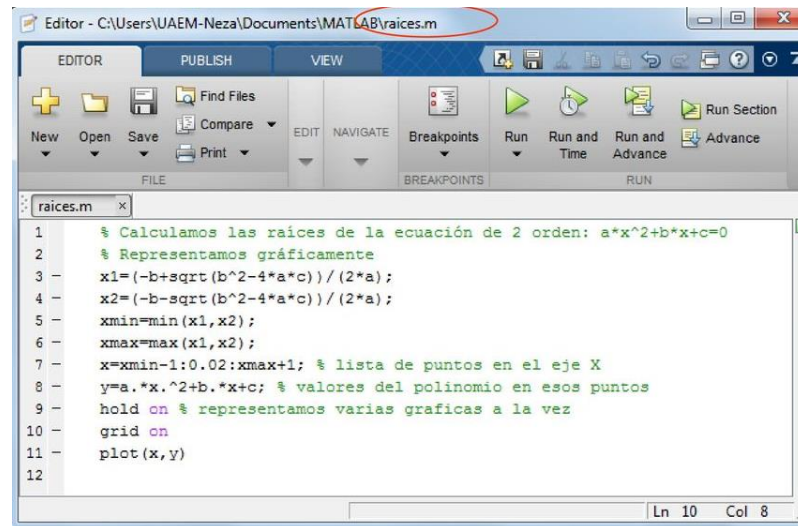


Figura 3. 6: Archivo con extensión .m.
Elaborado por: Autor.

Paso 8: una vez que se modifique el código de programación del archivo con extensión .m se guarda los cambios realizados en esta y se oprime el botón play, donde se tendrá nuestra guide finalizada, en la que se podrá realizar las operaciones que se haya declarado en el código.

3.2. Diseño de la aplicación

La aplicación que se diseñó está compuesta de 11 guides a las que se las denomina “interfaz”, a continuación, se procederá a mostrar las interfaces y los códigos de programación que se cambió en cada interfaz.

3.2.1. Interfaz 1

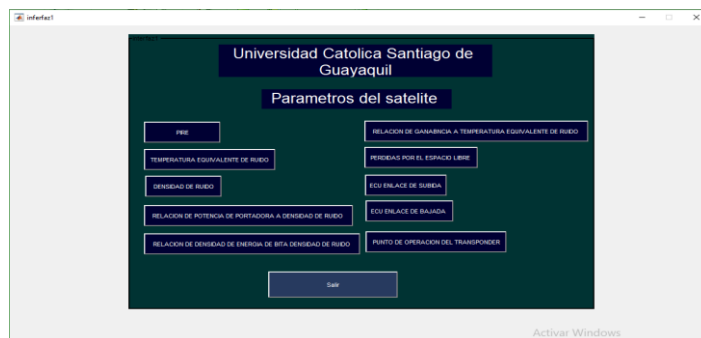


Figura 3. 7: Interfaz 1.
Elaborado por: Autor.

En la interfaz 1 se ingresaron 10 pushbutton los cuales nos enlazaran a las demás GUIs, para el botón de PIRE se lo enlazo con la interfaz 2, el botón de Temperatura Equivalente de Ruido se enlazo con la interfaz 3 y así sucesivamente con todos los botones, a continuación, se va a mostrar el código de programación que se utilizó para enlazar cada botón a su respectiva interfaz. A continuación, se muestra el código de programación que se utilizó en la interfaz 1:

```
% --- Executes on button press in pire.
function pire_Callback(hObject, eventdata, handles)
% hObject    handle to pire (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
interfaz2();

% --- Executes on button press in te.
function te_Callback(hObject, eventdata, handles)
% hObject    handle to te (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
interfaz3();

% --- Executes on button press in den.
function den_Callback(hObject, eventdata, handles)
% hObject    handle to den (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
untitled4();

% --- Executes on button press in denpor.
function denpor_Callback(hObject, eventdata, handles)
% hObject    handle to denpor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
inferfaz5();

% --- Executes on button press in denbit.
function denbit_Callback(hObject, eventdata, handles)
% hObject    handle to denbit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
interfaz6();

% --- Executes on button press in gtr.
function gtr_Callback(hObject, eventdata, handles)
% hObject    handle to gtr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
interfza7();

% --- Executes on button press in pel.
function pel_Callback(hObject, eventdata, handles)
% hObject    handle to pel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```

interfaz8();

% --- Executes on button press in ecusub.
function ecusub_Callback(hObject, eventdata, handles)
% hObject    handle to ecusub (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
interfaz9();

% --- Executes on button press in ecubaj.
function ecubaj_Callback(hObject, eventdata, handles)
% hObject    handle to ecubaj (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
interfaz10();

% --- Executes on button press in trans.
function trans_Callback(hObject, eventdata, handles)
% hObject    handle to trans (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
interfaz11();

% --- Executes on button press in exit.
function exit_Callback(hObject, eventdata, handles)
% hObject    handle to exit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close();

```

3.2.2. Interfaz 2 (PIRE)



Figura 3. 8: Interfaz 2.
Elaborado por: Autor.

En la interfaz 2 se realiza la operación que se especificó en la Ecuación 2.4, por lo que se declararon 4 variables, para lo que se añade un panel a la guide en la cual se van a introducir 4 edit text (son los valores que se va a ingresar), el ultimo edit text es donde se va mostrar el resultado de la operación, se introdujeron 4 static text (los cuales son para saber que

parámetros se van a ingresar y en que unidades se deben ingresar), se introdujeron 2 pushbutton, uno de ellos el botón Calcular (ya se declaró la operación que se va a realizar) una vez que se lo presione va a realizar la operación y nos mostrara el resultado, y el botón Inicio que nos cierra la interfaz2 y nos lleva a la interfaz1.

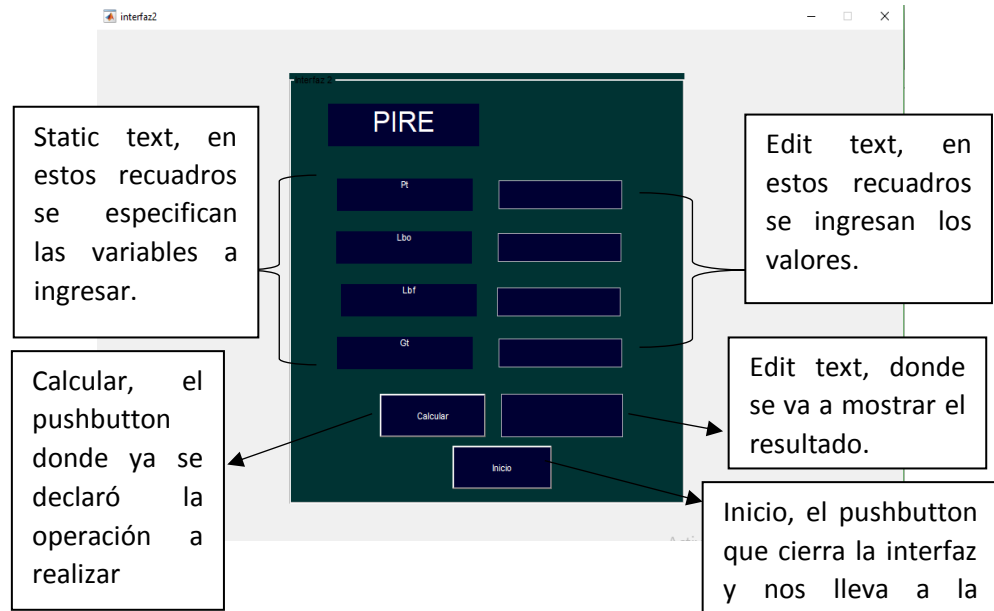


Figura 3. 9: Explicación de los botones de la interfaz.
Elaborado por: Autor.

Cabe mencionar que los valores a introducir de cada variable sus unidades deben ser en dB. A continuación, se va a mostrar el código de programación que se utilizó para configurar la interfaz 2.

```
function pt_Callback(hObject, eventdata, handles)
% hObject    handle to pt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A
A=str2double(get(hObject,'String'));

function lbo_Callback(hObject, eventdata, handles)
% hObject    handle to lbo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global B
B=str2double(get(hObject,'String'));

function lbf_Callback(hObject, eventdata, handles)
% hObject    handle to lbf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global C
C=str2double(get(hObject,'String'));
```

```

function gt_Callback(hObject, eventdata, handles)
% hObject    handle to gt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global D
D=str2double(get(hObject,'String'));

% --- Executes on button press in back1.
function back1_Callback(hObject, eventdata, handles)
% hObject    handle to back1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(interfaz2);

% --- Executes on button press in calcular.
function calcular_Callback(hObject, eventdata, handles)
% hObject    handle to calcular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
global B;
global C;
global D;
P=A-B-C+D;
set(handles.calculart,'String',P);

```

3.2.3. Interfaz 3 (Temperatura Equivalente de Ruido)

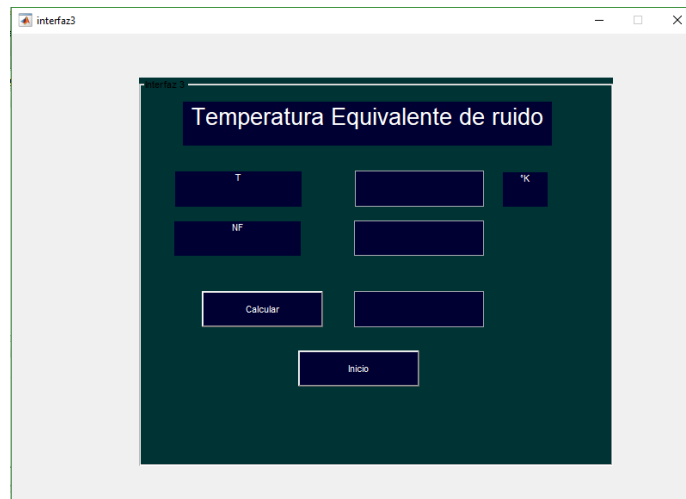


Figura 3. 10: Interfaz 3.
Elaborado por: Autor.

En la interfaz 3 se resuelve la Ecuación 2.8, y a diferencia de la anterior aquí solo se tendrá que ingresar dos variables. A continuación, podrán observar el código de programación empleado:

```

function t_Callback(hObject, eventdata, handles)
% hObject    handle to t (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
A=str2double(get(hObject,'String'));

```



```

function n_Callback(hObject, eventdata, handles)
% hObject    handle to n (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global B;
B=str2double(get(hObject,'String'));

function f_Callback(hObject, eventdata, handles)
% hObject    handle to f (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global C;
C=str2double(get(hObject,'String'));

% --- Executes on button press in Calcular.
function Calcular_Callback(hObject, eventdata, handles)
% hObject    handle to Calcular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
global B;
T=A*(B-1);
set(handles.calculart,'String',T);

% --- Executes on button press in back2.
function back2_Callback(hObject, eventdata, handles)
% hObject    handle to back2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(interfaz3);

```

3.2.4. Interfaz 4 (Densidad de Ruido)



Figura 3. 11: Interfaz 4.
Elaborado por: Autor.

En la interfaz 4 se ingresan los valores necesarios para resolver la ecuación 2.10, esta interfaz es similar a la anterior, pero los valores a ingresar son los que varían. A continuación, se muestra el código de programación que se usó para esta interfaz:

```

function te_Callback(hObject, eventdata, handles)
% hObject    handle to te (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
A=str2double(get(hObject,'String'));

function K_Callback(hObject, eventdata, handles)
% hObject    handle to K (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global B;
B=str2double(get(hObject,'String'));

% --- Executes on button press in calcular.
function calcular_Callback(hObject, eventdata, handles)
% hObject    handle to calcular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
global B;
P=A*B
set(handles.calculard,'String',P);

% --- Executes on button press in back3.
function back3_Callback(hObject, eventdata, handles)
% hObject    handle to back3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(untitled4);

```

3.2.5. Interfaz 5 (Relación de Potencia de Portadora a Densidad de Ruido)



Figura 3. 12: Interfaz 5.
Elaborado por: Autor.

En esta interfaz como en las anteriores se resuelve una de las ecuaciones antes mencionadas en este trabajo, para este caso se resuelve la Ecuación 2. 11, a diferencia de la interfaz anterior en esta se tiene que ingresar 3 valores para poder resolver la operación. A continuación, la programación utilizada:

```

function C_Callback(hObject, eventdata, handles)
% hObject      handle to C (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global A;
A=str2double(get(hObject,'String'));

function K_Callback(hObject, eventdata, handles)
% hObject      handle to K (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global B;
B=str2double(get(hObject,'String'));

function edit5_Callback(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global C;
C=str2double(get(hObject,'String'));

% --- Executes on button press in calcular.
function calcular_Callback(hObject, eventdata, handles)
% hObject      handle to calcular (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global A;
global B;
global C;
R=A/(B*C)
set(handles.calcularr,'String',R);

% --- Executes on button press in back5.
function back5_Callback(hObject, eventdata, handles)
% hObject      handle to back5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close(inferfaz5);

```

3.2.6. Interfaz 6 (Relación de Densidad de Energía de Bit a Densidad de Ruido)

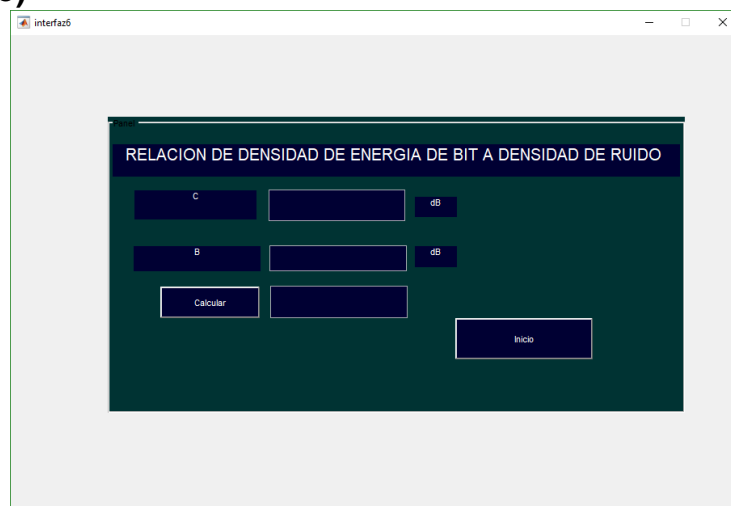


Figura 3. 13: Interfaz 6.
Elaborado por: Autor.

En la interfaz 6 se resuelve la Ecuación 2. 14, y a diferencia de la anterior en este se tiene que ingresar dos valores para poder resolver la operación. A continuación, se muestra la programación utilizada:

```
function C_Callback(hObject, eventdata, handles)
% hObject    handle to C (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
A=str2double(get(hObject,'String'));

function B_Callback(hObject, eventdata, handles)
% hObject    handle to B (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global B;
B=str2double(get(hObject,'String'));

% --- Executes on button press in calcular.
function calcular_Callback(hObject, eventdata, handles)
% hObject    handle to calcular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
global B;
RE=A-B
set(handles.calcularre,'String',RE);

% --- Executes on button press in back6.
function back6_Callback(hObject, eventdata, handles)
% hObject    handle to back6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(interfaz6);
```

3.2.7. Interfaz 7 (Relación de Ganancia a Temperatura de Ruido)

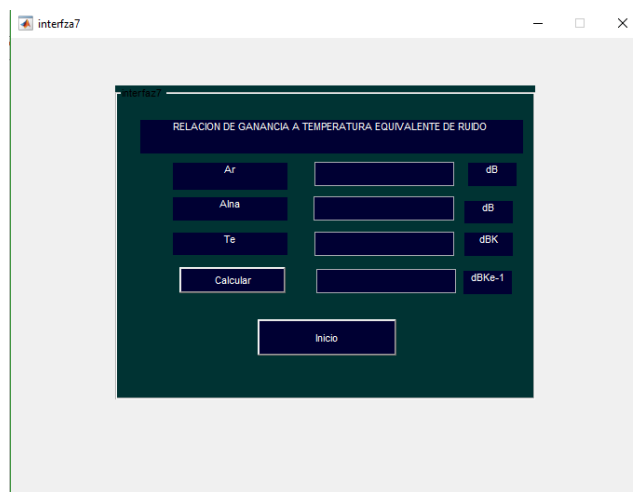


Figura 3. 14: Interfaz 7.
Elaborado por: Autor.

En la interfaz 7 se pasa a resolver la Ecuación 2. 16 para lo cual se tendrá que ingresar 3 valores para poder resolver la operación, los valores a ingresar deben tener sus unidades en dB. A continuación, se muestra el código de programación utilizado:

```
function gr_Callback(hObject, eventdata, handles)
% hObject    handle to gr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
A=str2double(get(hObject,'String'));

function glna_Callback(hObject, eventdata, handles)
% hObject    handle to glna (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global B;
B=str2double(get(hObject,'String'));

function te_Callback(hObject, eventdata, handles)
% hObject    handle to te (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global C;
C=str2double(get(hObject,'String'));

% --- Executes on button press in caclular.
function caclular_Callback(hObject, eventdata, handles)
% hObject    handle to caclular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
global B;
global C;
GT=(A+B)-C;
set(handles.calculargte,'String',GT);

% --- Executes on button press in back7.
function back7_Callback(hObject, eventdata, handles)
% hObject    handle to back7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close();
```

3.2.8. Interfaz 8 (Perdidas del Espacio Libre)

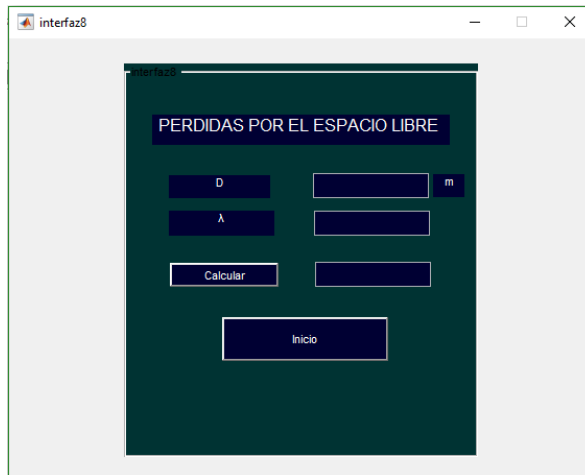


Figura 3. 15: Interfaz 8.
Elaborado por: Autor.

En esta interfaz se da solución a Ecuación 2. 17 para lo que se tendrá que ingresar dos valores para dar poder resolver la operación a diferencia de la anterior. A continuación, se muestra la programación utilizada:

```
function d_Callback(hObject, eventdata, handles)
% hObject    handle to d (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
A=str2double(get(hObject,'String'));

function landa_Callback(hObject, eventdata, handles)
% hObject    handle to landa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global B;
B=str2double(get(hObject,'String'));

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
global B;
PEL=((4*pi*A)/B)^2;
set(handles.calcularpel,'String',PEL);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close();
```

3.2.9. Interfaz 9 (Ecuación del Enlace de Subida)

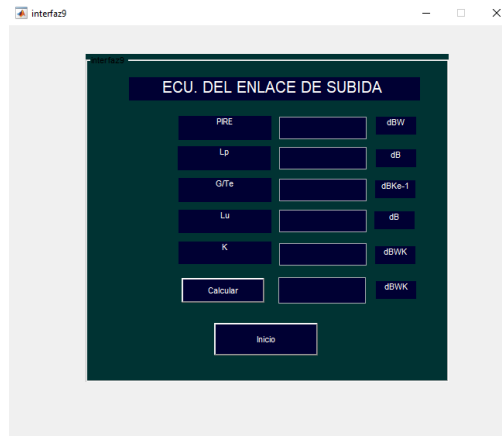


Figura 3. 16: Interfaz 9.
Elaborado por: Autor.

En esta interfaz a diferencia de la anterior se debe de ingresar 5 valores para poder resolver la Ecuación 2. 20, a diferencia de la interfaz anterior, los valores que se debe de ingresar tienen que tener sus unidades en dB. A continuación, se ve el código de programación empleado:

```
function gt_Callback(hObject, eventdata, handles)
% hObject    handle to gt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
A=str2double(get(hObject,'String'));

function pr_Callback(hObject, eventdata, handles)
% hObject    handle to pr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global B;
B=str2double(get(hObject,'String'));

function d_Callback(hObject, eventdata, handles)
% hObject    handle to d (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global C;
C=str2double(get(hObject,'String'));

function landa_Callback(hObject, eventdata, handles)
% hObject    handle to landa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global D;
D=str2double(get(hObject,'String'));

function g_Callback(hObject, eventdata, handles)
% hObject    handle to g (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global E;
```

```

E=str2double(get(hObject,'String'));

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
global B;
global C;
global D;
global E;
TOT=A-B+C-D-E;
set(handles.calcularecusub,'String',TOT);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close();

```

3.2.10. Interfaz 10 (Ecuación del Enlace de Bajada)

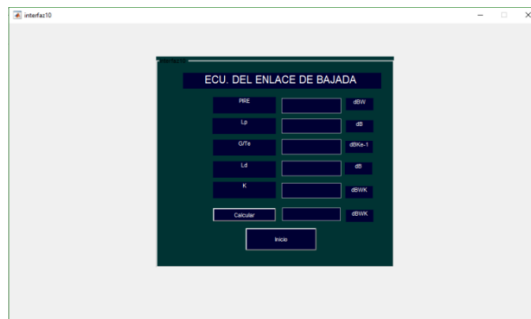


Figura 3. 17: Interfaz 10.
Elaborado por: Autor.

En esta interfaz se ingresa 5 valores para dar solución a la Ecuación 2.23, esta interfaz es similar a la anterior por eso están identificadas muy bien las variables a ingresar y de la misma manera deben de tener las unidades en dB. Se pasa a observar el código de programación:

```

function GT_Callback(hObject, eventdata, handles)
% hObject    handle to GT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
A=str2double(get(hObject,'String'));

function PR_Callback(hObject, eventdata, handles)
% hObject    handle to PR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global B;
B=str2double(get(hObject,'String'));

```



```

function D_Callback(hObject, eventdata, handles)
% hObject   handle to D (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
global C;
C=str2double(get(hObject,'String'));

function landa_Callback(hObject, eventdata, handles)
% hObject   handle to landa (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
global D;
D=str2double(get(hObject,'String'));

function Ld_Callback(hObject, eventdata, handles)
% hObject   handle to Ld (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
global E;
E=str2double(get(hObject,'String'));

% --- Executes on button press in calcular.
function calcular_Callback(hObject, eventdata, handles)
% hObject   handle to calcular (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
global A;
global B;
global C;
global D;
global E;
TOT=A-B+C-D-E;
set(handles.calcularq,'String',TOT);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject   handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
close();

```

3.2.11. Interfaz 11 (Punto de Operación del Transponder)

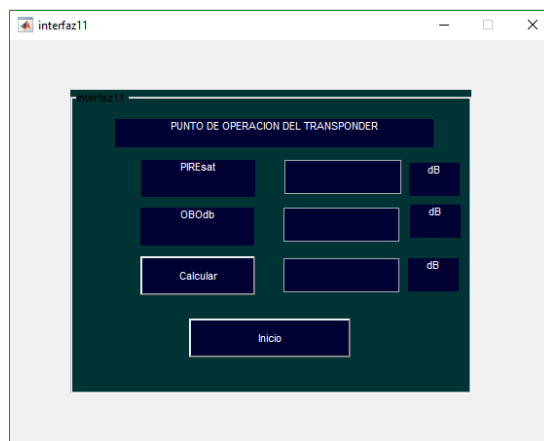


Figura 3. 18: Interfaz 11.
Elaborado por: Autor.

En la última interfaz se da solución a la Ecuación 2. 32, a diferencia de la interfaz anterior en esta se tendrá que ingresar solo 2 valores los cuales deben de tener sus unidades en dB. Se muestra el código de programación a continuación:

```
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
A=str2double(get(hObject,'String'));

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global B;
B=str2double(get(hObject,'String'));

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global A;
global B;
TRANS=A-B
set(handles.calcularpotrans,'String',TRANS);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close();
```

3.3. Demostración de la aplicación

Demostración de PIRE usando la aplicación diseñada:

Ejemplo 1: en una estación terrena se tiene que la antena tiene una potencia de entrada de 40dB, 3dB de pérdida por reducción, 3dB entre la pérdida total por ramificador y alimentador, y 40dB de ganancia de la antena que transmite. (Tomasi, 2003)



Figura 3. 19: Demostración de PIRE.
Elaborado por: Autor.

Demostración de la Temperatura Equivalente de Ruido en la aplicación.
Ejemplo 2: transformar las figuras de ruido de 4dB y 4.1dB a temperaturas equivalentes de ruido. Se debe de usar una temperatura de 300 K como la temperatura ambiente.

NF=4dB, F = 2.512; NF=4.1dB, F = 2.57

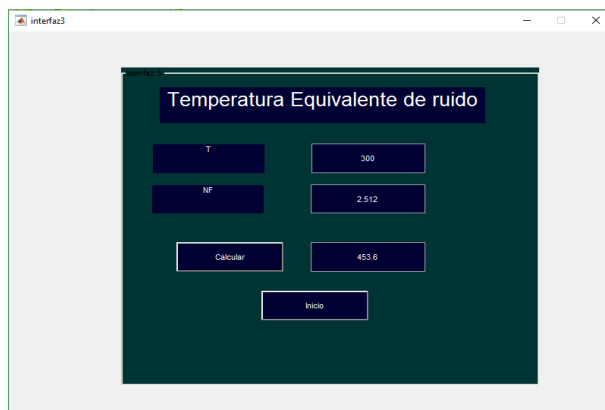


Figura 3. 20: Demostración de la Te.
Elaborado por: Autor.

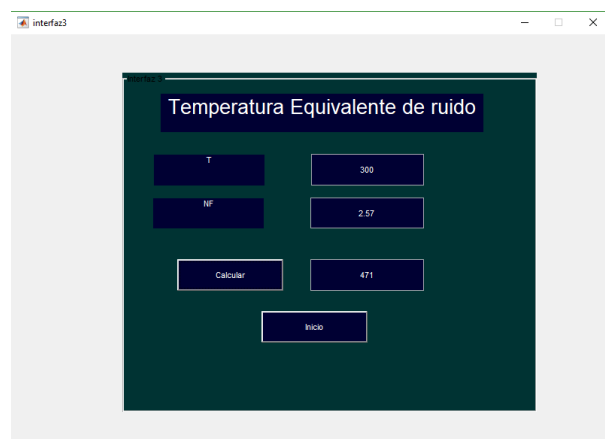


Figura 3. 21: Demostración de Temperatura equivalente de ruido.
Elaborado por: Autor.

Demostración de la Densidad de Ruido. Ejemplo 3: calcular la densidad de ruido, usando dos temperaturas equivalentes distintas: 458 y 500 K. tener en cuenta que la constante de Boltzmann es igual a $1.30e-23$.



Figura 3. 22: Demostración 2 de la No.
Elaborado por: Autor.



Figura 3. 23: Demostración de Densidad de ruido.
Elaborado por: Autor.

Demostración de Relación de Potencia de Portadora a Densidad de Ruido. Ejemplo 4: calcular la C/N_0 , teniendo en cuenta que potencia de la portadora es de 117.1dBW y la temperatura ambiente es de 350 K.



Figura 3. 24: Demostración de Potencia de portadora a densidad de ruido.
Elaborado por: Autor.

Demostración de la Relación de Densidad de Energía de Bit a Densidad de Ruido. Ejemplo 5: calcular E_b/N_0 , teniendo en cuenta que la relación C/N es igual a 8.8dB y la relación de B/f_b es igual a 3.5dB.

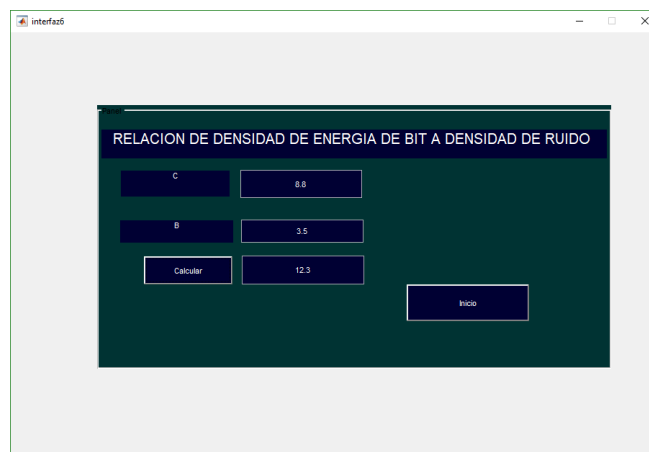


Figura 3. 25: Demostración de la Relación de densidad de energía de bit a densidad de ruido.
Elaborado por: Autor.

Demostración de la Relación de Ganancia a Temperatura equivalente de ruido. Ejemplo 6: se desea calcular la relación G/T_e para un satélite transpondedor que posee una ganancia de la antena que recibe de 12dB, una ganancia del LNA de 10dB, y una T_e de 26dBK.

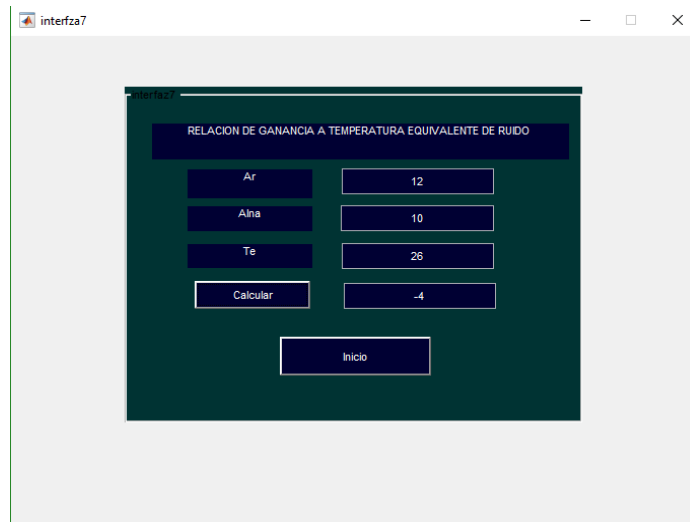


Figura 3. 26: Demostración de la relación G/Te.
Elaborad por: Autor.

Demostración de la ecuación del enlace subida. Ejemplo 7: calcular el valor de la ecuación del enlace de subida teniendo en cuenta que el PIRE es de 90dBW, las pérdidas del espacio libre son igual a 206.5dB, la relación G/Te es de -5.3dBK^{-1} , las perdidas adicionales de la atmosfera son de 0.6dB.

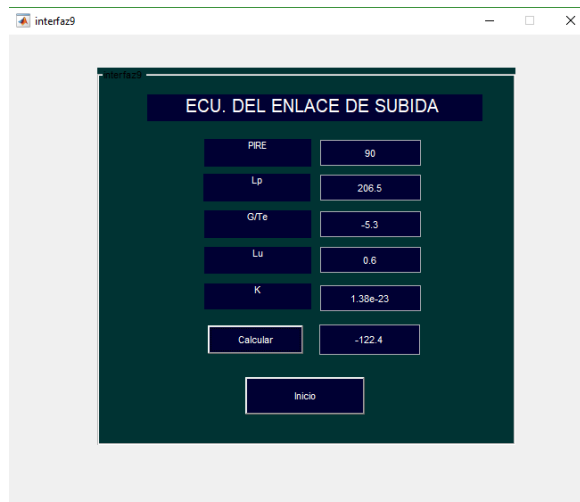


Figura 3. 27: Demostración de la Ecuación del enlace de subida.
Elaborado por: Autor.

Demostración de la ecuación del enlace de bajada. Calcular el valor de la ecuación del enlace de subida teniendo en cuenta que el PIRE es de 45dBW, las pérdidas del espacio libre son igual a 206.5dB, la relación G/Te es de 37.7dBK^{-1} , las perdidas adicionales de la atmosfera son de 0.4dB.



Figura 3. 28: Demostración de la ecuación del enlace de bajada.
Elaborado por: Autor.

Capítulo 4: Conclusiones y Recomendaciones

4.1. Conclusiones

- A partir de la investigación realizada en el presente trabajo se concluye que el programa MATLAB sirve de complemento para los estudiantes de la carrera de Ingeniería en Telecomunicaciones en la resolución de ejercicios de cálculos de los parámetros de un enlace satelital.
- Los beneficios del programa diseñado van dirigidos tanto a estudiantes como docentes ya que se presentó de una manera sencilla, clara y práctica el cálculo de los parámetros de un enlace satelital mediante el uso de MATLAB.
- Con el uso de botones dentro de cada interfaz el programa se hace práctico, didáctico y de uso efectivo para la resolución de los problemas propuestos en clase o fuera de ella.
- Quedo en evidencia que MATLAB es un programa que brinda un abanico de posibilidades y de usos que al utilizarse de una manera práctica y sin necesidad de dominar la programación ayudando a la creación de la aplicación detallada en el trabajo.
- El programa MATLAB es una herramienta que puede usarse como material didáctico en la mayoría de las materias para reforzar la teoría que se da en el aula de clase.

4.2. Recomendaciones

- Se recomienda tener en cuenta el tipo de satélite con el que se trabaja, para que fue diseñado, la banda en que opera y la órbita en la que se encuentra para poder hablar de sus parámetros.
- El uso de toda la variedad de botones que ofrece el programa de MATLAB para otras investigaciones a partir del trabajo desarrollado. de cada interfaz es recomendable para que el programa sea activo, dinámico y práctico.
- Es recomendable enlazar los resultados de cada interfaz para ingresar valores que no se han calculado.

- El uso de la herramienta SIMULINK que viene dentro del paquete de MATLAB es aconsejable para poder realizar este tipo de simulaciones en futuros trabajos porque permite una variedad de resultados.

Referencias Bibliográficas

Amado, F. (2008). *Técnicas de acceso múltiple y algoritmos de detección de múltiples usuarios en telefonía celular: análisis comparativo*. Recuperado a partir de <http://eie.ucr.ac.cr/uploads/file/proybach/pb0802t.pdf>

Borrell, G. (2007). *Introducción informal a matlab y octave*, 178.

Giner, S. (2008). *Un lenguaje de programación de alto nivel*, 124.

Hernandez, C., Corredor, O., & Pedraza, L. (2010). *Metodología para el diseño de enlaces satelitales*, 14, 16.

Jami, J., & Vega, I. (2012). *Implementación de una guía práctica de instalación y operación de un sistema de recepción de televisión satelital en banda C para 18 satélites que tienen huella en Ecuador*. Recuperado a partir de <http://bibdigital.epn.edu.ec/bitstream/15000/4964/1/CD-4503.pdf>

Rodríguez, F. (2008). *Spread spectrum, ganancia de procesamiento y la protección contra interferencia: "antijamming"*, 6.

Tomasi, W. (2010). *Sistemas de comunicaciones electrónicas*. Pearson Educación. Recuperado a partir de <https://hellsingge.files.wordpress.com/2014/08/sistemas-de-comunicaciones-electronicas-tomasi-4ta-edicion3b3n.pdf>

Vega, C. P., Maza, J. M. Z. S. de la, & López, A. C. (2007). *Sistemas de telecomunicación*. Ed. Universidad de Cantabria. Recuperado a partir de <https://books.google.com.ec/books?id=y5s3XlaE46UC&printsec=frontcover&hl=es#v=onepage&q&f=false>

Villarreal, A. (2015). *Desarrollo de una aplicación para determinar los parámetros físicos y lógicos de un enlace satelital en empresas corporativas*. Recuperado a partir de

<http://dspace.udla.edu.ec/bitstream/33000/4262/1/UDLA-EC-TIRT-2015-06%28S%29.pdf>

Zarzar, E. (2009). *Análisis del Desempeño de los Intercaladores Basados en Polinomios Cuadráticos de Permutación en Canales AWGN y Rayleigh*. Recuperado a partir de https://repositorio.itesm.mx/bitstream/handle/11285/569321/DocsTec_10045.pdf;jsessionid=1355BE307091B4189C0AC6D2CED686E9?sequence=1



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Mendoza Sánchez, Xavier Alexander** con C.C: # 0951605955 autor del Trabajo de Titulación: **Análisis y simulación de los parámetros de enlaces satelitales mediante Matlab/Simulink para fines didácticos** previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 7 de Marzo de 2018

f. _____

Nombre: Mendoza Sánchez, Xavier Alexander

C.C: 0951605955

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Análisis y simulación de los parámetros de enlaces satelitales mediante Matlab/Simulink para fines didácticos		
AUTOR(ES)	MENDOZA SANCHEZ, XAVIER ALEXANDER		
REVISOR(ES)/TUTOR(ES)	Ing. ZAMORA CEDEÑO, NÉSTOR ARMANDO		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería en Telecomunicaciones		
TÍTULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	7 DE Marzo de 2018	No. DE PÁGINAS:	75
ÁREAS TEMÁTICAS:	SISTEMAS SATELITALES, MATLAB, FUNDAMENTOS DE COMUNICACIÓN		
PALABRAS CLAVES/ KEYWORDS:	SATELITAL, MATLAB, INTERFAZ, PARÁMETROS, ENLACES, PROPAGACIÓN		
RESUMEN/ABSTRACT (150-250 palabras):	<p>El presente trabajo se realizó con intenciones didácticas para reforzar los conocimientos recibidos en el aula de clase sobre el cálculo de los parámetros de un enlace satelital, creando una aplicación mediante MATLAB teniendo como base los conocimientos previos adquiridos durante mi formación como profesional usando la herramienta GUI con la que se creó interfaces que facilitan el cálculo de los parámetros y sirve de ayuda para la comprensión de la teoría aplicada a la práctica. Además de servir como una herramienta de apoyo para los docentes durante la formación de profesionales en la carrera de Ingeniería en Telecomunicaciones y afines a esta. Se debe considerar que en la parte didáctica de los sistemas satelitales resulta favorable utilizar este tipo de programas de manera de ejemplo para complementar el aprendizaje. Durante el desarrollo del proyecto se presentó retos que fueron superados porque se puso en práctica conocimientos adquiridos en el salón de clase.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593982232281	E-mail: chavialex@hotmail.com	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Palacios Meléndez Edwin Fernando		
	Teléfono: +593-9-68366762		
	E-mail: edwin.palacios@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			