

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

Diseño de algoritmo para Medir y Caracterizar el Rango de Frecuencias asignado a Radiodifusión FM en un edificio central de Guayaquil

**PROYECTO INTEGRADOR**

Previo la obtención del Título de:

**Ingeniero en Electrónica y Telecomunicaciones**

Presentado por:

Rodrigo Farinango Peñarreta

Christopher Erwin Ortiz Pincay

GUAYAQUIL - ECUADOR

Año: 2018

## **DEDICATORIA**

Le dedico este trabajo a mis padres y hermano por siempre darme ánimos para seguir adelante y dar lo mejor cada día.

Le dedico este trabajo a mi papá, a mi mamá y a mis hermanas por ser pilar fundamental en mi vida y ayudarme a no rendirme.

## **AGRADECIMIENTOS**

Mi más sincero agradecimiento a mi familia por apoyarme en el transcurso de la carrera, a mis profesores por haberme guiado en el desarrollo de este trabajo.

Le agradezco a Dios y a mi familia por ser pacientes conmigo y ayudarme a ser una mejor persona cada día, a mis profesores por su dedicación en el seguimiento de este trabajo.

## **DECLARACIÓN EXPRESA**

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Christopher Erwin Ortiz Pincay y Rodrigo Farinango Peñarreta y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Christopher Erwin Ortiz  
Pincay

Rodrigo Farinango  
Peñarreta

# EVALUADORES

**M.Sc. Washington Medina**

PROFESOR DE LA MATERIA

**M.Sc. José Félix Moncayo**

PROFESOR TUTOR

## RESUMEN

El presente proyecto tiene como finalidad dar una solución a los problemas encontrados en cuanto a la recolección de datos del espectro FM y a su procesamiento. El GICOM, grupo de investigación de la ESPOL, está interesado en adquirirlo. La herramienta propuesta debe ser automatizada, permitir que el usuario escoja el tiempo y la emisora que desee medir y ser capaz de identificar que emisoras incumplen las normas técnicas de la ARCOTEL para frecuencia central, ancho de banda e intensidad de campo eléctrico.

Para la implementación del proyecto se utilizó una PC, el RTL-SDR y las librerías de RTL-SDR para Matlab y con ellas se crearon algoritmos para la medición del espectro por canal, la interfaz gráfica, el análisis de cada parámetro y la generación de los reportes con los parámetros analizados en una hoja de cálculo.

Una vez que se obtuvo el reporte final se encontró que los parámetros que se incumplieron fueron la frecuencia central y el ancho de banda con un porcentaje de 56 % y 9% respectivamente, produciendo que el 63% del total de las emisoras medidas incumplan la norma técnica.

Con los datos obtenidos de las emisoras de radiofrecuencia FM se realizó el diseño del reporte final, se logró identificar cuales presentan inconvenientes debido al incumplimiento de sus parámetros técnicos y de esta forma dar un seguimiento de su comportamiento a distintos horarios; por lo que se logró cumplir con los requerimientos del cliente.

**Palabras Clave:** GICOM, ARCOTEL, RTL-SDR

## **ABSTRACT**

*The purpose of this project is to provide a solution to the problems encountered in the collection of FM spectrum data and its processing. GICOM, the research group of ESPOL, is interested in acquiring it. The proposed tool must be automated, allow the user to choose the time and the radio station that it wants to measure and be able to identify which stations do not comply with ARCOTEL's technical standards for central frequency, bandwidth and electric field strength.*

*For the implementation of the project, a PC, the RTL-SDR and the RTL-SDR libraries for Matlab were used and with them, algorithms were created for the measurement of the spectrum by channel, the graphical interface, the analysis of each parameter and the generation of the reports with the parameters analyzed in a spreadsheet.*

*Once the final report was obtained, it was found that the parameters that were not complied with, were the central frequency and the bandwidth with a percentage of 56% and 9% respectively, producing that 63% of the total of the measured stations violate the norm technique.*

*With the data obtained from the FM radio frequency transmitters the final report was designed, it was possible to identify which present inconveniences due to the non-compliance of its technical parameters and in this way to monitor its behavior at different times; so, it was possible to meet the customer's requirements.*

*Keywords: GICOM, ARCOTEL, RTL-SDR.*

# ÍNDICE GENERAL

RESUMEN.....	I
ABSTRACT .....	II
ÍNDICE GENERAL .....	III
ABREVIATURAS.....	V
SIMBOLOGÍA.....	VI
ÍNDICE DE FIGURAS .....	VII
ÍNDICE DE TABLAS.....	IX
CAPÍTULO 1.....	1
1. Introducción .....	1
1.1 Descripción del problema .....	2
1.2 Justificación del problema .....	3
1.3 Objetivos .....	4
1.3.1 Objetivo General.....	4
1.3.2 Objetivos Específicos.....	4
1.4 Estado del arte .....	5
1.5 Marco Teórico .....	6
1.5.1 RTL-SDR .....	6
1.5.2 Emisoras de Radiodifusión .....	7
1.5.3 Cobertura.....	8
1.5.4 Interferencia de canal adyacente .....	9
1.5.5 Norma Técnica ARCOTEL Banda FM .....	10
1.5.6 MATLAB .....	11
1.5.7 Librerías del RTL-SDR para Matlab.....	13
1.5.8 Simulink .....	14
1.6 Técnicas de muestreo .....	14
1.7 Guardar e importar datos.....	15

1.7.1	Guardar e Importar Datos del RTL-SDR.....	15
1.7.2	Algunas ecuaciones a utilizar.....	16
1.7.3	Interfaz gráfica en MATLAB.....	17
1.8	Alcance.....	17
CAPÍTULO 2.....		18
2.	Metodología.....	18
2.1	Diseño de diagrama de flujo para la aplicación de medición y caracterización del espectro FM.....	19
2.2	Funcionamiento del RTL-SDR y conexión con MATLAB.....	20
2.3	Diseño de algoritmo para la adquisición y almacenamiento de datos del espectro FM.	22
2.4	Diseño de algoritmos para la obtención de parámetros para el espectro FM...	25
2.5	Diseño del algoritmo para generación automatizada del reporte.....	30
2.6	Algoritmo para descartar datos aberrantes.....	33
2.7	Diseño de la interfaz gráfica de la aplicación.....	34
CAPÍTULO 3.....		37
3.	Resultados Y ANÁLISIS.....	37
CAPÍTULO 4.....		49
4.	Conclusiones Y RECOMENDACIONES.....	49
4.1	Conclusiones.....	49
4.2	Recomendaciones.....	50
BIBLIOGRAFÍA.....		52
ANEXOS.....		54

## **ABREVIATURAS**

ESPOL	Escuela Superior Politécnica del Litoral.
SDR	Software Defined Radio (Radio definida por software).
UIT	Unión Internacional de Telecomunicaciones.
ARCOTEL	Agencia de Regulación y Control de las Telecomunicaciones.

## SIMBOLOGÍA

MHz	Mega hertz
$\mu\text{V}$	Microvoltio
dB	Decibelio
dB $\mu\text{V}/\text{m}$	Decibelio Microvoltio sobre metro
V	Voltio
W	Watts
$\Omega$	Ohms

# ÍNDICE DE FIGURAS

Figura 1.1 RTL-SDR y antena .....	6
Figura 1.2 RTL-SDR sin carcasa .....	6
Figura 1.3 Ventana de Matlab .....	11
Figura 1.4 Ventana Current Folder .....	12
Figura 1.5 Command Window Matlab .....	12
Figura 1.6 Ventana Workspace Matlab .....	13
Figura 1.7 Buscador de librerías y entorno de simulación simulink .....	14
Figura 1.8 Upsampling .....	14
Figura 1.9 Downsampling .....	15
Figura 1.10 Importar datos del RTL-SDR a MATLAB .....	15
Figura 1.11 Importar datos del RTL-SDR a Simulink .....	15
Figura 1.12 Guardar archivos RTL-SDR en Simulink .....	16
Figura 2.1 Diagrama de Flujo de la aplicación.....	19
Figura 2.2 Paquete de instalación “Communications Toolbox Support Package for RTL-SDR Radio” .....	20
Figura 2.3 Instalación del paquete completada .....	20
Figura 2.4 Asistente de configuración para la conexión del RTL-SDR .....	21
Figura 2.5 Conexión exitosa entre el RTL-SDR y MATLAB .....	21
Figura 2.6 Uso de la función <i>sdrinfo</i> para validar reconocimiento del RTL-SDR .....	22
Figura 2.7 Diagrama de flujo para adquisición de datos del espectro FM .....	22
Figura 2.8 Parámetros de la función <i>rtlsdr_rx_singlesweep.m</i> .....	23
Figura 2.9 Hoja de cálculo para almacenar datos obtenidos con 512 muestras por iteración .....	24
Figura 2.10 Diagrama de flujo de parámetros estadísticos de la frecuencia central.....	25
Figura 2.11 Diagrama de Flujo de valores estadísticos para ancho de banda .....	26
Figura 2.12 Diagrama de Flujo de valores estadísticos para intensidad de campo.....	27
Figura 2.13 Código para la obtención de las Frecuencias Centrales .....	28
Figura 2.14 Código para la obtención de valores estadísticos de las Frecuencias Centrales .....	28
Figura 2.15 Código para la obtención de los Anchos de Banda .....	28
Figura 2.16 Código para la obtención de valores estadísticos de los Anchos de Banda .....	29

Figura 2.17 Código para la obtención de las Intensidades de Campo .....	29
Figura 2.18 Código para la obtención de valores estadísticos de las Intensidades de Campo.....	29
Figura 2.19 Código para generación de reporte según el formato establecido .....	31
Figura 2.20 Reporte de la emisora medida .....	31
Figura 2.21 Análisis para Frecuencia Central.....	32
Figura 2.22 Análisis para Ancho de Banda.....	32
Figura 2.23 Análisis para Intensidad de Campo .....	32
Figura 2.24 Código para descartar datos aberrantes .....	33
Figura 2.25 Ventana Principal de la aplicación.....	35
Figura 2.26 Segunda Ventana de la aplicación .....	35
Figura 2.27 Tercer Ventana de la aplicación .....	36
Figura 2.28 Ventana generación Reporte Final de la aplicación .....	36
Figura 3.1 Distribución para la frecuencia central.....	43
Figura 3.2 Distribución para el ancho de banda .....	43
Figura 3.3 Distribución para la intensidad de campo.....	44
Figura 3.4 Reporte Final.....	45
Figura 3.5 Porcentaje de cumplimiento del parámetro frecuencia central .....	46
Figura 3.6 Porcentaje de cumplimiento del parámetro ancho de banda.....	46
Figura 3.7 Porcentaje de cumplimiento para la intensidad de campo .....	47
Figura 3.8 Porcentaje de cumplimiento de los parámetros técnicos.....	47
Figura 3.9 Coeficiente de correlación entre los valores de potencia del RTL-SDR y el analizador espectral .....	48

## ÍNDICE DE TABLAS

Tabla 1.1 Clasificación de Estaciones de radiodifusión.....	7
Tabla 1.2 Emisoras Fm de Guayaquil .....	9
Tabla 1.3 Nivel de modulación por subportadoras .....	10
Tabla 1.4 Intensidad de campo eléctrico. ....	11
Tabla 1.5 Funciones más comunes de la librería del RTL-SDR .....	13

# CAPÍTULO 1

## 1. INTRODUCCIÓN

El presente proyecto estuvo enfocado al diseño de un algoritmo capaz de medir y caracterizar el espectro radioeléctrico en la banda FM, utilizando un dispositivo de bajo costo, RTL SDR, el mismo que permitió hacer un estudio de los parámetros técnicos de las estaciones de radio FM, determinar si los factores de potencia y espaciamiento entre canales cumplen las especificaciones técnicas que controla la ARCOTEL.

En el primer capítulo se detalló la descripción del problema, su justificación y otros conceptos claves para poder entender cómo se desarrolló el proyecto. Se describió la evolución del sistema SDR, conceptos acerca de la radiodifusión FM, así como las emisoras FM en la ciudad de Guayaquil.

En el segundo capítulo se detalló las estrategias y los métodos a realizar para llegar a la solución del proyecto, entre estos se describieron el barrido de frecuencia para el espectro FM completo, la obtención de parámetros técnicos para el barrido completo, el barrido de frecuencia por canal, la interfaz gráfica, la generación del reporte final con los detalles de los parámetros de cada emisora.

En el tercer capítulo se detalló los resultados del proyecto y se comparó cada una de las emisoras FM con la norma técnica, para determinar si la incumplen; para esto se realizó un análisis estadístico a través de prueba de hipótesis con un nivel de confianza del 95% tomando como referencia una emisora al inicio del espectro FM, otra en la mitad y una al final.

En el último capítulo se determinaron las conclusiones que se llegaron una vez que se obtuvo el reporte final con el análisis detallado de cada emisora, de esta forma se logró identificar cuales emisoras presentaron incumplimiento en alguno de sus parámetros. Finalmente se mencionó las recomendaciones para una mejora en el desempeño del dispositivo utilizado en el desarrollo del diseño del proyecto.

## 1.1 Descripción del problema

Desde sus inicios, la radiodifusión FM ha tenido mayor aceptación por el público dada la mejora en la calidad de la señal a diferencia de su predecesora, la radio AM. Los parámetros de calidad se basan en la norma técnica controlada por la ARCOTEL, organismo encargado de medir y caracterizar, entre otros, el espectro del rango de frecuencias FM para validar el cumplimiento de calidad.

Una vez que la ARCOTEL obtiene los datos estos se procesan y se recolectan en alguna hoja de cálculo. Sin embargo, este procesamiento toma mucho tiempo ya que se realiza de forma manual, ingresándolos uno por uno y luego clasificándolos si cumplen o no la norma. Los equipos que utilizan para las mediciones, son analizadores espectrales sofisticados cuyo costo es elevado lo que limita la cantidad de equipos que se puedan adquirir.

El propósito de cumplir con las normas técnicas no recae únicamente en que las emisoras sean sancionadas o no, este problema va más allá, se puede producir interferencia a emisoras adyacentes, los niveles de potencia fuera del valor permitido recaen en un gasto innecesario de energía, o incluso afecta a la salud de la población por los altos niveles de radiación.

## 1.2 Justificación del problema

El GICOM, grupo de investigación de la ESPOL, está interesado en una herramienta que sea capaz de medir el espectro FM, debe ser automatizada, debe controlar el tiempo que se realizan las mediciones e identificar si el espectro asignado a las emisoras de radio FM existen parámetros técnicos que no cumplan las normas de la ARCOTEL.

Con un algoritmo de medición y caracterización se gana tiempo porque los reportes que se generen serán automatizados, dando la posibilidad de poder identificar de manera más eficaz que emisoras son las que cumplen las normas técnicas.

El dispositivo RTL-SDR es de bajo coste y de fácil adquisición a diferencia de los equipos de medición del espectro habituales como el AGILENT E4404B cuyo costo es elevado.

La banda de operación del RTL-SDR es menor que un equipo más costoso, pero para la aplicación de la caracterización del espectro en la banda FM es muy útil.

Con la caracterización del espectro se verifica que se cumpla la norma técnica y con esto corregir las falencias que existan en las emisoras de radio para evitar daños por el incremento en el nivel de radiación, además de ser un beneficio social al cumplir la norma les puede ahorrar en gastos de energía y a evitar pagar las multas por producir interferencia.

Además, con la caracterización se logra detectar que emisoras verdaderamente están cumpliendo la norma técnica, ver si a determinados instantes de tiempo presenta algún comportamiento irregular o si presenta alguna deficiencia en la potencia y está interfiriendo a un canal adyacente.

## **1.3 Objetivos**

### **1.3.1 Objetivo General**

Diseñar algoritmo para medir y caracterizar el espectro radioeléctrico usando un RTL-SDR para la banda FM

### **1.3.2 Objetivos Específicos**

1. Comprobar el correcto funcionamiento del RTL-SDR en base a una frecuencia centrada en 93.7 FM.
2. Realizar pruebas de medición utilizando el RTL-SDR.
3. Diseñar algoritmo para la captura almacenamiento de datos.
4. Medir el espectro FM, recolectar datos de medición y almacenarlos en una base datos.
5. Diseñar un algoritmo para procesamiento de los datos obtenidos del espectro mediante el programa MATLAB.
6. Diseñar un algoritmo para determinar que parámetros influyen en el incumplimiento de la norma técnica de una emisora.
7. Emitir un reporte con parámetros verificando que cumplen o incumplen las normas técnicas que controla la ARCOTEL para cada una de las emisoras de radiodifusión FM.

#### 1.4 Estado del arte

Según [1] se realizó la caracterización para la banda FM utilizando un analizador espectral modelo AGILENT E4404B y usando una antena receptora en la FIEC, con lo que lograron hacer modelamiento del espectro FM y comprobar si el uso del espectro FM es eficiente en ESPOL.

[2] propone la implementación de un sistema en FM el cual permita simular el funcionamiento de una radio definida por software, mediante varios experimentos donde el primero consiste en realizar un modulador FM en una FPGA y su simulación con bloques en Simulink y Quartus II.

El segundo experimento consiste en el diseño del demodulador FM basado en una PLL y su simulación en bloques usando Quartus II, donde finalmente se logra recrear el sistema planteado para el estudio de la traslación de frecuencias y la demodulación de señales FM.

[3] propone la evaluación del espectro para la banda UHF y VHF para parroquias rurales en el cantón Loja para determinar el nivel de ocupación del espectro y si es factible el despliegue de un sistema de radio cognitiva, el cual permita optimizar el espectro asignado.

El presente proyecto se diferencia de los mencionados, ya que se utiliza el RTL-SDR como dispositivo para tomar las mediciones, el cual es de bajo costo. Se realizará un análisis por canales de toda la banda FM.

## 1.5 Marco Teórico

### 1.5.1 RTL-SDR

Es un dispositivo de bajo costo que permite recibir señales de radiofrecuencia del aire las convierte a banda base y las digitaliza; cuenta con una entrada USB y una antena omnidireccional, como se muestra en las figuras 1.1 y 1.2. Por lo general opera entre los 25MHz hasta 1.75Ghz. Dependiendo del modelo se pueden capturar señales de bandas de frecuencias diferente. Para el presente proyecto se usará uno cuyo rango de frecuencias abarque la banda FM. [4]



Figura 1.1 RTL-SDR y antena [5]



Figura 1.2 RTL-SDR sin carcasa [6]

## 1.5.2 Emisoras de Radiodifusión

Las emisoras de radiodifusión se clasifican en:

- Las Estaciones Públicas
- Las estaciones privadas o comerciales.

**Las estaciones Públicas** son aquellas que son destinadas al servicio de la comunidad, sin fines de lucro y por este motivo no pueden utilizar publicidad comercial de ningún tipo. Los contenidos que emiten son de carácter educativo, agrícola, hogar, económico, desarrollo social, servicio a la comunidad; es decir incentivan al desarrollo de la cultura y de información que sea útil para la comunidad. [7]

**Las estaciones Privadas o Comerciales** son aquellas que cuentan con capital privado y son con fines de lucro, su clasificación se muestra en la Tabla 1.1.

**Tabla 1.1 Clasificación de Estaciones de radiodifusión**

<b>Estaciones de radiodifusión o Televisión Libre Terrestre:</b>
Estaciones para difusión unilateral de audio, video o datos. Utilizan ondas cercanas a la superficie de la tierra y son destinadas al público en general.
<b>Estaciones de Radiodifusión o Televisión Codificadas por audio, video y datos</b>
Son estaciones de comunicación unilateral de audio, video y datos, difundidas a través de ondas electromagnéticas. Utilizando códigos que permiten su recepción no esté dirigida al público en general
<b>Estaciones de Radiodifusión o Televisión por satélite de audio, video y datos</b>
Son estaciones que permiten la difusión unilateral de audio, video y datos desde satélites artificiales de la Tierra. Pueden ser destinadas al público en general por captación directa como codificadas para no ser receptadas por el público general.
<b>Estaciones de Radiodifusión o Televisión de circuito cerrado</b>
Son estaciones que permiten la difusión unilateral de audio, video y datos dentro de locales cerrados o para predios perfectamente delimitados. Las técnicas a utilizar serán por medios físicos o emisiones de características especiales.
<b>Otras estaciones de radiodifusión o Televisión Especiales</b>
son aquellas que pueden emitir de forma unilateral a un público predeterminado programas de música ambiental y servicios en general como: ayuda en la dirección de tránsito vehicular, búsqueda de personas o cosas, transmisión de mensajes, o auxilios.

### **1.5.2.1 Estación Matriz**

Se denomina al conjunto del transmisor, control maestro y todos los dispositivos adicionales necesarios para que una emisora pueda funcionar en el área de cobertura que está autorizada a transmitir. [8]

### **1.5.2.2 Estación repetidora**

Recepta toda programación de una estación matriz y la retransmite a la zona de cobertura que se le ha autorizado.

### **1.5.2.3 Estaciones Locales**

Son las estaciones de radiodifusión utilizadas para cubrir sectores de baja densidad poblacional, donde su frecuencia puede ser reutilizada por otra estación siempre y cuando pertenezcan a la misma zona.

## **1.5.3 Cobertura**

Se refiere al espacio geográfico donde es posible recibir una mínima cantidad de señal de radiofrecuencia, se clasifica en:

- área de cobertura principal.
- área de cobertura de protección o secundaria.

### **1.5.3.1 Área de cobertura principal**

Es el área que comprende el entorno urbano, como ciudades o poblaciones aledañas con infraestructuras, el valor de intensidad asignado para esta área esta depende si es monofónico o estereofónico. Para la toma de datos del RTL-SDR se asume estereofónico por lo tanto el valor de la intensidad para esta área debe ser mayor o igual a 54 dB $\mu$  V/m.

### **1.5.3.2 Área de cobertura secundaria**

Es el área que corresponde a los límites de las ciudades, lugares donde se debe esperar tener una recepción menor que la de una ciudad, para el caso estereofónico este valor oscila entre valores mayores a 50 dB $\mu$  V/m y menores a 54 dB $\mu$  V/m.

#### 1.5.4 Interferencia de canal adyacente

Esta interferencia se produce cuando dos subportadoras se encuentran muy cerca de la otra, produciendo un incremento en la amplitud y por ende un desperdicio de potencia. Para la radiodifusión FM existe una normativa técnica controlada por la ARCOTEL, la cual establece un ancho de banda de guarda de 200 kHz. Las emisoras de radiodifusión para la ciudad de Guayaquil se muestran a continuación en la Tabla 1.2.

**Tabla 1.2 Emisoras Fm de Guayaquil [9]**

<b>Frecuencia [MHz]</b>	<b>Nombre de la emisora</b>	<b>Frecuencia [MHz]</b>	<b>Nombre de la emisora</b>	<b>Frecuencia [MHz]</b>	<b>Nombre de la emisora</b>
88.1	Radio María	94.9	La otra FM	101.7	Radio Hit
88.5	Galaxia FM	95.3	Cupido FM	102.1	WQ Radio
88.9	Radio Diblu	95.7	Radio Metro Stereo	102.5	HCJB
89.3	Radio City	96.1	Onda Cero	102.9	Radio Sucre
89.7	Punto Rojo	96.5	Tropicana	103.3	Joya Stereo
90.1	Romance	96.9	Mas Candela	103.7	Sonorama
90.5	Canela	97.3	Nuevo Tiempo	104.1	Alfa Radio
91.3	Tropicálida	98.1	Radio Morena	104.5	La Tuya
91.7	Antena 3	98.5	JC Radio	104.9	Radio 11Q
92.1	Estrella	98.9	Radio i99	105.3	Radio Pública EC
92.5	Forever	99.3	Sabormix	105.7	Radio Fabu
92.9	Armónica	99.7	Radio Elite	106.1	BBN Radio
93.3	Majestad	100.1	La Prensa Sport	106.5	Radio Fuego
93.7	Radio Disney	100.5	RSN FM	106.9	R. Francisco Stereo
94.1	Onda Positiva	100.9	Radio Asamblea	107.3	Radio Rumba
94.5	Platinum FM	101.1	Radio Centro	107.7	Radio Visión

### **1.5.5 Norma Técnica ARCOTEL Banda FM**

La norma técnica de la ARCOTEL contempla 100 frecuencia cada un espaciamiento de 0.a con una separación de 200 kHz, el espectro FM comienza en 88MHz y termina en 108 MHz. Para frecuencias adyacentes deberá haber un espaciamiento de por lo menos 0.4 MHz emisora. [8]

#### **1.5.5.1 Ancho de Banda**

Se establece en 180 kHz para monofónico y en 220 kHz para estereofónico, con una tolerancia del 5%

#### **1.5.5.2 Porcentaje de Modulación**

Para sistemas monofónicos o estereofónicos no debe exceder el 100%, además el porcentaje de modulación depende de la cantidad de subportadoras como se muestra en la Tabla 1.3.

**Tabla 1.3 Nivel de modulación por subportadoras**

una subportadora	$\leq 95\%$
Dos o más subportadoras	$\leq 100\%$

#### **1.5.5.3 Tolerancia de Frecuencia**

La variación máxima permitida en una frecuencia central es de  $\pm 2$  kHz

#### **1.5.5.4 Niveles de Emisión no esenciales**

Se debe atenuar con un nivel de potencia mínimo de 80 dB por debajo del nivel medio del ancho de banda autorizado y con un porcentaje de modulación del 100%

#### **1.5.5.5 Intensidad de campo Mínima a proteger**

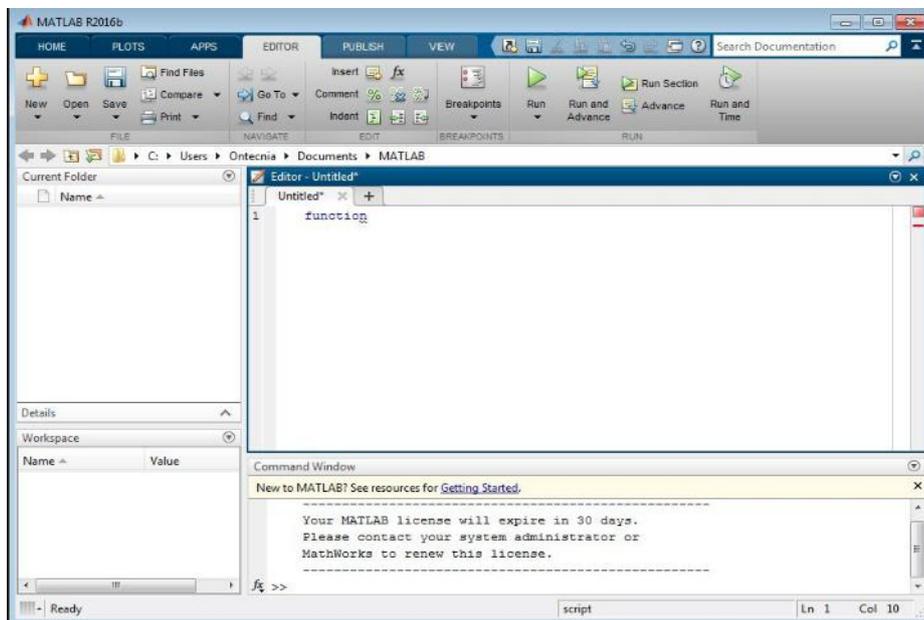
Los niveles de intensidad de campo para estaciones de frecuencia normal y locales, se clasifican por su cobertura principal y secundaria, dichos valores se muestran a continuación

**Tabla 1.4 Intensidad de campo eléctrico.**

En el borde del área de cobertura principal	
<b>Monofónicos</b>	<b>Estereofónicos</b>
$\geq 48 \text{ dB}\mu\text{V/m}$	$\geq 54 \text{ dB}\mu\text{V/m}$
En el borde del área de cobertura secundaria	
<b>Monofónicos</b>	<b>Estereofónicos</b>
$\geq 30 \text{ dB}\mu\text{V/m}$ y $< 48 \text{ dB}\mu\text{V/m}$	$\geq 50 \text{ dB}\mu\text{V/m}$ y $< 54 \text{ dB}\mu\text{V/m}$

### 1.5.6 MATLAB

Es una herramienta de software matemática que cuenta con un lenguaje de programación 'amigable', permite trabajar con matrices, arreglos y funciones matemáticas. Además, permite manipular diferentes variables, graficarlas para comprobar el funcionamiento de algún sistema que se esté simulando. Es muy utilizado en el ámbito profesional para diseño de procesos y despliegue de operaciones empresariales. [10]



**Figura 1.3 Ventana de Matlab**

La ventana de MATLAB se clasifica en:

**Current Folder (Carpeta Actual):** Muestra todos los archivos disponibles en el directorio actual utilizado para la ejecución del código. Permite la manipulación, creación o eliminación de los archivos.

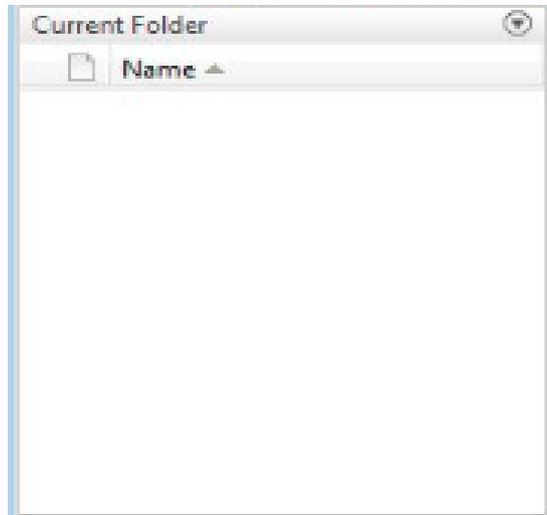


Figura 1.4 Ventana Current Folder

**Command Window (Ventana de comandos):** Es el lugar donde se ingresan todos los comandos de MATLAB.

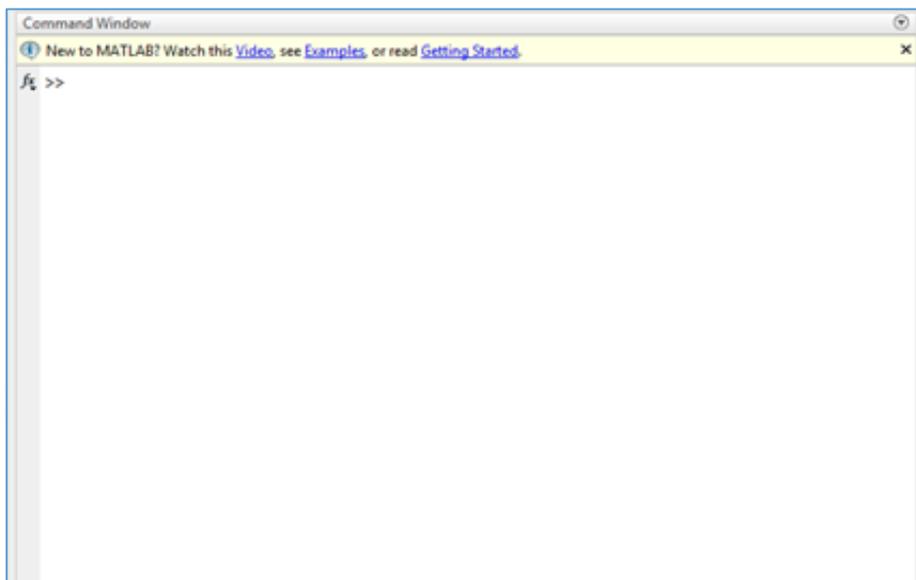


Figura 1.5 Command Window Matlab

**Workspace:** Es el lugar donde se almacenan todas las variables y se observan todos los cambios realizados en las mismas.

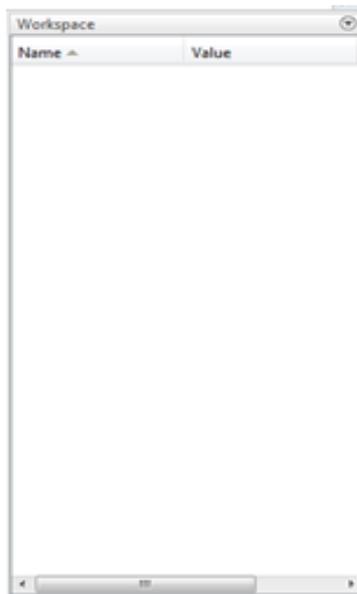


Figura 1.6 Ventana Workspace Matlab

### 1.5.7 Librerías del RTL-SDR para Matlab

Con estas librerías es posible dar tratamiento a las señales que captura el RTL-SDR, para ello se necesita de una versión de MATLAB 2017b o posterior y un sistema operativo adecuado (Windows 7 o posterior, algunas versiones de Linux). Algunas de las funciones que se pueden encontrar son:

**Tabla 1.5 Funciones más comunes de la librería del RTL-SDR [11]**

<b>comm.SDRRTLReciver()</b>	Crea un receptor RTL-SDR
<b>dsp.FIRFilter()</b>	Crea un filtro FIR
<b>dsp.FIRDecimator()</b>	Crea un Decimator FIR
<b>dsp.SpectrumAnalyzer()</b>	Crea un Analizador Espectral
<b>dsp.AudioPlayer()</b>	Crea un Audio Player
<b>Sdrinfo</b>	Muestra toda la información relacionada al RTL-SDR

### 1.5.8 Simulink

Es un entorno grafico que permite trabajar con bloques los cuales simulan la codificación por código en MATLAB, muy útil a la hora de simular procesos y facilita a algunos usuarios la manipulación de las variables. Simulink posee dos ventanas: la ventana del buscador de librerías y la ventana de Modelamiento. La ventana buscadora de librerías contiene todas las librerías que se pueden utilizar desde funciones matemáticas a funciones tales como Fourier entre otras, para la manipulación de señales del espectro.

La ventana de modelamiento se usa para simular los procesos creados a través de los bloques de las librerías. [13]

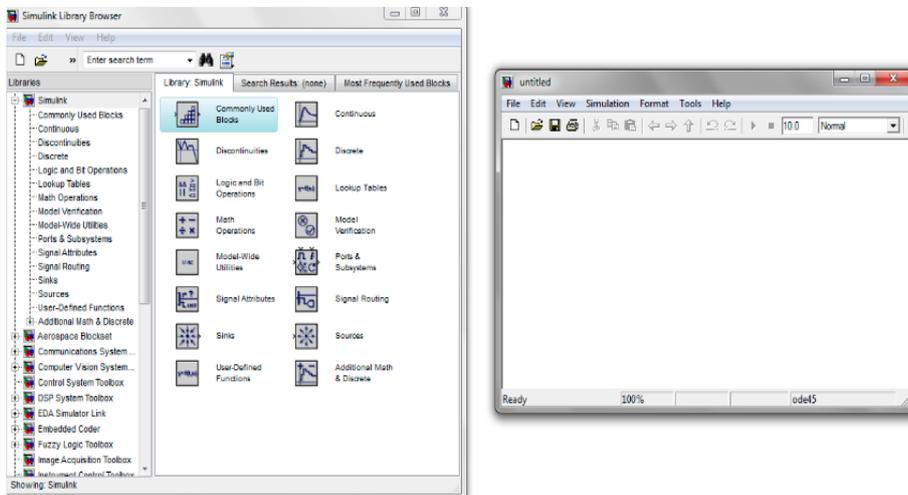


Figura 1.7 Buscador de librerías y entorno de simulación simulink

### 1.6 Técnicas de muestreo

**Upsampling** Consiste en incrementar el tiempo de muestreo al insertar ceros entre las muestras originales en tiempo discreto.

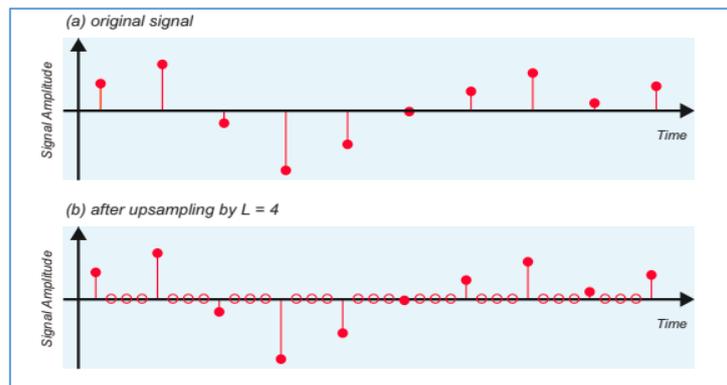


Figura 1.8 Upsampling [14]

**DownSampling** Consiste en descartar muestras intermedias basándose en un parámetro M con la finalidad de reducir la tasa de muestreo.

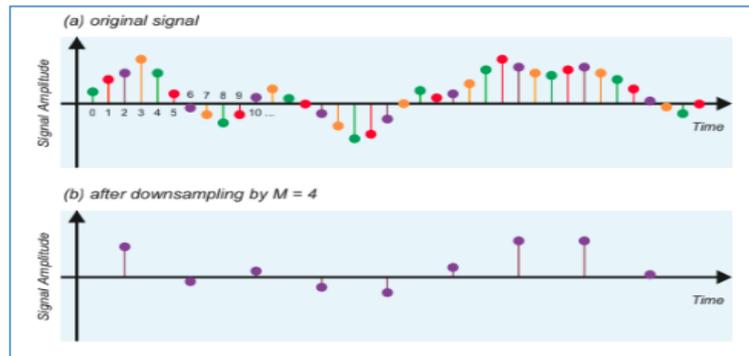


Figura 1.9 Downsampling [15]

## 1.7 Guardar e importar datos

### 1.7.1 Guardar e Importar Datos del RTL-SDR

Para importar datos del RTL-SDR a MATLAB se utiliza el objeto `import_rtlsdr_data`.

```
% create an 'import RTL-SDR data' object
obj_rtlsdr = import_rtlsdr_data(...
    'filepath', 'rec_data/am_dsb_tc.mat',...
    'frm_size', 4096,...
    'data_type', 'single');
```

Figura 1.10 Importar datos del RTL-SDR a MATLAB [16]

Para importar datos del RTL-SDR a través de Simulink se va a la ventana de librerías y se selecciona Additional Tools

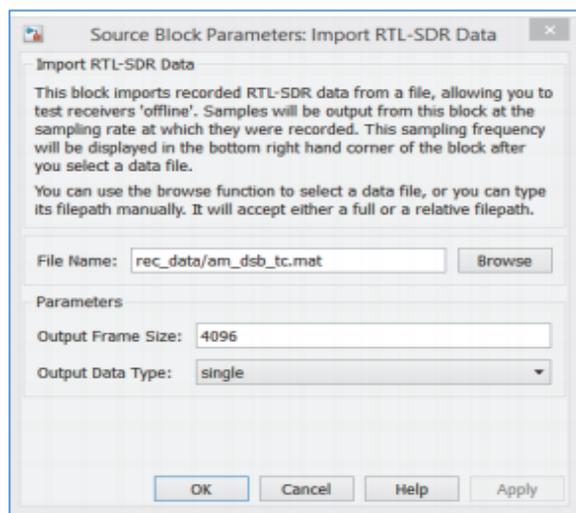


Figura 1.11 Importar datos del RTL-SDR a Simulink [16]

Para guardar los archivos del RTL-SDR en Simulink se va a RTL-SDR Book Library luego a Additional Tools y encontrar el bloque Save RTL-SDR Data Block

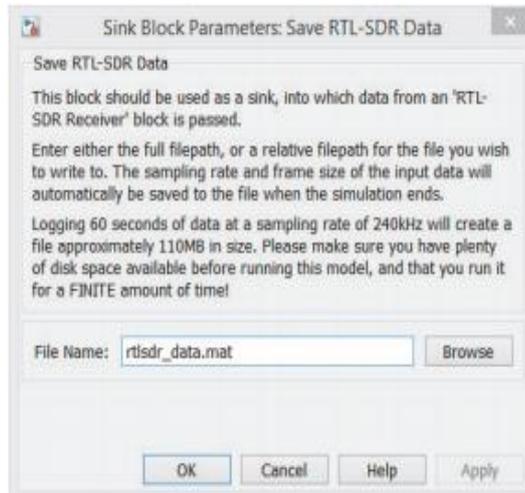


Figura 1.12 Guardar archivos RTL-SDR en Simulink [17]

### 1.7.2 Algunas ecuaciones a utilizar

$$P(W) = \frac{\text{Voltaje}(V)^2}{\text{Resistencia}(\Omega)} = \frac{\text{Voltaje}}{\text{corriente}} \quad (1.1)$$

Donde P es la potencia en Watts, Voltaje en unidades de voltios y resistencia en unidades de ohmios, la corriente en unidades de amperios

$$P(dB) = 10 \log\left(\frac{p}{1W}\right) \quad (1.2)$$

Fórmula para transformar la potencia en Watts a dB

$$P(dBm) = 10 \log\left(\frac{P}{1mw}\right) = 10 \log\left(\frac{P}{10^{-3}}\right) \quad (1.3)$$

Fórmula para transformar la potencia en mW a dBm [18]

$$P_r(dBm) = E(dB\mu) - 20 \log f(Mhz) - 77.2 + G_r \quad (1.4)$$

Fórmula para la obtención de la intensidad de campo eléctrico, donde  $P_r$  es la potencia en dBm, f la frecuencia en MHz, y  $G_r$  en este caso es la ganancia interna asignada del RTL-SDR

$$E(dB\mu) = P_r(dBm) + 20 \log f(Mhz) + 77.2 - G_r - G_{RTL} \quad (1.5)$$

Esta fórmula utiliza la ecuación 1.4 como formato y adicionalmente se le resta la ganancia del RTL.

### **1.7.3 Interfaz gráfica en MATLAB**

Las GUI o interfaces graficas de usuario dan al usuario el control de las aplicaciones del software, lo cual le facilita a realizar la función deseada sin necesidad de conocimiento alguno del entorno de programación de MATLAB. Las GUIs automatizan una tarea o un cálculo, entre los elementos que se pueden añadir a las GUIs están botones, controles deslizantes, menús, controles deslizantes, entre otros. Para la implementación de la interfaz gráfica final para el cliente se usará un grupo de ventanas las cuales contarán con dos controles deslizantes para la asignación de la frecuencia a elegir y otro para el tiempo de medición; además de botones que permiten el acceso entre las múltiples ventanas, la llamada a las funciones para distribución normal y la generación del reporte.

## **1.8 Alcance**

Para el presente proyecto se usará un RTL-SDR debido a que es de bajo costo. Este dispositivo tiene una amplia gama de frecuencias, pero se limitará solo a usar la banda FM y únicamente para las frecuencias de Guayaquil. Para la captura y lectura de los datos se necesita de una versión compatible de MATLAB, superior o igual a la versión 2017b, con las librerías del RTL-SDR. Se utiliza MATLAB ya que cuenta con una gran cantidad de herramientas que permiten trabajar con arreglos de datos y diseñar procesos con un lenguaje de programación no tan complejo. Para el desarrollo de los algoritmos se tomó como base el algoritmo desarrollado por los autores del libro '*MATLAB & Simulink and the SDR*' utilizado para el desarrollo de esta tesis, este algoritmo se puede encontrar en el Anexo C, el mismo que sirvió para el desarrollo de los algoritmos de medición del espectro por canal, lectura y procesamiento de datos para la generación del reporte, análisis de media, mediana y desviación estándar para los parámetros técnicos medidos, la interfaz gráfica para el usuario entre otros.

# CAPÍTULO 2

## 2. METODOLOGÍA

El grupo de investigación de la ESPOL desea un algoritmo, el cual le permita medir y evaluar el estado de cada una de las estaciones de radio, debe ser capaz de generar un reporte al final de la obtención de los datos. Para la obtención de los datos se usará el RTL-SDR que es un dispositivo que se conecta al computador y permite extraer datos como valores de potencia y demás. Para comprobar su correcto funcionamiento se debe realizar un test, el cual consiste en medir una frecuencia utilizando un analizador espectral, observar los niveles de potencia y sintonizarla. Una vez que se comprueba que el equipo funciona de manera correcta, se continúa con la creación del algoritmo en MATLAB, debe mostrar una ventana con un botón que permita medir en base a un tiempo que asigna el usuario, también debe tener la opción para almacenar el reporte, en un principio se planeaba utilizar MYSQL para este propósito, pero dado la gran cantidad de datos y columnas se optó por guardar en una hoja de cálculo.

Cuando se ha confirmado el tiempo que desea censar el espectro radioeléctrico en las frecuencias de las emisoras de radio FM, el RTL-SDR toma las lecturas necesarias y las guarda, para luego obtener un arreglo de datos para cada frecuencia en base al tiempo que se asignó en el algoritmo. El algoritmo será capaz de procesar dicha información, sacar un promedio de los datos que se obtienen y ser capaz de discriminar los datos aberrantes si existieran, así como la toma de decisión sobre si las emisoras cumplen los parámetros que controla la ARCOTEL

## 2.1 Diseño de diagrama de flujo para la aplicación de medición y caracterización del espectro FM

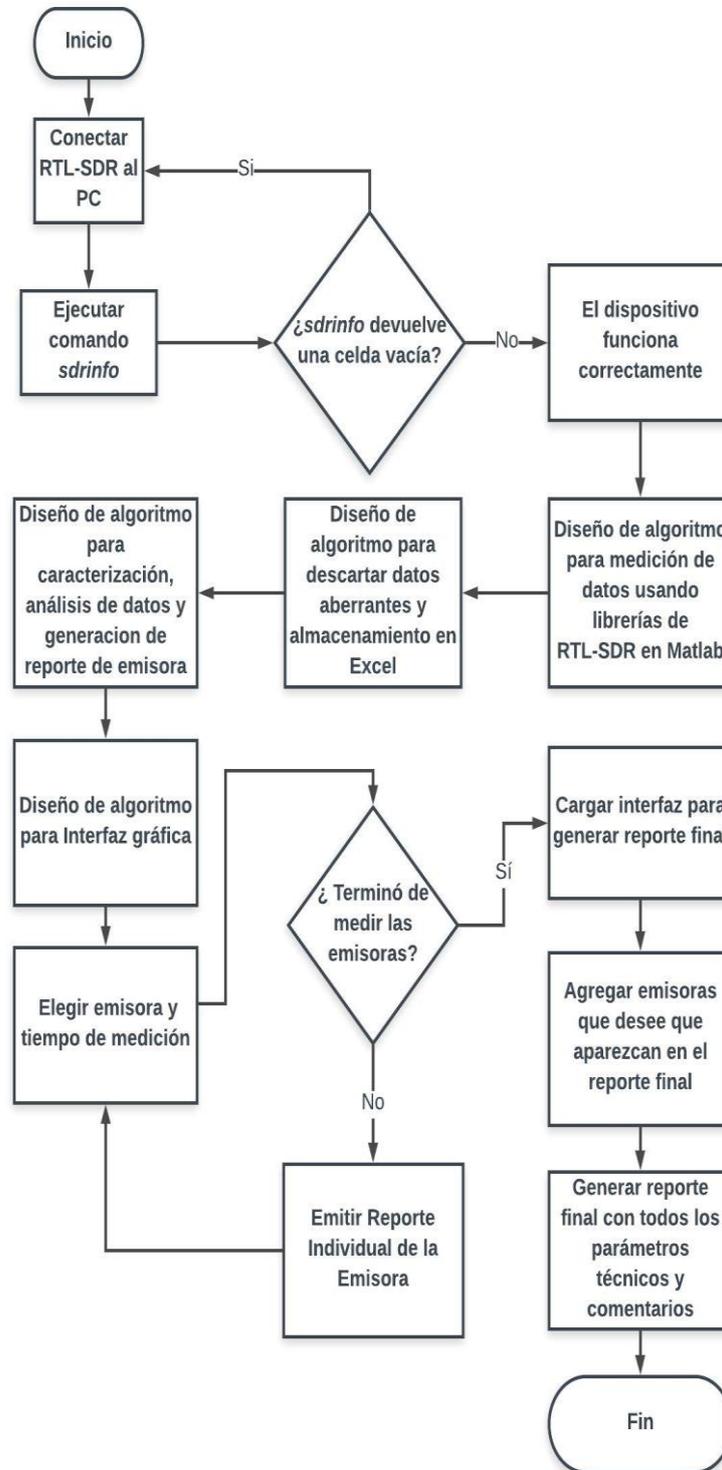


Figura 2.1 Diagrama de Flujo de la aplicación

## 2.2 Funcionamiento del RTL-SDR y conexión con MATLAB

Para realizar una conexión exitosa entre el RTL-SDR con MATLAB es necesario descargar e instalar la librería de tipo “Hardware Support Package” llamada “Communications Toolbox Support Package for RTL-SDR Radio” disponible en la sección “Add-Ons” de MATLAB. (ver Figura 2.2)

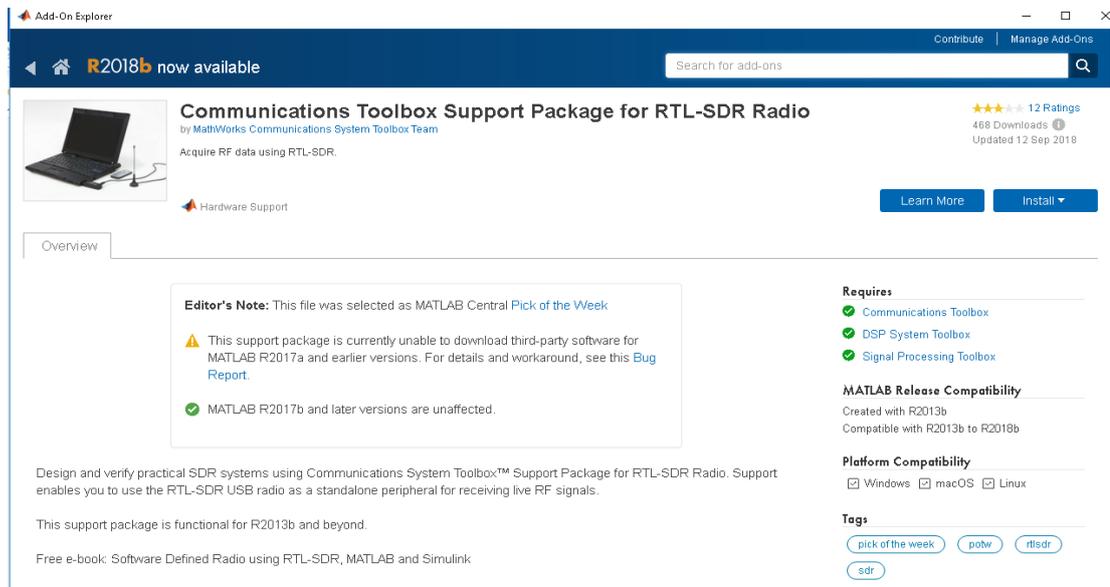


Figura 2.2 Paquete de instalación “Communications Toolbox Support Package for RTL-SDR Radio”

Una vez instalado este paquete (Figura 2.3) se procede a configurar la conexión con el RTL-SDR siguiendo paso a paso las indicaciones del asistente de configuración para la instalación de los respectivos drivers (Figura 2.4) y validar una conexión exitosa (Figura 2.5).

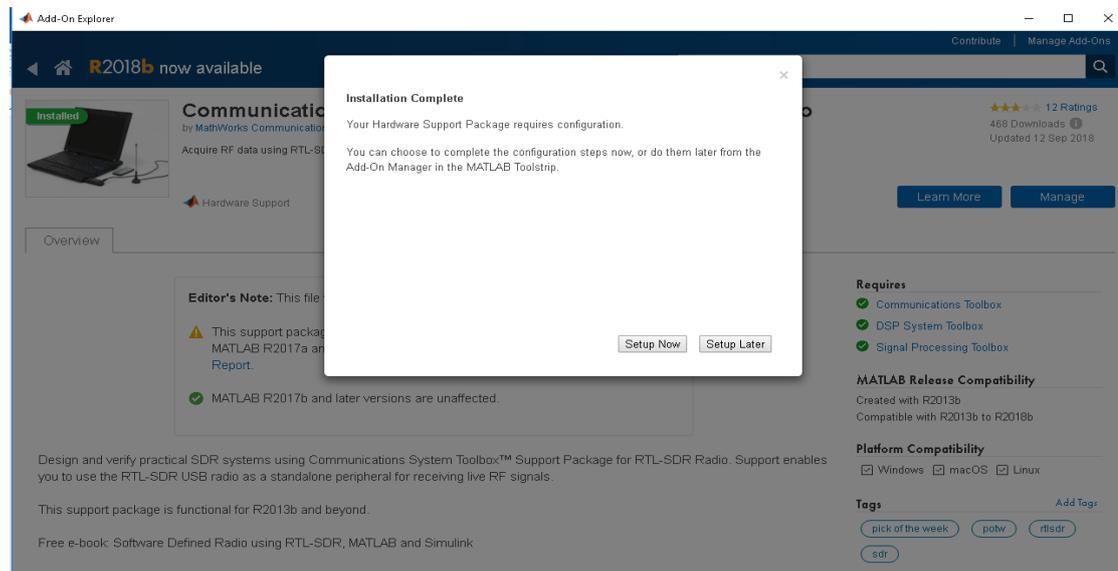


Figura 2.3 Instalación del paquete completada

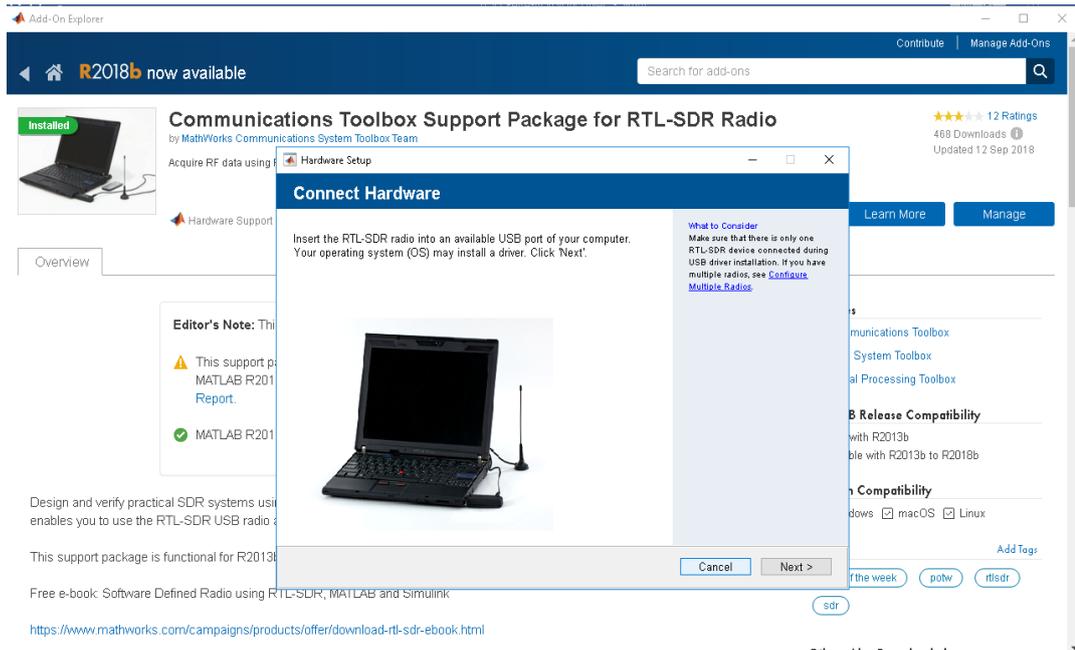


Figura 2.4 Asistente de configuración para la conexión del RTL-SDR

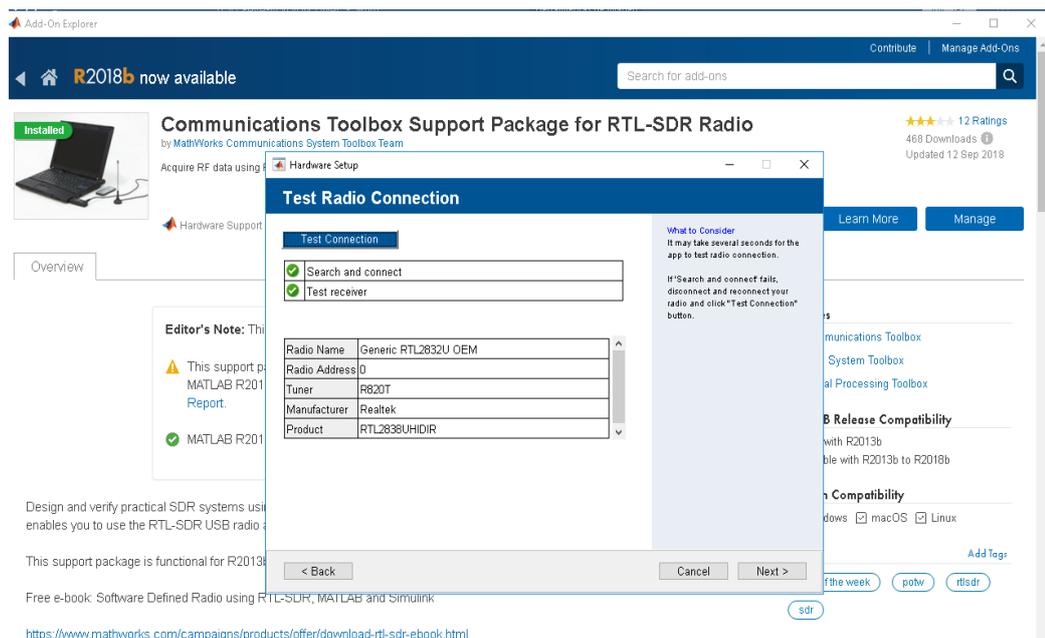


Figura 2.5 Conexión exitosa entre el RTL-SDR y MATLAB

Una vez obtenida la respuesta exitosa en la prueba de conexión por parte del asistente de configuración ya se puede hacer uso del RTL-SDR desde MATLAB mediante las funciones descritas en la tabla 1.5. Como por ejemplo la función *sdrinfo* para verificar que el RTL-SDR está conectado a la PC y MATLAB lo reconoce (Figura 2.6).

```

Command Window
>> sdrinfo
ans =
    RadioName: 'Generic RTL2832U OEM'
    RadioAddress: '0'
    RadioIsOpen: 0
    TunerName: 'R820T'
    Manufacturer: 'Realtek'
    Product: 'RTL2838UHDIR'
    GainValues: [29x1 double]
    RTLCrystalFrequency: 28800000
    TunerCrystalFrequency: 28800000
    SamplingMode: 'Quadrature'
    OffsetTuning: 'Disabled'
ft >>

```

Figura 2.6 Uso de la función *sdrinfo* para validar reconocimiento del RTL-SDR

### 2.3 Diseño de algoritmo para la adquisición y almacenamiento de datos del espectro FM.

El diagrama de flujo que se muestra a continuación en la Figura 2.7 muestra el algoritmo diseñado.

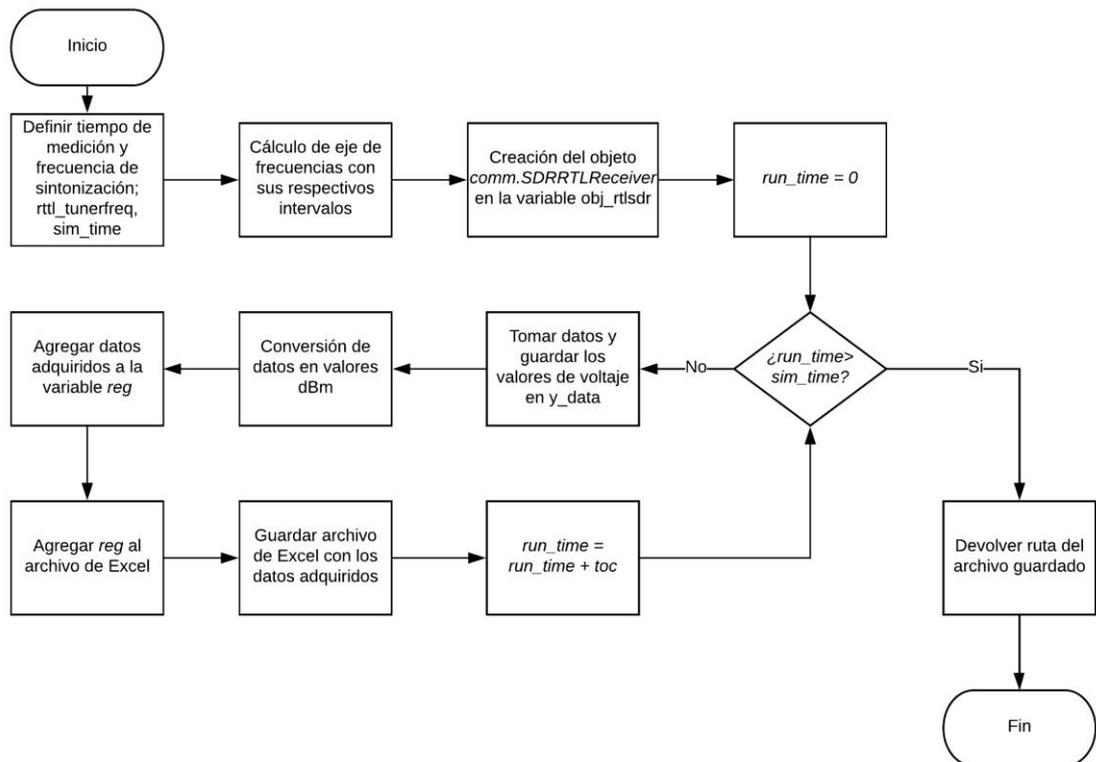


Figura 2.7 Diagrama de flujo para adquisición de datos del espectro FM

Para realizar la adquisición de datos del espectro FM se creó la función *rtlsdr\_rx\_singlesweep.m* la cual toma datos de acuerdo a los parámetros configurables antes de la medición (Figura 2.8) como lo son la frecuencia de muestreo, la ganancia del RTL-SDR, número de muestras a tomar, etc. Con parámetros de entrada de frecuencia de sintonización y tiempo de medición.

```

16
17 function [ruta] = rtlsdr_rx_singlesweep (rtlsdr_tunerfreq, sim_time)
18
19 % PARAMETERS
20 location = 'Guayaquil'; % Lugar; usado para el gráfico mostrado
21 start_freq = 88e6; % Frecuencia de inicio del espectro FM
22 stop_freq = 108e6; % Frecuencia final del espectro FM
23 rtlsdr_id = '0'; % RTL-SDR ID
24 rtlsdr_fs = 0.95e6; % RTL-SDR: Tasa de muestreo en Hz
25 rtlsdr_gain = 0; % RTL-SDR: Ganancia en dB
26 rtlsdr_frmLen = 16384; % RTL-SDR: Tamaño de frames de salida
27 rtlsdr_datatype = 'single'; % RTL-SDR: Tipo de datos de salida
28 rtlsdr_ppm = 0; % RTL-SDR tuner parts per million correction
29 nfrhold = 20; % Numero de muestras a tomar
30 fft_hold = 'avg'; % Función Hold "max" o "avg"
31 nfft = 16384; % Numero de FFTs (2^n)
32 dec_factor = 16; % Downsample de salida
33 overlap = 0.5; % FFT de salida para evitar rolloff
34 nfrmdump = 0; % Numero de muestras a desechar antes de retornar data
35 run_time=0;
36

```

Figura 2.8 Parámetros de la función *rtlsdr\_rx\_singlesweep.m*

Una vez se adquieren los datos, se almacenan en una hoja de cálculo con el nombre de la frecuencia sintonizada, en el archivo se muestra los intervalos de frecuencia del canal respecto a la frecuencia sintonizada dependiendo de la frecuencia de muestro del RTL-SDR, el segundo en el que se realizó la medición y los valores de potencia para cada intervalo de frecuencia (Figura 2.9). El algoritmo calcula estos intervalos de frecuencia y los almacena en un arreglo para usarlo en análisis posteriores, la obtención del arreglo se realiza mediante la siguiente ecuación 2.1 a continuación.

$$\begin{aligned}
 freq\_axis = & (rtlsdr\_tunerfreq - rtlsdr\_fs/2 * overlap: freq\_bin\_width \\
 & * dec\_factor: (rtlsdr\_tunerfreq + rtlsdr\_fs/2 * overlap \\
 & - freq\_bin\_width)/1e6; \qquad \qquad \qquad (2.1)
 \end{aligned}$$

Donde  $freq\_bin\_width = (rtlsdr\_fs/nfft)$ , generando el tamaño del arreglo de frecuencias y los valores de las variables están dados por los parámetros configurados que se mostraron en la Figura 2.8, dando como resultado un arreglo con 512 muestras en las que se obtiene un valor de potencia para cada muestra del intervalo de frecuencias en cada iteración hasta cumplir el tiempo de medición.

	A	B	C	IV	IW	IX	IY	IZ	SQ	SR	SS
	DataTunerFrequencyNotAberrant88										
	Numero	Numero	Numero	Numero	Numero	Numero	Numero	Numero	Numero	Numero	Numero
1	Tiempo	Frecuencia1	Frecuencia2	Frecuencia255	Frecuencia256	Frecuencia257	Frecuencia258	Frecuencia259	Frecuencia510	Frecuencia511	Frecuencia512
2	0	88.2625	88.2634	88.4981	88.4991	88.5000	88.5009	88.5019	88.7347	88.7356	88.7366
3	3.6738	-53.5841	-53.4992	-38.3716	-38.0614	-38.2003	-37.9944	-38.3434	-52.8489	-53.3247	-53.3800
4	4.3592	-53.2315	-53.1509	-36.8696	-37.0479	-36.6402	-36.5522	-37.1241	-53.6279	-53.3842	-53.6217
5	5.0573	-53.1573	-53.4898	-36.8532	-37.3125	-37.1914	-36.9411	-37.2180	-53.3101	-53.1747	-53.2314
6	5.8108	-53.0117	-53.4878	-36.6974	-36.5962	-37.2495	-36.4306	-36.3716	-53.7047	-53.2641	-53.4715
7	6.5105	-53.4359	-52.7171	-36.3081	-36.9169	-36.9372	-37.3308	-37.0909	-53.0289	-53.0453	-53.3120
8	7.2802	-53.2754	-53.1365	-36.7666	-36.6660	-36.6666	-36.8745	-36.8648	-52.6970	-53.4026	-52.9880
9	8.7305	-53.1125	-53.0144	-34.4938	-35.3918	-34.9754	-35.2024	-36.1480	-53.4979	-53.4397	-53.3859

Figura 2.9 Hoja de cálculo para almacenar datos obtenidos con 512 muestras por iteración

Para la captura de datos en cada valor de frecuencia del arreglo obtenido previamente se crea la función *capture\_and\_plot*, para esto se crea un objeto de sistema RTL-SDR llamado *obj\_rtlsdr*, definiéndolo con los parámetros anteriormente establecidos usando la función *comm.SDRRTLReceiver* descrita en la Tabla 1.5. Se modifica la variable *obj\_rtlsdr* dentro de un lazo *while* para obtener los valores de voltaje en cada iteración para cada intervalo de las 512 muestras de frecuencia, estos valores de voltaje que obtiene el RTL-SDR son convertidos a valores de potencia mediante la ecuación 2.2.

$$y\_data\_dbm = 10 * \log_{10}((y\_data.^2/50)) + 30 \text{ [dBm]} \quad (2.2)$$

Donde *y\_data* es el arreglo de datos del espectro obtenidos por el RTL-SDR. Una vez terminada la conversión, estos datos se agregan a la hoja de cálculo y se repite el proceso hasta cumplir el tiempo de medición. La función devuelve como parámetro de salida el nombre de la ruta donde se guardó el archivo generado.

## 2.4 Diseño de algoritmos para la obtención de parámetros para el espectro FM.

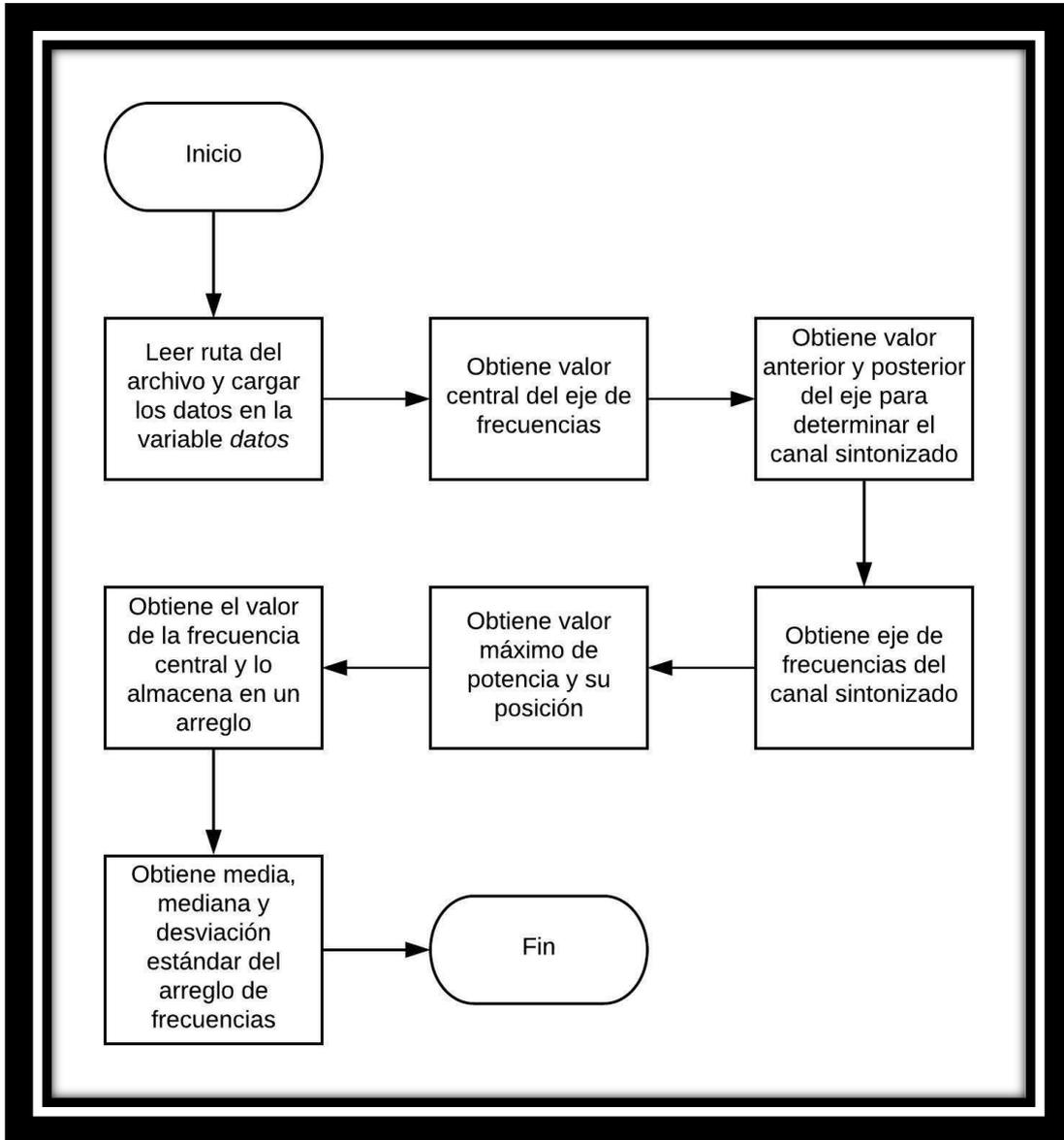


Figura 2.10 Diagrama de flujo de parámetros estadísticos de la frecuencia central

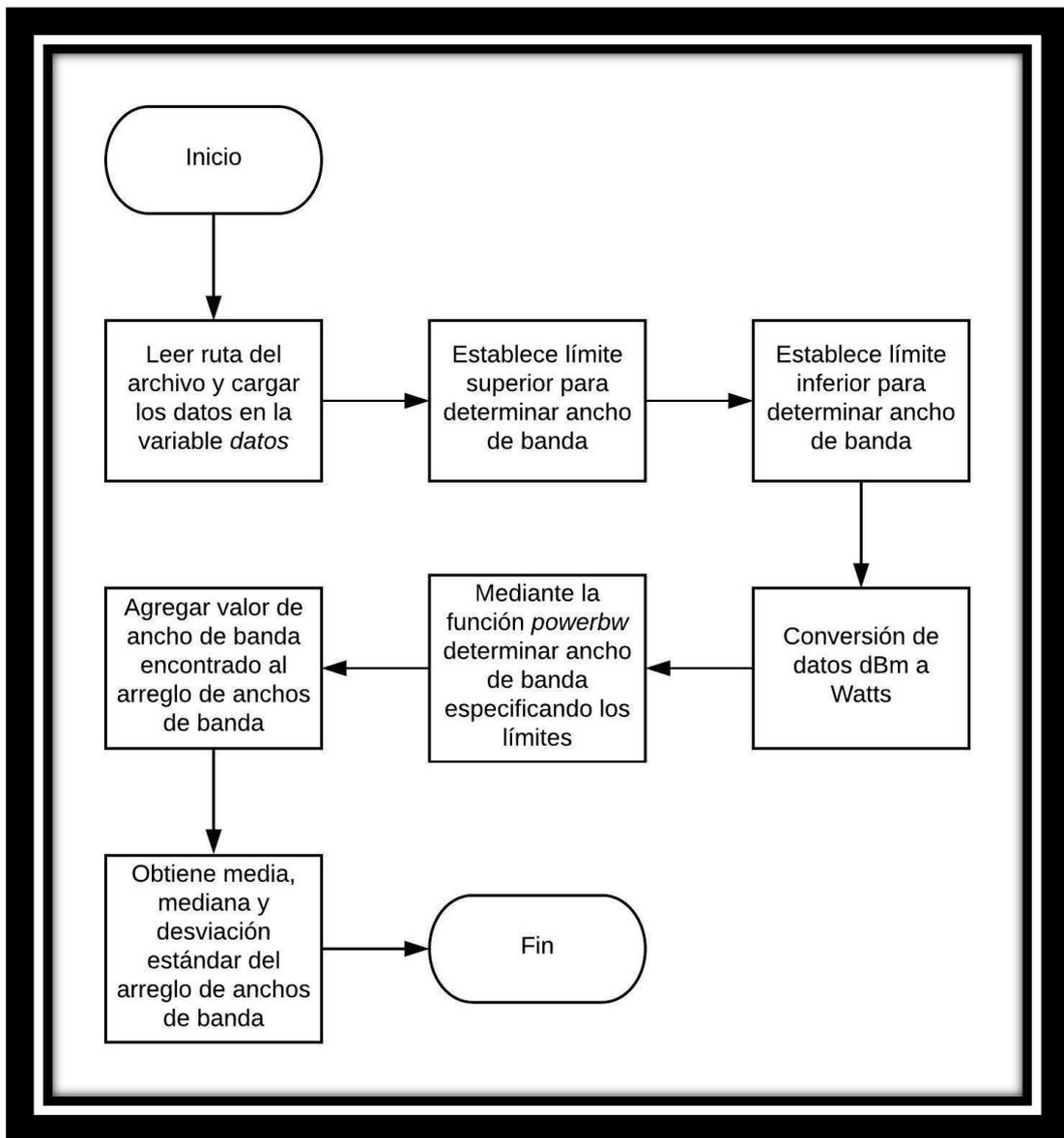


Figura 2.11 Diagrama de Flujo de valores estadísticos para ancho de banda

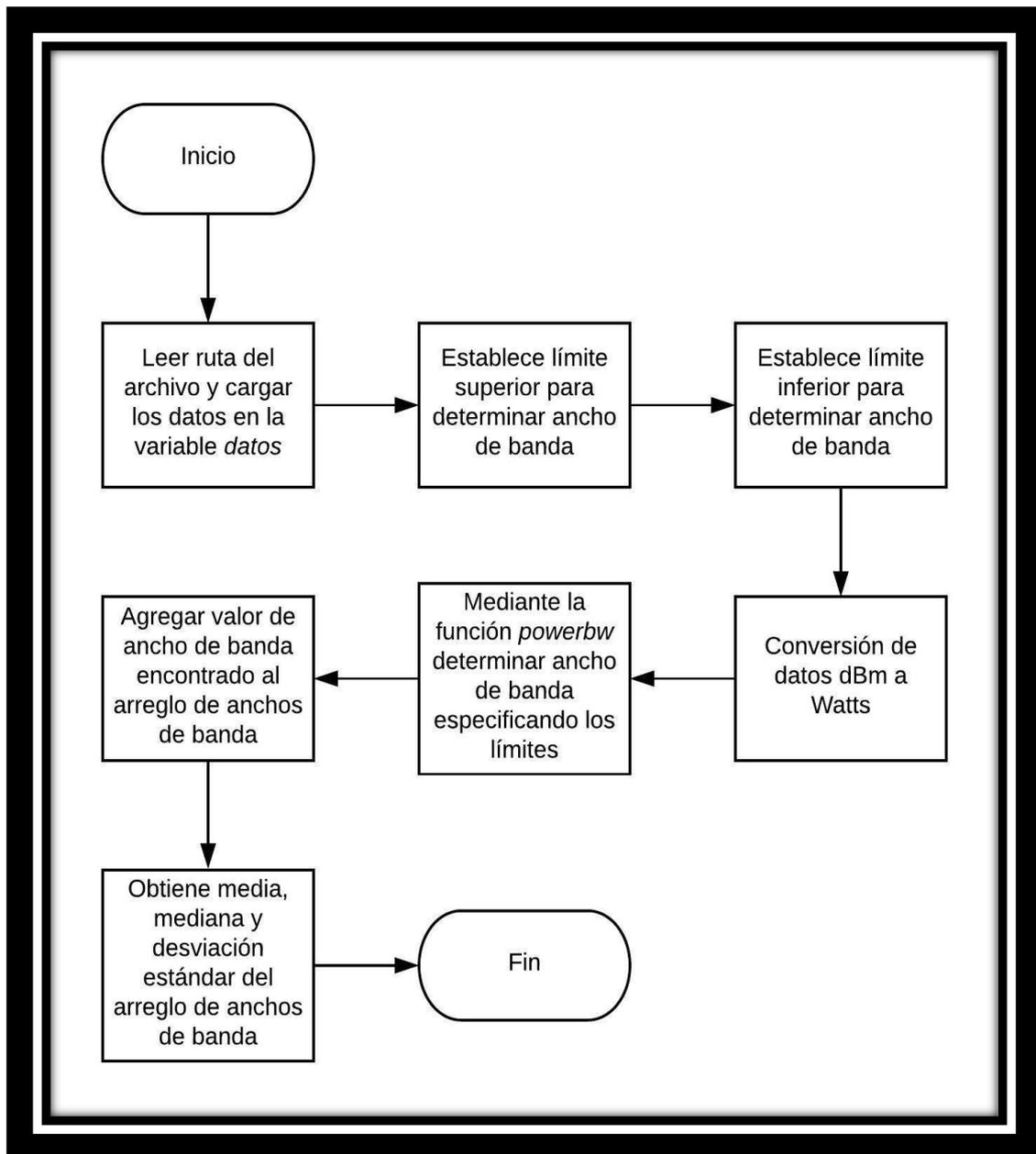


Figura 2.12 Diagrama de Flujo de valores estadísticos para intensidad de campo

Una vez culminada la adquisición y el almacenamiento de los datos del espectro FM es necesario analizar dichos datos para obtener los parámetros de interés. Para esto se crean las funciones *median\_fc.m*, *median\_bw.m* y *median\_ic.m* que indican los valores de “Frecuencia Central”, “Ancho de Banda” e “Intensidad de Campo Eléctrico” respectivamente, como su nombre sugiere, basados en los diagramas de flujo de las Figuras 2.10, 2.11 y 2.12.

En la Figura 2.13 se muestra el código implementado para la obtención de las frecuencias centrales de los datos del archivo cargado en la función.

```

median_fc.m  median_bw.m  median_ic.m  normaldistr.m  reporte_emisora.m  +
1  function [medianfc, meanfc, stdfc, frecuenciascentrales, n] = median_fc(ruta, tuner_freq)
2  datos=xlsread(ruta);
3  freq_axis=(datos(1, (2:end)));
4  frecuenciascentrales=[];
5  s=size(datos);
6  for i=2:1:s(1)
7      pxx_dbm=datos(i, (2:end));
8      pcentral = 256; %Valor central del eje de frecuencias
9      pant = 144; %Valor anterior para determinar el canal sintonizado
10     psig = 370; %Valor posterior para determinar el canal sintonizado
11     sub_arreglo_pot1=(pxx_dbm(1, (pant:pcentral)));
12     sub_arreglo_pot2=(pxx_dbm(1, (pcentral:psig)));
13     value1=min(sub_arreglo_pot1); %Menor valor en el sub_arreglo
14     value2=min(sub_arreglo_pot2); %Mayor valor en el sub_arreglo
15     pos1=find(pxx_dbm==value1); % Posición del menor valor encontrado en datos
16     pos2=find(pxx_dbm==value2); % Posición del mayor valor encontrado en datos
17     freq_axis_emisora=freq_axis(1, (pos1:pos2)); % Eje de frecuencias del canal
18     sub_arreglo_pxx=pxx_dbm(1, (pos1:pos2)); %Eje de potencias del canal
19     [posfc]=max(sub_arreglo_pxx); % Posición de valor maximo de potencia
20     fc=freq_axis_emisora(posfc); % Valor de frecuencia central
21     frecuenciascentrales=[frecuenciascentrales fc]; %Arreglo de frecuencias
22     clear pcentral;
23 end

```

Figura 2.13 Código para la obtención de las Frecuencias Centrales

Con el arreglo que contiene los valores de frecuencias centrales se procede a obtener los valores estadísticos de media, mediana y desviación estándar con el siguiente código de la Figura 2.14.

```

24 a=sort(frecuenciascentrales,'ascend');
25 meanfc=mean(a,'omitnan'); %Cálculo de Media
26 medianfc=median(a,'omitnan'); %Cálculo de Mediana
27 stdfc=std(a,'omitnan'); % Cálculo de Desviación Estándar
28 n=s(1);
29 end
30

```

Figura 2.14 Código para la obtención de valores estadísticos de las Frecuencias Centrales

En la Figura 2.15 se muestra el código implementado para la obtención de los anchos de banda de los datos del archivo cargado en la función.

```

median_fc.m  median_bw.m  median_ic.m  normaldistr.m  reporte_emisora.m  +
1  function [medianbw, meanbw, stdbw, bws] = median_bw(ruta, tuner_freq)
2  datos=xlsread(ruta); % Guarda los datos del archivo cargado en esta variable
3  tunerfreq=tuner_freq*1e6; %Convierte la frecuencia de sintonización a MHz
4  x=tunerfreq-0.03e6; %Determina el limite inferior para determinar el ancho de banda
5  y=tunerfreq+0.03e6; %Determina el limite superior para determinar el ancho de banda
6  freq_axis=(datos(1, (2:end)))*1e6; %Obtiene el eje de frecuencias en MHz
7  bws=[];
8  s = size(datos);
9  for i=2:1:s(1) %Recorre todos los datos del archivo cargado
10     pxx_dbm=datos(i, (2:end));
11     pxx_db=pxx_dbm-30;
12     pxx=10.^(pxx_db/10); %Conversion a Watts
13     bw=powerbw(pxx, freq_axis, [x y]); %Funcion Powerbw determinar el ancho de banda del canal especificado
14     bws=[bws bw]; %Arreglo de Anchos de Banda
15 end
16
17

```

Figura 2.15 Código para la obtención de los Anchos de Banda

Con el arreglo que contiene los valores de anchos de banda se procede a obtener los valores estadísticos de media, mediana y desviación estándar con el siguiente código de la Figura 2.16.

```

18 - b=sort(bws,'ascend');
19 - meanbw=mean(b,'omitnan');           %Cálculo de Media
20 - medianbw=median(b,'omitnan');      %Cálculo de Mediana
21 - stdbw=std(b,'omitnan');           %Cálculo de Desviación Estándar
22 - end

```

Figura 2.16 Código para la obtención de valores estadísticos de los Anchos de Banda

En la Figura 2.17 se muestra el código implementado para la obtención de las intensidades de campo de los datos del archivo cargado en la función.

```

1 function [medianic, meanic, stdic, fieldintensity] = medianic(ruta,tuner_freq)
2 %Guarda los datos del archivo cargado en esta variable
3 datos=xlsread(ruta); %Eje de frecuencias del archivo cargado
4 freq_axis=(datos(1,(2:end)));
5 s=size(datos);
6 fieldintensity=[]; %Arreglo vacio donde se guardan las Intensidades_Campo
7 for i=2:1:s(1) %Recorre todos los datos del archivo cargado
8 pxx_dbm=datos(i,(2:end)); %Datos de la iesima iteracion del archivo cargado
9 pcentral = 256; %Valor central del eje de frecuencias
10 pant = 144; %Valor anterior para determinar el canal sintonizado
11 psig = 370; %Valor posterior para determinar el canal sintonizado
12 sub_arreglo_pot1=(pxx_dbm(1,(pant:pcentral)));
13 sub_arreglo_pot2=(pxx_dbm(1,(pcentral:psig)));
14 value1=min(sub_arreglo_pot1); %Menor valor en el sub_arreglo
15 value2=min(sub_arreglo_pot2); %Mayor valor en el sub_arreglo
16 pos1=find(pxx_dbm==value1); %Posicion del menor valor encontrado en datos
17 pos2=find(pxx_dbm==value2); %Posicion del mayor valor encontrado en datos
18 freq_axis_emisora=freq_axis(1,(pos1:pos2)); %Eje de frecuencias del canal
19 sub_arreglo_pxx=pxx_dbm(1,(pos1:pos2)); %Eje de potencias del canal
20 [pot, posfc]=max(sub_arreglo_pxx); %Posicion y valor maximo de potencia
21 fc=freq_axis_emisora(posfc); %Valor de frecuencia central
22 dBu=pot+(20*log10(fc))+77.2-6; %Conversion a dBu para determinar Intensidad de Campo
23 fieldintensity=[fieldintensity dBu]; %Arreglo de Intensidades de Campo del archivo cargado
24 end

```

Figura 2.17 Código para la obtención de las Intensidades de Campo

Con el arreglo que contiene las intensidades de campo se procede a obtener los valores estadísticos de media, mediana y desviación estándar con el siguiente código de la Figura 2.18.

```

24 - a=sort(fieldintensity);
25 - medianic=median(a,'omitnan');      %Cálculo de Media
26 - meanic=mean(a,'omitnan');        %Cálculo de Mediana
27 - stdic=std(a,'omitnan');          %Cálculo de Desviación Estándar
28 - end

```

Figura 2.18 Código para la obtención de valores estadísticos de las Intensidades de Campo

## 2.5 Diseño del algoritmo para generación automatizada del reporte

Para la obtención del reporte los datos tomados son procesados a través de un criterio prueba de hipótesis con un 95% de confiabilidad, esto se realiza con la finalidad de tener valores estandarizados con los cuales se pueda trabajar, ya que los valores tanto para la frecuencia central, ancho de banda e intensidad de campo varían respecto al tiempo, por lo que se obtiene su media, mediana, desviación estándar.

El formato del reporte se realiza de manera automatizada mediante el código de la Figura 2.19, mostrando los valores estadísticos obtenidos de las funciones anteriores para los parámetros que se analizan y un comentario indicando si la emisora cumple o no la norma técnica, como se muestra en la Figura 2.20. Para el criterio de la frecuencia central elegida basta que la variación se encuentre en el rango de  $\pm 0.02$  kHz para cumplir la norma (Figura 2.21), sin embargo se les aplica el criterio de prueba de hipótesis para reafirmar dicha hipótesis utilizando los resultados obtenidos para la mediana y la media, si se da el caso que ambas cumplen, la frecuencia elegida será la que tenga menor variación respecto a la frecuencia autorizada; para la columna de comentarios se escribe si este parámetro cumple la norma, en caso de no hacerlo se guarda un comentario indicando que no cumple la norma, en la frecuencia elegida se muestra el error relativo respecto a la frecuencia central teórica.

El valor elegido para el ancho de banda y la intensidad de campo eléctrico está dado por escoger el mayor de estos valores, si la mediana es mayor a la media, se escoge como valor de ancho de banda elegido o viceversa, ya que este parámetro no está restringido por un límite inferior. Para el parámetro de ancho de banda basta que sea menor a 220 kHz para cumplir (Figura 2.22). La intensidad de campo no está restringida por un límite superior, basta con que sea mayor a 54dBu V/m (Figura 2.23) para cumplir por tal motivo se opta por el valor mayor entre la media y la mediana para escoger la intensidad de campo eléctrico. Finalmente debajo de las tres categorías de los parámetros mencionados anteriormente se crea una fila para el comentario final, que tiene como finalidad determinar si dicha frecuencia autorizada está cumpliendo la norma, para ello debe cumplir los tres parámetros analizados: frecuencia central, ancho de banda, intensidad de campo eléctrico, cuando se cumplen dichas condiciones se puede decir que la frecuencia autorizada

no está infligiendo la norma técnica de la ARCOTEL y opera correctamente de lo contrario la incumple y se genera un comentario diciendo “La frecuencia autorizada no cumple la norma técnica”.

```

1  function ruta = reporte_emisora(fPortadoraTeorica,medianfc,meanfc,stdfc,medianbw,meanbw,stdbw,mediar
2  VarNombreEmisora = {'Frecuencia Sintonizada [MHz]'};
3  VarFrecCentral = {'Valores estadísticos para Frec. Central [MHz]'};
4  VarValoresStd = {'Media','Mediana','Desviación'};
5  VarFrecCElegida = {'Frec. Central Elegida o Error Relativo'};
6  VarAnchoBanda = {'Valores estadísticos para Ancho de Banda [kHz]'};
7  VarBWElegido = {'Ancho de Banda Elegido o Error Relativo '};
8  VarIntensidadCampo = {'Valores estadísticos para Intensidad de Campo [dBuV/m]'};
9  VarICElegido = {'Intensidad de Campo Elegido o Error Relativo'};
10 comment = {'Comentarios'};
11 finalcomment = {'Comentario Final'};
12 fCentralTeoricaMax=fPortadoraTeorica+0.002; % Valor teorico maximo permitido por la norma tecnica A
13 fCentralTeoricaMin=fPortadoraTeorica-0.002; %Valor teorico minimo permitido por la norma tecnica arc
14 e = actxserver('Excel.Application');
15 eWorkbook = e.Workbooks.Add;
16 e.Visible = 1;
17 eSheets = e.ActiveWorkbook.Sheets;
18 eSheet1 = eSheets.get('Item',1);
19 eSheet1.Activate;
20
21
22 %FORMATO REPORTE
23 eActiveSheetRange = e.ActiveSheet.get('Range','D2');
24 eActiveSheetRange.Value = fPortadoraTeorica;
25 eActiveSheetRange = e.ActiveSheet.get('Range','A2:C2');
26 eActiveSheetRange.MergeCells = 1;
27 eActiveSheetRange.Value = VarNombreEmisora;
28 eActiveSheetRange = e.ActiveSheet.get('Range','A3:C4');
29 eActiveSheetRange.MergeCells = 1;
30 eActiveSheetRange.Value = VarFrecCentral;
31 eActiveSheetRange = e.ActiveSheet.get('Range','A5:C5');
32 eActiveSheetRange.Value = VarValoresStd;
33 eActiveSheetRange = e.ActiveSheet.get('Range','D3:D4');
34 eActiveSheetRange.MergeCells = 1;
35 eActiveSheetRange.Value = VarFrecCElegida;
36 eActiveSheetRange = e.ActiveSheet.get('Range','E3:E4');
37 eActiveSheetRange.MergeCells = 1;
38 eActiveSheetRange.Value = comment;
39 eActiveSheetRange = e.ActiveSheet.get('Range','A7:C8');
40 eActiveSheetRange.MergeCells = 1;
41 eActiveSheetRange.Value = VarAnchoBanda;
42 eActiveSheetRange = e.ActiveSheet.get('Range','A9:C9');
43 eActiveSheetRange.Value = VarValoresStd;
44 eActiveSheetRange = e.ActiveSheet.get('Range','D7:D8');
45 eActiveSheetRange.MergeCells = 1;
46 eActiveSheetRange.Value = VarBWElegido;

```

Figura 2.19 Código para generación de reporte según el formato establecido

Nombre de la emisora			88.1	
Valores estadísticos para Frec. Central			Frec Central	Comentarios
Media	Mediana	Desviación		
88.0993705	88.1	0.0034073	88.1	La frecuencia central cumple la norma tecnica tomando el valor de la mediana
Valores estadísticos para Ancho de Banda			Ancho de Banda	Comentarios
Media	Mediana	Desviación		
60147.6708	72157.7646	25806.92856	72157.7646	El ancho de banda cumple la norma tecnica tomando el valor de la mediana
Valores estadísticos para Intensidad de Campo			Intensidad de Campo	Comentarios
Media	Mediana	Desviación		
70.4803541	69.5967297	2.441451234	70.4803541	La intensidad de campo cumple la norma tecnica tomando el valor de la mediana
Comentario Final				
La emisora cumple la norma tecnica				

Figura 2.20 Reporte de la emisora medida

```

93 %Condicion norma tecnica
94 %Intervalo de confianza 95%
95 % ANALISIS FRECUENCIA CENTRAL
96 Z1=1.64;
97 Z2=-1.64;
98 Homean1fc=0; %Hipotesis nula u>= fc-0.002
99 Homean2fc=0; %Hipotesis nula u<= fc+0.002
00 Homeedian1fc=0;
01 Homeedian2fc=0;
02 Zmean=(meanfc-fCentralTeoricaMin)/(stdfc/(sqrt(n))); %Z para media
03 Zmedian=(medianfc-fCentralTeoricaMin)/(stdfc/(sqrt(n))); %Z para mediana
04 if Zmean>Z2
05 Homean1fc=1;
06 else
07 Homean1fc=0;
08 end
09
10 if Zmedian>Z2
11 Homeedian1fc=1;
12 else
13 Homeedian1fc=0;

```

Figura 2.21 Análisis para Frecuencia Central

```

224 %ANALISIS ANCHO DE BANDA
225 Z1=1.64;
226 Z2=-1.64;
227 HoMeanBw=0; % Hipotesis nula con mediaBw u<=220 kHz;
228 HoMedianBw=0; %Hipotesis nula con medianaBw u<=220 kHz
229 ZmeanBw=(meanbw-220000)/(stdbw/(sqrt(n))); %Z para media
230 ZmedianBw=(medianbw-220000)/(stdbw/(sqrt(n))); %Z para mediana
231 if ZmeanBw<Z1
232 HoMeanBw=1;
233 flagcumplemedia=1;
234 else
235 HoMeanBw=0;
236 flagcumplemedia=0;
237 end
238
239 if ZmedianBw<Z1
240 HoMedianBw=1;
241 flagcumplemediana=1;
242 else
243 HoMedianBw=0;
244 flagcumplemediana=0;
245 end
246

```

Figura 2.22 Análisis para Ancho de Banda

```

326 % ANALISIS INTENSIDAD DE CAMPO
327 Z1=1.64;
328 Z2=-1.64;
329 HoMeanIC=0; % Hipotesis nula con mediaIC u<=56 dBu;
330 HoMedianIC=0; %Hipotesis nula con medianaIC u<=56 dBu
331
332 ZmeanIC=(meanfieldint-56)/(stdfieldint/(sqrt(n))); %Z para media
333 ZmedianIC=(medianfieldint-56)/(stdfieldint/(sqrt(n))); %Z para mediana
334
335 if ZmeanIC>Z2
336 HoMeanIC=1;
337 flagcumplemedia=1;
338 else
339 HoMeanIC=0;
340 flagcumplemedia=0;
341 end
342
343 if ZmedianIC>Z2
344 HoMedianIC=1;
345 flagcumplemediana=1;
346 else
347 HoMedianIC=0;
348 flagcumplemediana=0;
349 end

```

Figura 2.23 Análisis para Intensidad de Campo

## 2.6 Algoritmo para descartar datos aberrantes

Dado que el RTL-SDR toma valores cada instante de tiempo existe la posibilidad de presentar datos muy por debajo de los valores permitidos que, aunque para el análisis correspondiente a la norma técnica son correctos no son valores que puedan ser considerados reales. Para esto se diseñó un algoritmo que elimina datos aberrantes y genera una nueva hoja de cálculo con los nuevos datos sin incluir los datos aberrantes. A continuación, se puede ver el código de la función *data\_no\_aberrante.m* basada en dicho algoritmo, la cual devuelve los valores estadísticos de los datos no aberrantes y la ruta donde se guardó dicho archivo. (Figura 2.24)

```
median_fc.m  median_bw.m  median_ic.m  reporte_emisora.m  data_no_aberrante.m  +
1  %Crea el nuevo excel con la data descartando los valores aberrantes,
2  %Tambien devuelve los valores de media, mediana, desviacion estandar y ancho de banda
3  function [ruta, medianbw, meanbw, stdbw, bws] = data_no_aberrante(ruta,tuner_freq)
4      datos=xlsread(ruta);
5      tunerfreq=tuner_freq*1e6;
6      x=tunerfreq-0.03e6;
7      y=tunerfreq+0.03e6;
8      freq_axis=(datos(1,(2:end)))*1e6;
9      vfreq_axis=(datos(1,(2:end)));
10     bws=[];
11     s = size(datos);
12     variables=cell(1,length(freq_axis));
13     variables(1)={'Tiempo'};
14     z=length(freq_axis)+1;
15     for n=2:1:z
16         variables(n)={'Frecuencia: ', num2str(n-1)};
17     end
18
19     frecuencias=[0 vfreq_axis];
20     xlswrite(['Data Tuner Frequency Not Aberrant ',num2str(tunerfreq/1e6),' MHz.xlsx'],variables);
21     xlswrite(['Data Tuner Frequency Not Aberrant ',num2str(tunerfreq/1e6),' MHz.xlsx'],frecuencias,')
22     j=3;
23
24     for i=3:1:s(1)
25         pxx_dbm=datos(i,(2:end));
26         reg = datos(i,(1:end));
27         pxx_db=pxx_dbm-30;
28         pxx=10.^(pxx_db/10);
29         bw = powerbw(pxx,freq_axis,[x y]);
30         if bw > 30*1e3
31             xlswrite(['Data Tuner Frequency Not Aberrant ',num2str(tunerfreq/1e6),' MHz.xlsx'],reg,'Hojal
32             j=j+1;
33             bws=[bws bw];
34         end
35     end
36     b=sort(bws,'ascend');
37     meanbw=mean(b,'omitnan');
38     medianbw=median(b,'omitnan');
39     stdbw=std(b,'omitnan');
40     ruta= ['Data Tuner Frequency Not Aberrant ',num2str(tunerfreq/1e6),' MHz.xlsx'];
41
42     end
```

Figura 2.24 Código para descartar datos aberrantes

## 2.7 Diseño de la interfaz gráfica de la aplicación

El algoritmo de MATLAB genera una ventana para que el usuario final pueda interactuar y manipular directamente las variables al realizar las mediciones. La primera ventana tiene como título “Medición y caracterización del Espectro FM en la ciudad de Guayaquil”, tiene dos botones: el primero sirve para **medir** y el segundo para **salir**, como se muestra en la Figura 2.25; en la misma ventana se da al usuario la posibilidad de escoger el tiempo que se desea medir en intervalos de 5,10,15,20,25 y 30 minutos, de igual manera puede escoger la emisora que se requiera caracterizar.

Cuando se ha cumplido el tiempo de medición asignado por el usuario, es decir haber seleccionado la opción **Medir**, aparece otra ventana la cual tiene dos botones, con las opciones **Generar Reporte y continuar** y **Volver al menú anterior**, como se muestra en la Figura 2.26; si se escoge la opción generar reporte el dispositivo RTL-SDR procesa los datos que ha tomado y los guarda en un archivo.xlsx.

Una vez que se ha generado el Reporte, aparece otra ventana con tres botones, como se muestra en la figura 2.27 , la primera opción **Abrir reporte** permite ver la hoja de cálculo de Excel que se generó en la ventana anterior con todos los detalles de la frecuencia central, ancho de banda e intensidad de campo para dicha emisora, la segunda opción **Mostrar distribución de datos** permite al usuario ver el comportamiento estadístico de cómo están distribuidos los datos para cada uno de los parámetros medidos y la tercera opción **Volver al Menú inicial** permite al usuario volver a realizar una nueva medición o si desea salir, seleccionar la opción **Salir** para dar por finalizada la sesión. Una vez que se han añadido todas las emisoras que el usuario desea, se da la opción de generar un reporte final con el detalle de todos los parámetros a través de la ventana **Reporte Final** como se muestra en la figura 2.28.



Figura 2.25 Ventana Principal de la aplicación



Figura 2.26 Segunda Ventana de la aplicación



Figura 2.27 Tercer Ventana de la aplicación



Figura 2.28 Ventana generación Reporte Final de la aplicación

# CAPÍTULO 3

## 3. RESULTADOS Y ANÁLISIS

En el proyecto se pudo constatar que las emisoras de radiofrecuencia FM que pertenecen a Guayaquil cumplieron con los parámetros técnicos de la ARCOTEL en su mayoría. La frecuencia central por emisora presentó poca variación cuando se utilizó el barrido de frecuencias por emisora, para esto se obtuvo valores estadísticos como la desviación estándar, la media y la mediana; donde se realizó pruebas de hipótesis a un 95% de confiabilidad.

Se obtuvo la mediana como una posible opción ante la media, se escogió siempre la mayor de estas dos para poder así obtener el mejor valor para realizar el análisis estadístico. Se demostró a distintos intervalos de frecuencia y se tomaron como ejemplo la primera frecuencia central Radio María 88.1 MHz, una frecuencia intermedia Radio Disney 93.7 MHz y una frecuencia que este cerca de la última emisora FM como lo es BBN Radio 106.1 MHz

La norma técnica de la ARCOTEL estableció que, para frecuencia central, la tolerancia no debería superar  $\pm 2\text{kHz}$  por lo tanto para este caso la mediana y la media deben estar dentro de este rango, el ancho de banda debe ser menor o igual a 220 KHz y la intensidad de campo eléctrico debió ser mayor a  $54\text{dB}\mu\text{V/m}$ . Una vez que se estableció estas condiciones se plantearon las hipótesis nulas, Recordando que para un 95% de confiabilidad y una condición mayor o igual en la hipótesis nula se obtuvo una cota a la izquierda en una curva de la normal estándar a un valor Z igual a -1.64 , si el valor Z obtenido mediante la media , la desviación estándar y el número de muestras es inferior a la cota izquierda entonces la hipótesis nula se rechaza, caso contrario si fuese mayor al valor de la cota izquierda entonces se puede decir que la hipótesis nula es aceptada. Para el caso que la hipótesis nula tenga una condición que es menor o igual se obtiene una cota a la derecha en Z igual a 1.64, para que la hipótesis nula sea aceptada debería ser menor al valor de la cota derecha caso contrario si es mayor que Z igual a 1.64 entonces la hipótesis nula es rechazada. Todos estos datos se tomaron el día 22 de julio del 2018 durante 10 min a las 18:00 horas. El número de muestras que se obtuvieron variaron entre 270 y 280 muestras.

Para la frecuencia central se planteó

$$H_o : \mu \geq f_c - 0.002$$

$$H_1 : \mu < f_c - 0.002$$

$$H_o : \mu \leq f_c + 0.002$$

$$H_1 : \mu > f_c + 0.002$$

Para el ancho de banda se planteó

$$H_o : \mu \leq 220$$

$$H_1 : \mu > 220$$

Para la intensidad de campo se planteó

$$H_o : \mu \geq 54$$

$$H_1 : \mu < 54$$

La primera emisora que fue analizada fue Radio María (88.1 MHz), se determinaron sus medias y desviaciones estándares para los parámetros de frecuencia central, ancho de banda e intensidad de campo eléctrico. Para la frecuencia central se tuvo una media de 88.105, una desviación estándar de 0.01204 por tanto se procedió a realizar las siguientes pruebas de hipótesis.

$$H_o : \mu \geq 88.098$$

$$H_1 : \mu < 88.098$$

$$H_o : \mu \leq 88.102$$

$$H_1 : \mu > 88.102$$

$$Z = \frac{x - \mu}{\sigma / (\sqrt{n})} = \frac{88.105 - 88.098}{0.01204 / (\sqrt{270})}$$

$$Z = \frac{x - \mu}{\sigma / (\sqrt{n})} = \frac{88.105 - 88.102}{0.01204 / (\sqrt{270})}$$

$$Z = 9.55$$

$$Z = 4.09$$

Para la primera hipótesis el valor de Z es mayor a  $Z = -1.64$  por lo tanto la hipótesis se acepta, es decir el valor cumple con el límite inferior para las frecuencias centrales, pero para la segunda hipótesis nula el valor Z debería ser menor a  $Z = 1.64$  para que la hipótesis se acepte, pero no es así, por este motivo la hipótesis nula se rechaza. Cuando se analizó los valores de frontera de la frecuencia central se pudo comprobar que cumplió para el límite superior, sin embargo, para el límite superior no cumplió el parámetro por ende la frecuencia central para esta emisora no cumple con la norma técnica.

Luego se analizó el ancho de banda para esta emisora con las siguiente mediana y desviación estándar de 96.62 y 27.16 respectivamente. A continuación, se planteó la siguiente hipótesis

$$H_0 : \mu \leq 220$$

$$H_1 : \mu > 220$$

$$Z = \frac{x - \mu}{\sigma/(\sqrt{n})} = \frac{96.62 - 220}{27.16/(\sqrt{270})}$$

$$Z = -76.64$$

Para cumplir que la hipótesis nula sea aceptada el valor Z tuvo que haber sido menor a  $Z = 1.64$ , como este fue el caso se puede decir con un 95% de confiabilidad que esta emisora si cumple la norma técnica para el ancho de banda.

Finalmente se analizó su intensidad de campo eléctrico con media igual a 78.36 y desviación estándar de 4.3206, luego se planteó la siguiente hipótesis

$$H_0 : \mu \geq 54$$

$$H_1 : \mu < 54$$

$$Z = \frac{x - \mu}{\sigma/(\sqrt{n})} = \frac{78.36 - 54}{4.3206/(\sqrt{270})}$$

$$Z = 92.64$$

Para que la hipótesis nula fuera aceptada el valor Z debió ser mayor que  $Z = -1.64$  como este fue el caso, se pudo determinar con un 95% de confiabilidad que Radio María si cumplió con la norma técnica para la intensidad de campo eléctrico.

A continuación, se analizó la emisora Radio Disney, de la misma manera que la emisora anterior; para la frecuencia central se tuvo una media de 93.699 y desviación estándar de 0.012576, se procedió a realizar las siguientes pruebas de hipótesis.

$$H_0 : \mu \geq 93.698$$

$$H_1 : \mu < 93.698$$

$$H_0 : \mu \leq 93.702$$

$$H_1 : \mu > 93.702$$

$$Z = \frac{x - \mu}{\sigma/(\sqrt{n})} = \frac{93.699 - 93.698}{0.012576/(\sqrt{280})}$$

$$Z = \frac{x - \mu}{\sigma/(\sqrt{n})} = \frac{93.699 - 93.702}{0.012576/(\sqrt{280})}$$

$$Z = 1.33$$

$$Z = -3.99$$

Con un nivel de confianza del 95% cuando la hipótesis nula es mayor que tiene una cota a la derecha donde se rechaza la hipótesis nula cuando es menor a un valor Z igual a -1.64, por lo tanto, para la primera hipótesis nula se acepta; para la segunda hipótesis nula cuando se usa la expresión con el termino menor igual a un 95% de confianza la cota está a la derecha y debe ser mayor a Z igual a 1.64 para ser rechazada, como no es así la hipótesis se acepta.

Como se pudo apreciar ambas hipótesis fueron aceptadas, es decir están dentro del rango permitido por lo tanto se pudo decir con un 95% de confiabilidad que Radio Disney si cumplió con la norma técnica para la frecuencia central.

Para el ancho de banda se tiene una media de 86.46 kHz y una desviación estándar de 24.96 utilizando 270 muestras, por lo tanto, para establecer con un nivel de confianza del 95% se aplicó la siguiente hipótesis:

$$H_0 : \mu \leq 220$$

$$H_1 : \mu > 220$$

$$Z = \frac{x - \mu}{\sigma/(\sqrt{n})} = \frac{86.46 - 220}{24.96/(\sqrt{270})}$$

$$Z = -87.91$$

Como el valor de Z fue menor que  $Z = 1.64$  la hipótesis nula se acepta, comprobando así que el ancho de banda si cumplió el parámetro de ancho de banda al ser menor que 220 KHz.

Además, se obtuvo la media y la desviación estándar para la intensidad de campo cuyos valores son 66.7627 y 2.0922 respectivamente. Recordando que para cumplir el parámetro técnico para intensidad de campo eléctrico debe ser mayor a 54 dB $\mu$ V/m

Con estos datos se obtuvo su respectiva prueba de hipótesis con un 95% de confiabilidad usando:

$$H_0 : \mu \geq 54$$

$$H_1 : \mu < 54$$

$$Z = \frac{x - \mu}{\sigma/(\sqrt{n})} = \frac{66.7627 - 54}{2.0922/(\sqrt{270})}$$

$$Z = 100$$

Como el valor de Z fue mayor que  $Z = -1.64$  la hipótesis nula se acepta es decir que el parámetro de intensidad de campo eléctrico si se cumple para esta frecuencia.

La última emisora que se analizó fue BBN Radio 106.1 MHz, se obtuvieron medias y desviaciones estándares para los tres parámetros a analizar, para su posterior análisis estadístico mediante prueba de hipótesis. Para la frecuencia central esta emisora tiene una media de 106.102, una desviación estándar de 0.0039, a continuación, se planteó la siguiente hipótesis.

$$H_o : \mu \geq 106.098$$

$$H_1 : \mu < 106.098$$

$$H_o : \mu \leq 106.102$$

$$H_1 : \mu > 106.102$$

$$Z = \frac{x - \mu}{\sigma / (\sqrt{n})} = \frac{106.102 - 106.098}{0.0039 / (\sqrt{270})}$$

$$Z = \frac{x - \mu}{\sigma / (\sqrt{n})} = \frac{106.102 - 106.102}{0.0039 / (\sqrt{270})}$$

$$Z = 16.85 \quad Z = 0$$

Para la primera hipótesis el valor de  $Z = 16.85$  que es mayor que  $Z = -1.64$  por lo tanto la hipótesis se acepta, es decir si cumple con el límite inferior para esta frecuencia, la segunda hipótesis nula el valor  $Z = 0$  que fue menor a  $Z = 1.64$  por este motivo la hipótesis nula se acepta. Cuando se analizó los valores de frontera de la frecuencia central se pudo comprobar que cumplió para el límite superior y con el límite superior, por lo tanto, la frecuencia central para esta emisora si cumple con la norma técnica

Luego se analizó el ancho de banda para esta emisora con las siguiente mediana y desviación estándar de 72.97y 29.87 respectivamente. A continuación, se planteó la siguiente hipótesis

$$H_o : \mu \leq 220$$

$$H_1 : \mu > 220$$

$$Z = \frac{x - \mu}{\sigma / (\sqrt{n})} = \frac{72.97 - 220}{29.87 / (\sqrt{270})}$$

$$Z = -80.88$$

Para cumplir que la hipótesis nula sea aceptada el valor Z tuvo que haber sido menor a  $Z = 1.64$ , como este fue el caso se puede decir que con un 95% de confiabilidad que esta emisora si cumple la norma técnica para el ancho de banda. Finalmente se analizó su intensidad de campo eléctrico con media igual a 80.51 y desviación estándar de 4.8064, luego se planteó la siguiente hipótesis

$$H_0 : \mu \geq 54$$

$$H_1 : \mu < 54$$

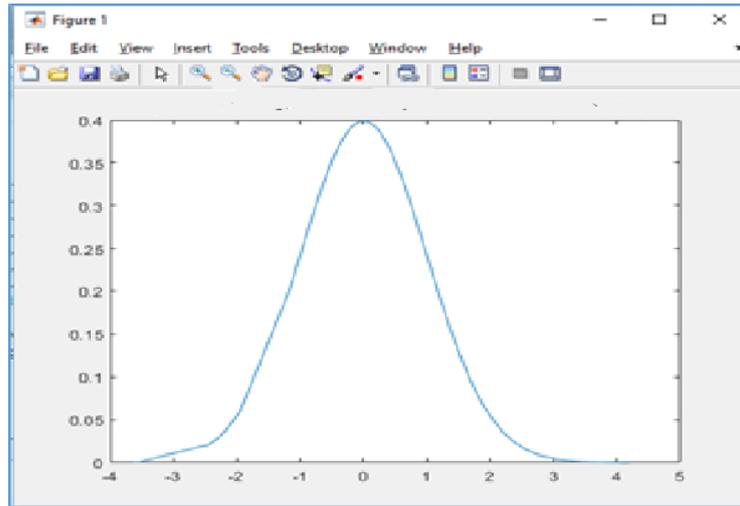
$$Z = \frac{x - \mu}{\sigma / (\sqrt{n})} = \frac{80.51 - 54}{4.8064 / (\sqrt{270})}$$
$$Z = 90.63$$

Para que la hipótesis nula fuera aceptada el valor Z debió ser mayor que  $Z = -1.64$  como este fue el caso, se pudo determinar con un 95% de confiabilidad que BBN Radio si cumplió con la norma técnica para la intensidad de campo eléctrico

Con este análisis se logró determinar mediante un análisis estadístico cuando un parámetro cumple la norma técnica, ya que la toma de datos se realizó a distintos intervalos de tiempo a las emisoras FM de la ciudad de Guayaquil ; en algunos instantes de tiempo si cumplían pero en otros no; la frecuencia central no era la misma para cada tiempo ni el ancho de banda ni la intensidad de campo, por lo que se optó por este análisis para que se pueda tener un dato que abarque un promedio de todas las mediciones que se obtuvieron en los 10 min de medición dando como resultados que fue el único parámetro que presentaba el incumplimiento en algunas emisoras, los demás parámetros como ancho de banda e intensidad de campo eléctrico siempre superaban el umbral de los 54dB $\mu$ V/m debido a esto y que la cobertura principal se refirió a un entorno urbano, dichas emisoras si pueden ser escuchadas en el área geográfica de Guayaquil, algunas mejor , otras en menor calidad pero la sintonización de las mismas si es apreciable, La calidad de la señal en la sintonización se debe además a la ubicación geográfica donde se estuvieran escuchando, obstáculos como colinas o valles donde la recepción es menor pudieran provocar que la calidad de la señal fuese menor, sin embargo eso no significó que las emisoras estuvieran incumpliendo esa norma técnica. El ancho de banda se cumplió para todas las emisoras dado que mediante el programa MATLAB se obtuvo que este fue menor al asignado en la norma técnica, variaciones en el ancho de banda se deben a lo que se estuviera transmitiendo a dichas horas.

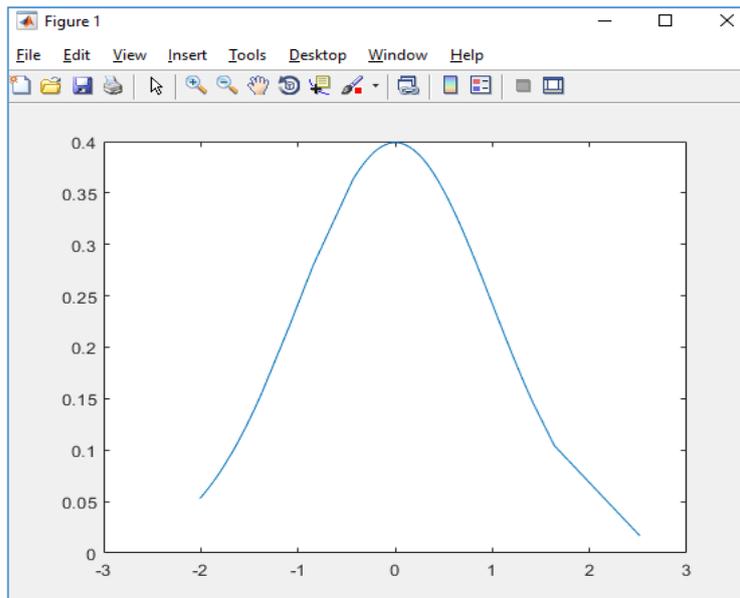
Luego se realizó la estandarización de los datos para la frecuencia central, el ancho de banda y la intensidad de campo se obtienen los siguientes resultados.

La distribución para la frecuencia central muestra una curva con sus datos distribuidos alrededor de la media en proporción similar, es decir existen casi la misma cantidad de valores superiores e inferiores a la media para este parámetro, como se observa en la figura 3.1



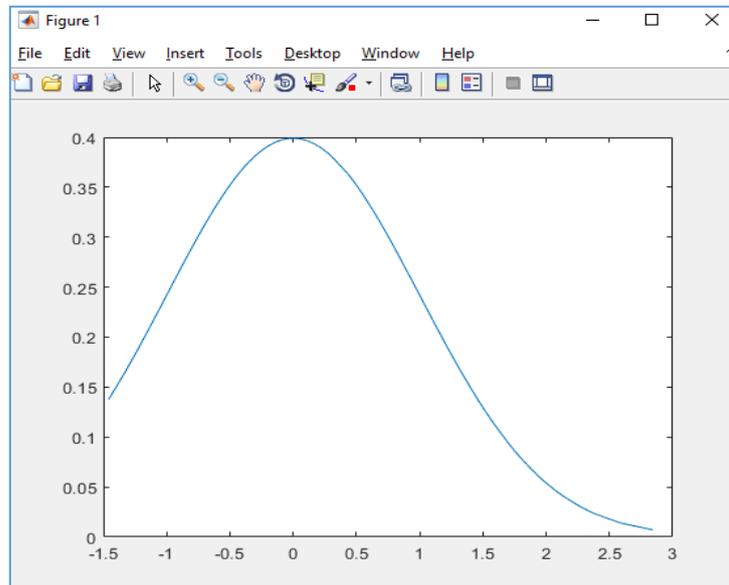
**Figura 3.1 Distribución para la frecuencia central**

En el caso del ancho de banda se puede apreciar que la mayor parte de los datos se concentra en valores mayores a la media, con un ligero desfase respecto a la media., como se observa en la figura 3.2



**Figura 3.2 Distribución para el ancho de banda**

La distribución mostrada para la intensidad de campo mostrada, muestra un mayor desfase respecto al ancho de banda, ya que en su gran mayoría sus valores son superiores a la media y cuenta con pocos valores que sean menores., como se observa en la figura 3.3



**Figura 3.3 Distribución para la intensidad de campo**

El Reporte Final con las emisoras medidas el día 27 de agosto se muestra a continuación en la Figura 3.4

Frecuencia Autorizada [MHz]	Frecuencia Central [MHz]	Ancho de Banda [kHz]	Intensidad de Campo [dBuV/m]	Comentarios
88.1	88.1	78.40549603	69.50210653	La emisora cumple la norma tecnica
88.5	Error de 0.0108% respecto de Frecuencia Central	66.35614155	67.88598451	La emisora no cumple la norma tecnica
89.3	89.30185547	131.835685	65.48059147	La emisora cumple la norma tecnica
89.7	Error de 0.0162284 % respecto de la Frecuencia Central	83.77785918	64.56352253	La emisora no cumple la norma tecnica
90.1	90.1	46.41972851	73.28079409	La emisora cumple la norma tecnica
90.5	Error de 0.0147959 % respecto de la Frecuencia Central	49.40785352	75.91439874	La emisora no cumple la norma tecnica
90.9	90.9027832	Error de 115.487393 % respecto del ancho de banda permitido	60.79440116	La emisora no cumple la norma tecnica
91.3	Error de 0.0055327 % respecto de la Frecuencia Central	73.02543534	69.70336478	La emisora no cumple la norma tecnica
91.7	Error de 0.036589 % respecto de la Frecuencia Central	79.75963848	77.1276407	La emisora no cumple la norma tecnica
92.1	Error de 0.018757 % respecto de la Frecuencia Central	88.00540991	72.18908978	La emisora no cumple la norma tecnica
92.5	92.49958087	44.00683601	75.56348641	La emisora cumple la norma tecnica
92.9	Error de 0.005972 % respecto de la Frecuencia Central	67.68614737	76.15993095	La emisora no cumple la norma tecnica
93.3	Error de 0.65104 % respecto de la Frecuencia Central	65.36504347	72.25227581	La emisora no cumple la norma tecnica
93.7	93.6972168	111.1384053	64.22076122	La emisora cumple la norma tecnica
94.1	Error de 0.0077 % respecto de la Frecuencia Central	112.9930223	66.02800942	La emisora no cumple la norma tecnica
94.5	94.5	64.67169205	64.95499037	La emisora cumple la norma tecnica
94.9	94.90148609	164.7009322	63.21896117	La emisora cumple la norma tecnica
95.3	95.29953613	52.00965124	70.5455965	La emisora cumple la norma tecnica
96.5	Error de 0.003845 % respecto de la Frecuencia Central	Error de 32.1965 % respecto del ancho de banda permitido	62.40829863	La emisora no cumple la norma tecnica
96.9	Error de 0.00287 % respecto de la Frecuencia Central	157.3662377	64.42556527	La emisora no cumple la norma tecnica
97.7	Error de 0.005697 % respecto de la Frecuencia Central	179.1746996	63.88910881	La emisora no cumple la norma tecnica
98.5	98.4972168	122.0735611	63.65445301	La emisora cumple la norma tecnica
98.9	98.89907227	54.21732137	73.51556732	La emisora cumple la norma tecnica
99.3	99.30092773	47.66288937	69.10812589	La emisora cumple la norma tecnica
99.7	Error de 0.0027915 % respecto de la Frecuencia Central	52.83198644	71.06801481	La emisora no cumple la norma tecnica
100.5	100.4972168	Error de 115.48739 % respecto del ancho de banda permitido	60.63222278	La emisora no cumple la norma tecnica
100.9	Error de 0.0314767 % respecto de la Frecuencia Central	108.6466	67.51261458	La emisora no cumple la norma tecnica
101.3	Error de 0.022496 % respecto de la Frecuencia Central	98.51735866	67.23150018	La emisora no cumple la norma tecnica
101.7	Error de 0.0054729 % respecto de la Frecuencia Central	50.42249243	72.04320903	La emisora no cumple la norma tecnica
102.1	Error de 0.0059059 % respecto de la Frecuencia Central	124.8205075	63.99327902	La emisora no cumple la norma tecnica
102.5	Error de 0.008438 % respecto de la Frecuencia Central	50.26519792	72.28283885	La emisora no cumple la norma tecnica
102.9	102.8983079	70.72295772	73.3279763	La emisora cumple la norma tecnica
103.3	103.3018171	Error de 62.4778 % respecto del ancho de banda permitido	61.35241192	La emisora no cumple la norma tecnica
103.7	103.6976807	69.43278289	72.28746281	La emisora cumple la norma tecnica
104.1	Error de 0.0117339 % respecto de la Frecuencia Central	89.85682686	64.9698294	La emisora no cumple la norma tecnica
104.5	Error de 0.0164 % respecto de la Frecuencia Central	89.74198084	71.68772407	La emisora no cumple la norma tecnica
105.3	105.3	71.36157233	70.15606912	La emisora cumple la norma tecnica
105.7	Error de 0.00965468 % respecto de la Frecuencia Central	112.6143145	63.35494576	La emisora no cumple la norma tecnica
106.1	106.1022095	49.70412259	74.38446944	La emisora cumple la norma tecnica
106.5	Error de 0.0077596 % respecto de la Frecuencia Central	75.31119524	74.6597881	La emisora no cumple la norma tecnica
106.9	Error de 0.0069429 % respecto de la Frecuencia Central	69.29925706	71.10212741	La emisora no cumple la norma tecnica
107.3	Error de 0.00379776 % respecto de la Frecuencia Central	68.08361896	70.43388334	La emisora no cumple la norma tecnica
107.7	107.7017318	54.01183002	72.87548232	La emisora cumple la norma tecnica

Figura 3.4 Reporte Final

Con el reporte final se obtuvieron los siguientes resultados, 19 de las emisoras medidas cumplen con el parámetro de frecuencia central y 24 no cumplen, es decir que cumplen el 44% e incumplen en un 56%, como se observa en la Figura 3.5.

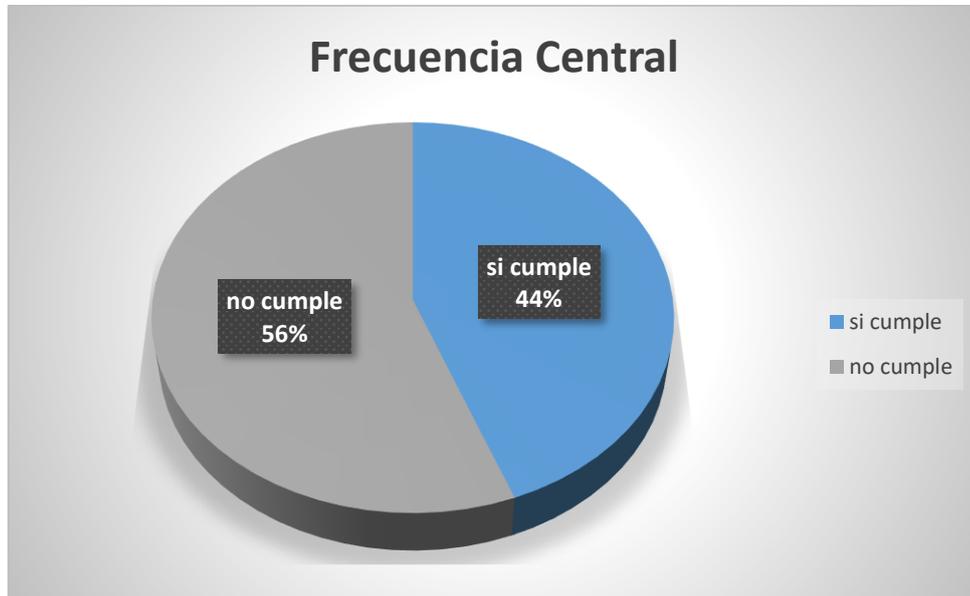


Figura 3.5 Porcentaje de cumplimiento del parámetro frecuencia central

En cuanto al ancho de banda, 39 de las emisoras cumplen y 4 no lo hacen, es decir que cumplen en un 91% y un 9% de estas lo incumplen, como se observa en la Figura 3.6.



Figura 3.6 Porcentaje de cumplimiento del parámetro ancho de banda

La intensidad de campo es el único parámetro que se cumple en todas las emisoras, es decir el 100% de las emisoras medidas cumplen este parámetro, como se observa en la figura 3.7.

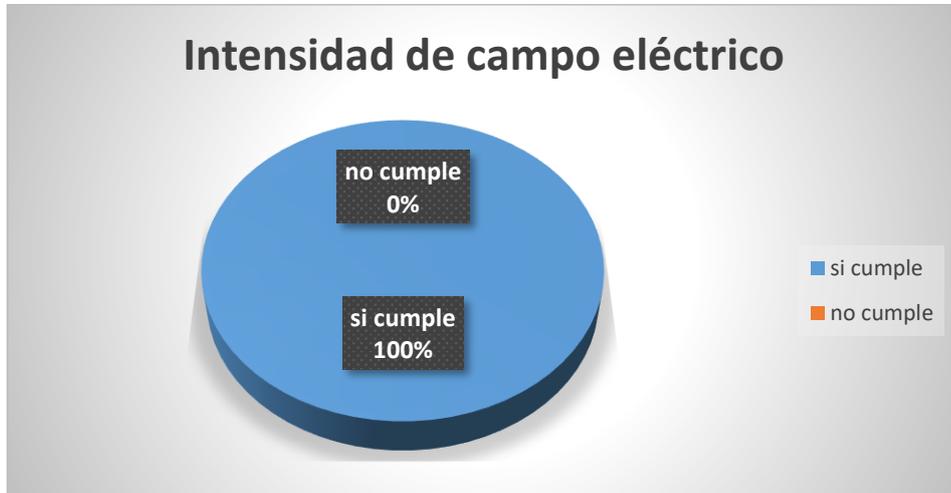


Figura 3.7 Porcentaje de cumplimiento para la intensidad de campo

De forma general, para que se pudiera decir que una emisora cumplía la norma, la frecuencia central, el ancho de banda y la intensidad de campo eléctrico debían cumplir, basta que uno no cumpla para decir que la emisora no cumple con los parámetros técnicos, teniendo esto presente se obtuvieron los siguientes resultados, como se observa en la Figura 3.8.

- 16 emisoras de radiodifusión cumplen.
- 27 emisoras de radiodifusión incumplen.

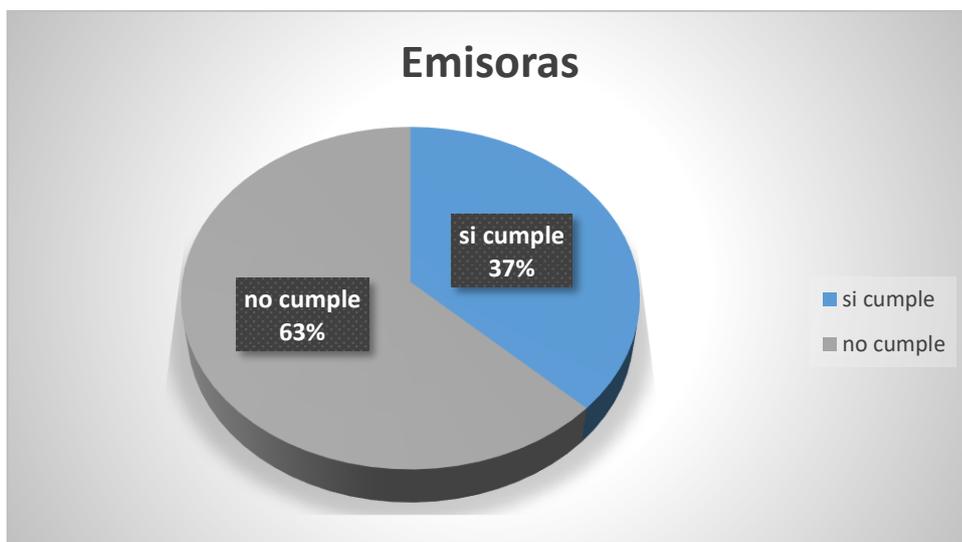


Figura 3.8 Porcentaje de cumplimiento de los parámetros técnicos

Además para determinar si existía una similitud entre los datos obtenidos en ambos equipos se compararon los valores de potencias en el Anexo A., cuya tabla muestra 99 valores de cada equipo y se encontró que tenían un factor de correlación de 0.872 por lo que se demostró que los datos obtenidos en el RTL-SDR se asemejan a los de un analizador espectral, se muestran a continuación en la figura 3.9.

```

data = xlsread('Spectral Analyzer & RTL-SDR Data.xlsx');
%Datos de potencia central del Analizador Espectral
spectral_analyzer_data = data((1:end), 1);
%Datos de potencia del RTL-SDR
rtl_sdr_data = data((1:end),2);
%Correlacion: Datos Analizador Espectral vs. RTL-SDR
R = corrcoef(spectral_analyzer_data, rtl_sdr_data);
disp(R);

```

R =		
	1.0000	0.8727
	0.8727	1.0000

**Figura 3.9** Coeficiente de correlación entre los valores de potencia del RTL-SDR y el analizador espectral

# CAPÍTULO 4

## 4. CONCLUSIONES Y RECOMENDACIONES.

### 4.1 Conclusiones

1. En base a los valores de potencia obtenidos para la frecuencia 93.7 MHz en un analizador espectral y luego con el RTL-SDR se obtuvo un coeficiente de correlación de 0.872; debido a que la correlación es alta se determinó que los valores en ambos equipos son similares.
2. Mediante la implementación del algoritmo de caracterización, se logró reducir el tiempo necesario para el procesamiento de parámetros técnicos facilitando su análisis y emitiendo un reporte detallado
3. La implementación del proyecto proporcionó una herramienta al GICOM que puede ser usada con fines académicos para el estudio del espectro FM.
4. En base a los resultados obtenidos en cuanto a la visualización de los datos se puede concluir que las lecturas por canales fueron mejores ya que se muestra únicamente la emisora que se desea conocer, el espaciamiento es mayor, por lo tanto, la probabilidad de tomar una frecuencia adyacente errónea como la propia frecuencia central es menor, sus parámetros técnicos se obtienen con menor procesamiento y análisis que con el barrido de espectro completo.
5. La implementación de un algoritmo que permita caracterizar el espectro FM mediante MATLAB y RTL-SDR, es viable ya que, aunque se obtiene datos más exactos con un analizador espectral, el algoritmo proporciona una alternativa con resultados similares y de menor costo.
6. Mediante la implementación del algoritmo de caracterización, se logró reducir el tiempo necesario para el procesamiento de parámetros técnicos facilitando su análisis y emitiendo un reporte detallado.
7. En base a los datos obtenidos en el análisis por canal se puede concluir que tanto la frecuencia central, como el ancho de banda y la intensidad de campo tienen un comportamiento de una distribución normal; siendo el ancho de banda el parámetro con mayor desviación debido a que el ancho de banda es estocástico y varía constantemente en el tiempo, a diferencia de los otros parámetros los cuales sus valores fluctuaban, pero en menor proporción

8. Considerando las normas técnicas de la ARCOTEL para la transmisión para emisoras de radiofrecuencias para FM se observa que en un 63% incumplen, además en un 57% de las emisoras medidas se encontró que la frecuencia central se incumple, siendo el parámetro que más se incumple, adicionalmente el ancho de banda también se incumple, pero sólo en un 9%, en contraste con la intensidad de campo eléctrico que siempre cumple.

## **4.2 Recomendaciones**

1. Para futuras mediciones se sugiere contar con un computador con por lo menos 4 GB de RAM, debido a que tomar muchos datos por largos instantes de tiempo produce que tanto el computador como el dispositivo RTL-SDR se vuelvan más lentos produciendo que algunos datos no sean los que se esperan obtener.
2. Para mejorar la calidad con la que se muestran las señales se sugiere utilizar un dispositivo con mayor resolución que el RTL-SDR modelo 820T2, tal es el caso de los últimos modelos del RTL2832U, se paga un mayor costo, pero se obtienen mejores resultados, más visibles y apreciables a la hora de hacer un barrido de espectro completo.
3. Se debe tomar un mayor número de muestras y en mayor tiempo si se desea trabajar con el barrido de frecuencia completo, sin embargo, este modelo de RTL-SDR 820T2 debido a su resolución no toma las lecturas en cada segundo, sino aproximadamente cada quince segundos, debido a esto se sugiere realizar un análisis por canal para este modelo, ya que se obtienen lecturas más fiables y los datos si son tomados cada segundo.
4. Se sugiere realizar la toma de datos por canal, porque se puede apreciar con claridad el espectro que pertenece a cada emisora FM, de esta manera se pueden obtener los parámetros técnicos con menos errores, como es el caso del ancho de banda que al no tener picos de frecuencia tan cerca de la frecuencia central, hay una mínima posibilidad que la función powerbw lo tome como una errónea frecuencia central.
5. Se sugiere sintonizar cada frecuencia y comprobar que se puedan escuchar las transmisiones de las emisoras FM con la finalidad de poder detectar posibles irregularidades en el espectro, como son las interferencias entre canales adyacentes o interferencias por personas apropiándose del espectro ilegalmente.

6. Algunas emisoras como 104.9 y 105.3 FM tienen un bajo nivel de potencia en la noche por lo que para poder medirlas se deben realizar en un horario inferior a las 6 PM
7. Se sugiere tomar mediciones de una emisora por lo menos por cinco minutos para que la calidad de la imagen del espectro promedio sea apreciable, recordando que al hacer el análisis por datos aberrantes y sin datos aberrantes, tomará cinco minutos para generar el reporte con datos aberrantes y cinco minutos adicionales para los datos no aberrantes.
8. Se sugiere dejar descansar un par de minutos cuando se toman muchas mediciones consecutivas para que el dispositivo se enfríe y no arroje valores erróneos.

# BIBLIOGRAFÍA

- [1] Contreras Valdivieso José David, Carvajal Martínez Dennis Becker, Autores. “Medición, caracterización y modelamiento del rango de frecuencias asignado a radiodifusión FM (88-108 MHz) de la banda VHF del espectro electromagnético en la FIEC Campus Prosperina”. Tesis multidisciplinaria de Grado de Ingeniería en Electrónica y Telecomunicaciones e Ingeniería en Electricidad. Electrónica, Escuela Superior Politécnica del Litoral, Ecuador.
- [2] Zhang ke, Autor “Implementación de un demodulador de radio FM definido por Software”. Tesis de Grado de Ingeniería en Sistemas de Telecomunicación, Sonido e Imagen, Universidad Politécnica de Valencia, España.
- [3] Vega León Andy, Autor “Evaluación del espectro radioeléctrico VHF/UHF en parroquias urbanas y rurales del cantón Loja para el despliegue de sistemas de radio cognitiva, Tesis de Maestría en Magister en Redes de Información y Conectividad, Universidad de las Fuerzas Armadas, Ecuador
- [4] Stewart Robert W., Barlee Kenneth W., Atkinson Dale S. W., Crockett Louise H., (2015). Software Defined Radio using MATLAB & Simulink and the RTL-SDR, 1st ed, 2015, pp 3, Strathclyde Academic Media.
- [5] Imagen RTL-SDR Disponible en :  
[https://www.google.com/search?q=rtl+sdr&source=lnms&tbn=isch&sa=X&ved=0ahUKEwj5nqvZ8YHdAhXtqFkKHYPuCYAQ\\_AUICigB&biw=1600&bih=794#imgrc=mwpxsw6hgmPZM](https://www.google.com/search?q=rtl+sdr&source=lnms&tbn=isch&sa=X&ved=0ahUKEwj5nqvZ8YHdAhXtqFkKHYPuCYAQ_AUICigB&biw=1600&bih=794#imgrc=mwpxsw6hgmPZM):
- [6] Imagen RTL-SDR Disponible en:  
[https://www.google.com/search?q=rtl+sdr&source=lnms&tbn=isch&sa=X&ved=0ahUKEwj5nqvZ8YHdAhXtqFkKHYPuCYAQ\\_AUICigB&biw=1600&bih=794#imgrc=oA-5-1\\_JOzJ8DM](https://www.google.com/search?q=rtl+sdr&source=lnms&tbn=isch&sa=X&ved=0ahUKEwj5nqvZ8YHdAhXtqFkKHYPuCYAQ_AUICigB&biw=1600&bih=794#imgrc=oA-5-1_JOzJ8DM):
- [7] Situación de las Radios Comunitarias en el Ecuador. Disponible en:  
[http://www.imaginar.org/docs/Radios\\_Comunitarias.pdf](http://www.imaginar.org/docs/Radios_Comunitarias.pdf)
- [8] Norma técnica para el servicio de radiodifusión sonora en frecuencia modulada analógica disponible en: <http://www.arcotel.gob.ec/wp-content/uploads/downloads/2015/04/NORMA-TECNICA.pdf>
- [9] Emisoras de radio en Guayaquil. Disponible en: <http://worldradiomap.com/ec/guayaquil>

- [10] Stewart Robert W., Barlee Kenneth W., Atkinson Dale S. W., Crockett Louise H., (2015). Software Defined Radio using MATLAB & Simulink and the RTL-SDR, 1st ed, 2015, pp 100-122, Strathclyde Academic Media.
- [11] Stewart Robert W., Barlee Kenneth W., Atkinson Dale S. W., Crockett Louise H., (2015). Software Defined Radio using MATLAB & Simulink and the RTL-SDR, 1st ed, 2015, pp 91-92. Strathclyde Academic Media.
- [12] Stewart Robert W., Barlee Kenneth W., Atkinson Dale S. W., Crockett Louise H., (2015). Software Defined Radio using MATLAB & Simulink and the RTL-SDR, 1st ed, 2015, pp 124. Strathclyde Academic Media
- [13] Stewart Robert W., Barlee Kenneth W., Atkinson Dale S. W., Crockett Louise H., (2015). Software Defined Radio using MATLAB & Simulink and the RTL-SDR, 1st ed, 2015, pp 125-127. Strathclyde Academic Media.
- [14] Stewart Robert W., Barlee Kenneth W., Atkinson Dale S. W., Crockett Louise H., (2015). Software Defined Radio using MATLAB & Simulink and the RTL-SDR, 1st ed, 2015, pp 152. Strathclyde Academic Media.
- [15] Stewart Robert W., Barlee Kenneth W., Atkinson Dale S. W., Crockett Louise H., (2015). Software Defined Radio using MATLAB & Simulink and the RTL-SDR, 1st ed, 2015, pp 153. Strathclyde Academic Media.
- [16] Stewart Robert W., Barlee Kenneth W., Atkinson Dale S. W., Crockett Louise H., (2015). Software Defined Radio using MATLAB & Simulink and the RTL-SDR, 1st ed, 2015, pp 166. Strathclyde Academic Media.
- [17] Stewart Robert W., Barlee Kenneth W., Atkinson Dale S. W., Crockett Louise H., (2015). Software Defined Radio using MATLAB & Simulink and the RTL-SDR, 1st ed, 2015, pp 167. Strathclyde Academic Media.
- [18] Murillo-Fuentes Juan J.,(2007) Fórmulas de Radiopropagación en Decibelios.Documento en Línea, Escuela Técnica Superior de Ingeniería,Universidad de Sevilla. España.

# ANEXOS

## ANEXO A

Datos Analizador Espectral	Datos RTL_SDR
-41,95	-38,708
-41,95	-38,974
-41,954	-39,857
-41,958	-39,981
-41,966	-40,103
-41,97	-40,614
-41,986	-40,828
-41,994	-40,837
-42,014	-41,349
-42,018	-41,389
-42,05	-41,77
-42,054	-41,904
-42,09	-41,979
-42,094	-41,989
-42,138	-42,35
-42,142	-42,418
-42,186	-42,425
-42,202	-42,435
-42,242	-42,467
-42,266	-42,483
-42,302	-42,499
-42,338	-42,513
-42,366	-42,537
-42,418	-42,597
-42,434	-42,598
-42,506	-42,609
-42,506	-42,611
-42,578	-42,65
-42,606	-42,656
-42,65	-42,661
-42,71	-42,673
-42,726	-42,682
-42.802	-42.725

Coeficiente de Correlacion
0,87269064

-42,822	-42,742
-42,878	-42,745
-42,938	-42,766
-42,95	-42,768
-43,022	-42,822
-43,058	-42,86
-43,078	-42,88
-43,094	-42,891
-43,162	-42,907
-43,17	-42,93
-43,182	-42,931
-43,226	-42,936
-43,258	-42,946
-43,29	-42,95
-43,306	-42,962
-43,334	-43,007
-43,35	-43,047
-43,406	-43,054
-43,406	-43,059
-43,434	-43,19
-43,454	-43,195
-43,47	-43,262
-43,502	-43,288
-43,526	-43,356
-43,546	-43,37
-43,558	-43,392
-43,574	-43,404
-43,586	-43,441
-43,614	-43,481
-43,622	-43,544
-43,65	-43,549
-43,65	-43,554
-43,674	-43,555
-43,678	-43,575

-43,686	-43,614
-43,694	-43,636
-43,702	-43,643
-43,706	-43,722
-43,718	-43,782
-43,718	-43,804
-43,726	-43,813
-43,726	-43,816
-43,806	-43,818
-43,922	-43,827
-44,038	-43,83
-44,142	-43,863
-44,242	-43,873
-44,334	-43,877
-44,41	-43,885
-44,474	-43,953
-44,53	-43,963
-44,582	-43,977
-44,626	-43,998
-44,666	-44,095
-44,71	-44,144
-44,75	-44,165
-44,794	-44,24
-44,842	-44,361
-44,894	-44,456
-44,954	-44,491
-45,026	-44,649
-45,11	-44,76
-45,206	-45,116
-45,314	-45,857

## ANEXO B

### Reporte\_emisora.m

Algoritmo para generar el reporte de cada emisora con sus respectivos parámetros técnicos

```
function ruta =  
reporte_emisora(fPortadoraTeorica,medianfc,meanfc,stdfc,medianbw,meanbw,std  
bw,medianfieldint,meanfieldint,stdfieldint,n,notaberrant)  
VarNombreEmisora = {'Frecuencia Sintonizada [MHz]'};  
VarFreqCentral = {'Valores estadísticos para Frec. Central [MHz]'};  
VarValoresStd = {'Media', 'Mediana', 'Desviación'};  
VarFreqCElegida = {'Frec. Central Elegida o Error Relativo'};  
VarAnchoBanda = {'Valores estadísticos para Ancho de Banda [kHz]'};  
VarBWElegido = {'Ancho de Banda Elegido o Error Relativo'};  
VarIntensidadCampo = {'Valores e estadísticos para Intensidad de Campo  
[dBuV/m]'};  
VarICElegido = {'Intensidad de Campo Elegido o Error Relativo'};  
comment = {'Comentarios'};  
finalcomment = {'Comentario Final'};
```

```
fCentralTeoricaMax=fPortadoraTeorica+0.002; % Valor teorico maximo permitido  
por la norma tecnica Arcotel  
fCentralTeoricaMin=fPortadoraTeorica-0.002; %Valor teorico minimo permitido por  
la norma tecnica arcotel
```

```
e = actxserver('Excel.Application');  
  
eWorkbook = e.Workbooks.Add;  
e.Visible = 1;  
  
eSheets = e.ActiveWorkbook.Sheets;  
  
eSheet1 = eSheets.get('Item',1);  
eSheet1.Activate;
```

### %FORMATO REPORTE

```
eActivsheetRange = e.Activesheet.get('Range','D2');
eActivsheetRange.Value = fPortadoraTeorica;
eActivsheetRange = e.Activesheet.get('Range','A2:C2');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = VarNombreEmisora;
eActivsheetRange = e.Activesheet.get('Range','A3:C4');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = VarFreqCentral;
eActivsheetRange = e.Activesheet.get('Range','A5:C5');
eActivsheetRange.Value = VarValoresStd;
eActivsheetRange = e.Activesheet.get('Range','D3:D4');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = VarFreqCElegida;
eActivsheetRange = e.Activesheet.get('Range','E3:E4');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = comment;
eActivsheetRange = e.Activesheet.get('Range','A7:C8');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = VarAnchoBanda;
eActivsheetRange = e.Activesheet.get('Range','A9:C9');
eActivsheetRange.Value = VarValoresStd;
eActivsheetRange = e.Activesheet.get('Range','D7:D8');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = VarBWElegido;
eActivsheetRange = e.Activesheet.get('Range','E7:E8');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = comment;
eActivsheetRange = e.Activesheet.get('Range','A11:C12');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = VarIntensidadCampo;
eActivsheetRange = e.Activesheet.get('Range','A13:C13');
eActivsheetRange.Value = VarValoresStd;
eActivsheetRange = e.Activesheet.get('Range','D11:D12');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = VarICElegido;
eActivsheetRange = e.Activesheet.get('Range','E11:E12');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = comment;
eActivsheetRange = e.Activesheet.get('Range','A15:C15');
eActivsheetRange.MergeCells = 1;
eActivsheetRange.Value = finalcomment;
```

### %Imprime valores estadísticos en el reporte

```
eActivsheetRange = e.Activesheet.get('Range','A6');
eActivsheetRange.Value = meanfc;
```

```
eActivsheetRange = e.Activesheet.get('Range','B6');
eActivsheetRange.Value = medianfc;
```

```
eActivsheetRange = e.Activesheet.get('Range','C6');  
eActivsheetRange.Value = stdfc;
```

```
eActivsheetRange = e.Activesheet.get('Range','A10');  
eActivsheetRange.Value = meanbw;
```

```
eActivsheetRange = e.Activesheet.get('Range','B10');  
eActivsheetRange.Value = medianbw;
```

```
eActivsheetRange = e.Activesheet.get('Range','C10');  
eActivsheetRange.Value = stdbw;
```

```
eActivsheetRange = e.Activesheet.get('Range','A14');  
eActivsheetRange.Value = meanfieldint;
```

```
eActivsheetRange = e.Activesheet.get('Range','B14');  
eActivsheetRange.Value = medianfieldint;
```

```
eActivsheetRange = e.Activesheet.get('Range','C14');  
eActivsheetRange.Value = stdfieldint;
```

## %FORMATO REPORTE

### %Condicion norma tecnica

%intervalo de confianza 95 %

### % ANALISIS FRECUENCIA CENTRAL

Z1=1.64;

Z2=-1.64;

Homean1fc=0; % Hipotesis nula  $u \geq fc - 0.002$

Homean2fc=0; % Hipotesis nula  $u \leq fc + 0.002$

Homedian1fc=0;

Homedian2fc=0;

Zmean=(meanfc-fCentralTeoricaMin)/(stdfc/(sqrt(n))); %Z para media

Zmedian=(medianfc-fCentralTeoricaMin)/(stdfc/(sqrt(n))); %Z para mediana

```
if Zmean>Z2  
    Homean1fc=1;  
else  
    Homean1fc=0;  
end
```

```
if Zmedian>Z2  
    Homedian1fc=1;  
else  
    Homedian1fc=0;  
end
```

```

Zmean=(meanfc-fCentralTeoricaMax)/(stdfc/(sqrt(n)));
Zmedian=(medianfc-fCentralTeoricaMax)/(stdfc/(sqrt(n))); %Z para mediana

if Zmean<Z1
    Homean2fc=1;
else
    Homean2fc=0;
end

if Zmedian<Z1
    Homedian2fc=1;
else
    Homedian2fc=0;
end

if Homean1fc==1 && Homean2fc==1
    flagcumplemedia=1;
else
    flagcumplemedia=0;
end

if Homedian1fc==1 && Homedian2fc==1
    flagcumplemediana=1;
else
    flagcumplemediana=0;
end

if flagcumplemedia==1 && flagcumplemediana==1
    cumplefc=1;
    dif1=abs(fPortadoraTeorica-meanfc);
    dif2=abs(fPortadoraTeorica-medianfc);

    if dif1<dif2
        fcelegida=meanfc;

        eActivsheetRange = e.Activesheet.get('Range','D6');
        eActivsheetRange.Value = fcelegida;

        comment={'La frecuencia central cumple la norma tecnica tomando el valor
de la media'};
        eActivsheetRange = e.Activesheet.get('Range','E6');
        eActivsheetRange.Value = comment;
    else
        fcelegida=medianfc;

        eActivsheetRange = e.Activesheet.get('Range','D6');
        eActivsheetRange.Value = fcelegida;
    end
end

```

```
comment={'La frecuencia central cumple la norma tecnica tomando el valor de la mediana'};  
eActivsheetRange = e.Activesheet.get('Range','E6');  
eActivsheetRange.Value = comment;
```

```
end
```

```
elseif flagcumplemedia==1 && flagcumplemediana==0  
cumplefc=1;  
fcelegida=meanfc;  
eActivsheetRange = e.Activesheet.get('Range','D6');  
eActivsheetRange.Value = fcelegida;
```

```
comment={'La frecuencia central cumple la norma tecnica tomando el valor de la media'};  
eActivsheetRange = e.Activesheet.get('Range','E6');  
eActivsheetRange.Value = comment;
```

```
elseif flagcumplemedia==0 && flagcumplemediana==1  
cumplefc=1;  
fcelegida=medianfc;
```

```
eActivsheetRange = e.Activesheet.get('Range','D6');  
eActivsheetRange.Value = fcelegida;
```

```
comment={'La frecuencia central cumple la norma tecnica tomando el valor de la mediana'};  
eActivsheetRange = e.Activesheet.get('Range','E6');  
eActivsheetRange.Value = comment;
```

```
else
```

```
cumplefc=0;
```

```
dif1=abs(fPortadoraTeorica-meanfc);  
dif2=abs(fPortadoraTeorica-medianfc);
```

```
if dif1<dif2
```

```
fcelegida=meanfc;  
errp = abs((fPortadoraTeorica-fcelegida)/fPortadoraTeorica)*100;  
errorporcentual= ['Error de ',num2str(errp),'% respecto de la Frecuencia Central'];
```

```
eActivsheetRange = e.Activesheet.get('Range','D6');  
eActivsheetRange.Value = errorporcentual;
```

```
comment={['La frecuencia central no cumple con la norma técnica porque está desviada ',num2str(errp),' % respecto de la frecuencia sintonizada']};  
eActivsheetRange = e.Activesheet.get('Range','E6');  
eActivsheetRange.Value = comment;
```

```
else
```

```
fcelegida=medianfc;
```

```

    errp = abs((fPortadoraTeorica-fcelegida)/fPortadoraTeorica)*100;
    errorporcentual= ['Error de ',num2str(errp),'% respecto de la Frecuencia
Central'];
    eActivsheetRange = e.Activesheet.get('Range','D6');
    eActivsheetRange.Value = errorporcentual;

    comment=(['La frecuencia central no cumple con la norma técnica porque
está desviada ',num2str(errp),' % respecto de la frecuencia sintonizada']);
    eActivsheetRange = e.Activesheet.get('Range','E6');
    eActivsheetRange.Value = comment;

end

```

```
end
```

```
%ANALISIS ANCHO DE BANDA
```

```
Z1=1.64;
Z2=-1.64;
```

```
HoMeanBw=0; % Hipotesis nula con mediaBw  $u \leq 220$  kHz;
HoMedianBw=0; %Hipotesis nula con medianaBw  $u \leq 220$  kHz
```

```
ZmeanBw=(meanbw-220000)/(stdbw/(sqrt(n))); %Z para media
ZmedianBw=(medianbw-220000)/(stdbw/(sqrt(n))); %Z para mediana
```

```
if ZmeanBw<Z1
    HoMeanBw=1;
    flagcumplemedia=1;
```

```
else
    HoMeanBw=0;
    flagcumplemedia=0;
```

```
end
```

```
if ZmedianBw<Z1
    HoMedianBw=1;
    flagcumplemediana=1;
```

```
else
    HoMedianBw=0;
    flagcumplemediana=0;
```

```
end
```

```
if flagcumplemedia==1 && flagcumplemediana==1
```

```
    cumplebw=1;
```

```
    if meanbw>medianbw % Si ambas cumple la norma se toma el mayor valor
        bwcelegida=meanbw/1000;
```

```

eActivsheetRange = e.Activesheet.get('Range','D10');
eActivsheetRange.Value = bwcelegida;

comment={'El ancho de banda cumple la norma tecnica tomando el valor de
la media'};
eActivsheetRange = e.Activesheet.get('Range','E10');
eActivsheetRange.Value = comment;

else
bwcelegida=medianbw/1000;

eActivsheetRange = e.Activesheet.get('Range','D10');
eActivsheetRange.Value = bwcelegida;

comment={'El ancho de banda cumple la norma tecnica tomando el valor de
la mediana'};
eActivsheetRange = e.Activesheet.get('Range','E10');
eActivsheetRange.Value = comment;

end

elseif flagcumplemedia==1 && flagcumplemediana==0
cumplebw=1;
bwcelegida=meanbw/1000;
eActivsheetRange = e.Activesheet.get('Range','D10');
eActivsheetRange.Value = bwcelegida;

comment={'El ancho de banda cumple la norma tecnica tomando el valor de la
media'};
eActivsheetRange = e.Activesheet.get('Range','E10');
eActivsheetRange.Value = comment;

elseif flagcumplemedia==0 && flagcumplemediana==1
cumplebw=1;
bwcelegida=medianbw/1000;

eActivsheetRange = e.Activesheet.get('Range','D10');
eActivsheetRange.Value = bwcelegida;
comment={'El ancho de banda cumple la norma tecnica tomando el valor de la
mediana'};
eActivsheetRange = e.Activesheet.get('Range','E10');
eActivsheetRange.Value = comment;

else
cumplebw=0;

if meanbw>medianbw % Si no cumple la norma se toma el mayor valor
bwcelegida=meanbw/1000;

```

```

errp = abs((220-bwcelegida)/220)*100;
errorporcentualbw= ['Error de ',num2str(errp),'% respecto del ancho de banda
permitido'];
eActivsheetRange = e.Activesheet.get('Range','D10');
eActivsheetRange.Value = errorporcentualbw;

comment=(['El ancho de banda no cumple con la norma técnica porque está
desviada ',num2str(errp),' % respecto del ancho de banda permitido']);
eActivsheetRange = e.Activesheet.get('Range','E10');
eActivsheetRange.Value = comment;

```

```

else

```

```

    bwcelegida=medianbw/1000;

```

```

    errp = abs((220-bwcelegida)/220)*100;
    errorporcentualbw= ['Error de ',num2str(errp),'% respecto del ancho de banda
permitido'];
    eActivsheetRange = e.Activesheet.get('Range','D10');
    eActivsheetRange.Value = errorporcentualbw;

```

```

    comment=(['El ancho de banda no cumple con la norma técnica porque está
desviada ',num2str(errp),' % respecto del ancho de banda permitido']);
    eActivsheetRange = e.Activesheet.get('Range','E10');
    eActivsheetRange.Value = comment;

```

```

end

```

```

end

```

```

% ANALISIS INTENSIDAD DE CAMPO

```

```

Z1=1.64;
Z2=-1.64;

```

```

HoMeanIC=0; % Hipotesis nula con medialC  $u \leq 56$  dBu;
HoMedianIC=0; %Hipotesis nula con medianalC  $u \leq 56$  dBu

```

```

ZmeanIC=(meanfieldint-56)/(stdfieldint/(sqrt(n))); %Z para media
ZmedianIC=(medianfieldint-56)/(stdfieldint/(sqrt(n))); %Z para mediana

```

```

if ZmeanIC>Z2
    HoMeanIC=1;
    flagcumplemedia=1;

```

```

else
    HoMeanIC=0;
    flagcumplemedia=0;

```

```

end

```

```

if ZmedianIC>Z2
    HoMedianIC=1;
    flagcumplemediana=1;
else
    HoMedianIC=0;
    flagcumplemediana=0;
end

if flagcumplemedia==1 && flagcumplemediana==1

    cumpleic=1;
    if meanfieldint>medianfieldint % Si ambas cumple la norma se toma el mayor
valor
        icelegida=meanfieldint;

        eActivsheetRange = e.Activesheet.get('Range','D14');
        eActivsheetRange.Value = icelegida;

        comment={'La intensidad de campo cumple la norma tecnica tomando el
valor de la media'};
        eActivsheetRange = e.Activesheet.get('Range','E14');
        eActivsheetRange.Value = comment;

    else
        icelegida=medianfieldint;

        eActivsheetRange = e.Activesheet.get('Range','D14');
        eActivsheetRange.Value = icelegida;

        comment={'La intensidad de campo cumple la norma tecnica tomando el
valor de la mediana'};
        eActivsheetRange = e.Activesheet.get('Range','E14');
        eActivsheetRange.Value = comment;

    end

elseif flagcumplemedia==1 && flagcumplemediana==0
    cumpleic=1;
    icelegida=meanfieldint;
    eActivsheetRange = e.Activesheet.get('Range','D14');
    eActivsheetRange.Value = icelegida;
    comment={'La intensidad de campo cumple la norma tecnica tomando el valor
de la media'};
    eActivsheetRange = e.Activesheet.get('Range','E14');
    eActivsheetRange.Value = comment;

elseif flagcumplemedia==0 && flagcumplemediana==1

```

```
cumpleic=1;
icelegida=medianfieldint;
```

```
eActivsheetRange = e.Activesheet.get('Range','D14');
eActivsheetRange.Value = icelegida;
```

```
comment={'La intensidad de campo cumple la norma tecnica tomando el valor
de la mediana'};
```

```
eActivsheetRange = e.Activesheet.get('Range','E14');
eActivsheetRange.Value = comment;
```

```
else
```

```
  cumpleic=0;
```

```
  if meanfieldint>medianfieldint % Si no cumple la norma se toma el mayor valor
    icelegida=meanfieldint;
```

```
    errp = abs((54-icelegida)/54)*100;
    errorporcentualic= ['Error de ',num2str(errp),'% respecto a la intensidad de
campo permitido.'];
```

```
    eActivsheetRange = e.Activesheet.get('Range','D14');
    eActivsheetRange.Value = errorporcentualic;
```

```
    comment=(['La intensidad de campo no cumple con la norma técnica porque
está desviada ',num2str(errp),' % respecto de la intensidad de campo permitida']);
```

```
    eActivsheetRange = e.Activesheet.get('Range','E14');
    eActivsheetRange.Value = comment;
```

```
else
```

```
  icelegida=meanfieldint;
```

```
  errp = abs((54-icelegida)/54)*100;
  errorporcentualic= ['Error de ',num2str(errp),'% respecto a la intensidad de
campo permitido'];
```

```
  eActivsheetRange = e.Activesheet.get('Range','D14');
  eActivsheetRange.Value = errorporcentualic;
```

```
  comment=(['La intensidad de campo no cumple con la norma técnica porque
está desviada ',num2str(errp),' % respecto de la intensidad de campo permitida']);
```

```
  eActivsheetRange = e.Activesheet.get('Range','E14');
  eActivsheetRange.Value = comment;
```

```
end
```

```
end
```

```
if cumplefc==1 && cumplebw==1 && cumpleic==1
  comment={'La emisora cumple la norma tecnica'};
```

```

    eActivsheetRange = e.Activesheet.get('Range','A16');
    eActivsheetRange.Value = comment;
else
    comment={'La emisora no cumple la norma tecnica'};
    eActivsheetRange = e.Activesheet.get('Range','A16');
    eActivsheetRange.Value = comment;
end

```

```

if notaberrant == 0
    SaveAs(eWorkbook,['Reporte Emisora
',num2str(fPortadoraTeorica),'MHz.xlsx']);
else
    SaveAs(eWorkbook,['Reporte Emisora Not Aberrant
',num2str(fPortadoraTeorica),'MHz.xlsx']);

end
eWorkbook.Saved = 1;
Close(eWorkbook);
Quit(e);
delete(e);

ruta = ['Reporte Emisora ',num2str(fPortadoraTeorica),'MHz.xlsx'];
end

```

rtl\_singlesweep.m

Algoritmos para la adquisición de los datos del espectro para un solo canal. El mismo que puede ser configurad sus diversos parámetros como ganancia, tasa de muestreo, entre otros.

```

%RTL-SDR Spectrum Sweep
% - You can use this script to sweep and record the RF spectrum with your
%   RTL-SDR
% - Change the "location" parameter (line 22) to something that identifies
%   your location, eg Glasgow, New York or Sydney
% - You may change range that the RTL-SDR will sweep over by changing the
%   values of "start_freq" and "stop_freq" (lines 23 and 24)
% - If you wish, you can also change the RTL-SDR sampling rate by changing
%   "rtlsdr_fs", and the tuner gain by modifying "rtlsdr_gain" (lines 26
%   and 27)
% - At the end of the simulation, the recorded data will be processed and
%   plotted in a popup figure
% - This figure will be saved to the MATLAB 'current folder' for later
%   viewing
% - NOTE: to end simulation early, use |Ctrl| + |C|

```

```

function [ruta] = rtlsdr_rx_singlesweep (rtlsdr_tunerfreq, sim_time)

```

```

% PARAMETERS (can change)

```

```

location      = 'Guayaquil'; % location used for figure name
start_freq    = 88e6;        % sweep start frequency
stop_freq     = 108e6;       % sweep stop frequency
rtlsdr_id     = '0';        % RTL-SDR stick ID
rtlsdr_fs     = 0.95e6;     % RTL-SDR sampling rate in Hz
rtlsdr_gain   = 0;          % RTL-SDR tuner gain in dB
rtlsdr_frmLen = 16384;      % RTL-SDR output data frame size
rtlsdr_datatype = 'single'; % RTL-SDR output data type
rtlsdr_ppm    = 0;          % RTL-SDR tuner parts per million correction
%rtlsdr_tunerfreq = 90.7e6; % RTL-SDR tuner frequency (user may change)

```

```

% PARAMETERS (can change, but may break code)

```

```

nfrmhold      = 20;        % number of frames to receive
fft_hold      = 'avg';     % hold function "max" or "avg"
nfft          = 16384;     % number of points in FFTs (2^something)
dec_factor    = 16;        % output plot downsample
overlap       = 0.5;       % FFT overlap to counter rolloff
nfrmdump      = 0;        % number of frames to dump after retuning (to clear
buffer)

```

```

run_time=0;

```

```

%sim_time=10;

```

```

% CALCULATIONS

```

```

%rtlsdr_tunerfreq = start_freq:rtlsdr_fs*overlap:stop_freq; % range of tuner
frequency in Hz

```

```

%if ( max(rtlsdr_tunerfreq) < stop_freq ) % check the whole range is
covered, if not, add an extra tuner freq

```

```

    %rtlsdr_tunerfreq(length(rtlsdr_tunerfreq)+1) =
max(rtlsdr_tunerfreq)+rtlsdr_fs*overlap;

```

```

%end

```

```

%1 = length(rtlsdr_tunerfreq); % calculate number of retunes
required
freq_bin_width = (rtlsdr_fs/nfft); % create axis
freq_axis = (rtlsdr_tunerfreq-rtlsdr_fs/2*overlap : freq_bin_width*dec_factor :
(rtlsdr_tunerfreq+rtlsdr_fs/2*overlap)-freq_bin_width)/1e6;

% create spectrum figure
h_spectrum = create_spectrum;

%crea una tabla en la base de datos

variables=cell(1,length(freq_axis));
variables(1)={'Tiempo'};
x=length(freq_axis)+1;
for n=2:1:x
    variables(n)={'Frecuencia: ', num2str(n-1)};
end

frecuencias=[0 freq_axis];
xlswrite(['Data Tuner Frequency ',num2str(rtlsdr_tunerfreq/1e6),'
MHz.xlsx'],variables);
xlswrite(['Data Tuner Frequency ',num2str(rtlsdr_tunerfreq/1e6),'
MHz.xlsx'],frecuencias,'Hoja1','A2');

% run capture and plot
capture_and_plot;

% make spectrum visible
h_spectrum.fig.Visible = 'on';

% save data
filename =
['rtlsdr_rx_singlesweep_',num2str(start_freq/1e6),'MHz_',num2str(stop_freq/1e6),'
MHz_',location,'.fig'];
savefig(filename);

%% FUNCTION to create spectrum window
function h_spectrum = create_spectrum

% colours
h_spectrum.line_blue = [0.0000 0.4470 0.7410]; % spectrum analyzer blue
h_spectrum.line_orange = [1.0000 0.5490 0.0000]; % spectrum analyzer
orange
h_spectrum.window_grey = [0.95 0.95 0.95]; % background light grey
h_spectrum.axes_grey = [0.1 0.1 0.1]; % dark grey for axes titles etc
h_spectrum.plot_white = [1 1 1]; % white for plot background

```

```

% sizes
fig_w = 1200;
fig_h = 600;
scnsize = get(0,'ScreenSize');
if scnsize(3) < fig_w
    fig_w = scnsize(3);
end
if scnsize(4) < fig_h
    fig_h = scnsize(h);
end
fig_pos = [(scnsize(3)-fig_w)/2 (scnsize(4)-fig_h)/2 fig_w fig_h]; % set to
open in middle of monitor 1

% create new figure
h_spectrum.fig = figure(...
    'Color',h_spectrum.window_grey,...
    'Position',fig_pos,...
    'SizeChangedFcn',@resize_spectrum,...
    'Name',['RTL-SDR Spectrum Sweep: ',location],...
    'Visible', 'off');
h_spectrum.fig.Renderer = 'painters';

% subplot 1
h_spectrum.axes1 = axes(...
    'Parent',h_spectrum.fig,...
    'YGrid','on','YColor',h_spectrum.axes_grey,...
    'XGrid','on','XColor',h_spectrum.axes_grey,...
    'GridLineStyle','--',...
    'Color',h_spectrum.plot_white);
box(h_spectrum.axes1,'on');
hold(h_spectrum.axes1,'on');
xlabel(h_spectrum.axes1,'Frequency (MHz)');
ylabel(h_spectrum.axes1,'Power Ratio (dBm) [relative to 50 \Omega load] ');
xlim(h_spectrum.axes1,[(rtlsdr_tunerfreq-
rtlsdr_fs/2*overlap)/1e6,((rtlsdr_tunerfreq+rtlsdr_fs/2*overlap)-
freq_bin_width)/1e6]);
%xlim(h_spectrum.axes1,[freq_axis(1),freq_axis(end)]);

% subplot 2
h_spectrum.axes2 = axes(...
    'Parent',h_spectrum.fig,...
    'YGrid','on','YColor',h_spectrum.axes_grey,...
    'XGrid','on','XColor',h_spectrum.axes_grey,...
    'GridLineStyle','--',...
    'Color',h_spectrum.plot_white);
box(h_spectrum.axes2,'on');
hold(h_spectrum.axes2,'on');
xlabel(h_spectrum.axes2,'Frequency (MHz)');
ylabel(h_spectrum.axes2,'Relative Power (Watts)');

```

```

        xlim(h_spectrum.axes2,[(rtlsdr_tunerfreq-
rtlsdr_fs/2*overlap)/1e6,((rtlsdr_tunerfreq+rtlsdr_fs/2*overlap)-
freq_bin_width)/1e6]);

    % figure title
    title(h_spectrum.axes1,['RTL-SDR Spectrum Sweep || Range =
',num2str(start_freq/1e6),'MHz to ',...
        num2str(stop_freq/1e6),'MHz || Bin Width =
',num2str(freq_bin_width*dec_factor/1e3),...
        'kHz || Number of Bins = ',num2str(length(freq_axis)),'] || Number of
Retunes = ',...
        num2str(1)]);

    % position axes
    axes_position(fig_w,fig_h);

    % link plots together for zooming
    linkaxes([h_spectrum.axes1,h_spectrum.axes2],'x');

end

%% FUNCTION to calculate axes positions
function axes_position(fig_w,fig_h)

    h_spectrum.axes1.Position = [...    % dBm axes
        70/fig_w,...                % 70px from left
        (fig_h/2)/fig_h,...        % at centre line
        (fig_w-100)/fig_w,...      % 100px from right
        (fig_h/2-30)/fig_h];      % 80px from top

    h_spectrum.axes2.Position = [...    % Watts axes
        70/fig_w,...                % 70px from left
        50/fig_h,...                % 50px from bottom
        (fig_w-100)/fig_w,...      % 100px from right
        (fig_h/2-100)/fig_h];      % 100px below centre line

end

%% FUNCTION (callback) to resize axes in spectrum window
function resize_spectrum(hObject,callbackdata)

    % find current sizes
    fig_w = h_spectrum.fig.Position(3);
    fig_h = h_spectrum.fig.Position(4);

    % update axes positions
    axes_position(fig_w,fig_h);

end

```

```

%% FUNCTION to capture data from the RTL-SDR and plot it
function capture_and_plot

% START TIMER
tic;
disp(' ');

% SYSTEM OBJECTS
% RTL-SDR system object
obj_rtlsdr = comm.SDRRTLReceiver(...
    rtlsdr_id,...
    'CenterFrequency',    rtlsdr_tunerfreq(1),...
    'EnableTunerAGC',    false,...
    'TunerGain',          rtlsdr_gain,...
    'SampleRate',         rtlsdr_fs, ...
    'SamplesPerFrame',   rtlsdr_frmlen,...
    'OutputDataType',    rtlsdr_datatype ,...
    'FrequencyCorrection', rtlsdr_ppm );

% FIR decimator
obj_decntr = dsp.FIRDecimator(...
    'DecimationFactor',  dec_factor,...
    'Numerator',         fir1(300,1/dec_factor));

% CALCULATIONS (others)
rtlsdr_data_fft = zeros(1,nfft);           % fullsize matrix to hold calculated
fft [1 x nfft]
fft_reorder = zeros(length(nfrmhold),nfft*overlap); % matrix with overlap
compensation to hold re-ordered ffts [navg x nfft*overlap]
fft_dec = zeros(1,nfft*overlap/dec_factor); % matrix with overlap
compensation to hold all ffts [1 x nfft*overlap/data_decimate]

% SIMULATION
% check if RTL-SDR is active
if ~isempty(sdrinfo(obj_rtlsdr.RadioAddress))
else
    error(['RTL-SDR failure. Please check connection to ',...
        'MATLAB using the "sdrinfo" command.']);
end

% create progress variable
tune_progress = 0;

i=3;
tic;
% TIEMPO DE TOMA DE DATOS ELEGIDA POR EL USUARIO
while run_time<= sim_time

```

```

% for each of the tuner values

% tune RTL-SDR to new centre frequency
obj_rtlsdr.CenterFrequency = rtlsdr_tunerfreq;

% dump frames to clear software buffer
for frm = 1:1:nfrmdump
    % fetch a frame from the rtlsdr stick
    rtlsdr_data = step(obj_rtlsdr);
end

% display current centre frequency
disp(['          fc = ',num2str(rtlsdr_tunerfreq/1e6),'MHz']);

% loop for nfrmhold frames
for frm = 1:1:nfrmhold

    % fetch a frame from the rtlsdr stick
    rtlsdr_data = step(obj_rtlsdr);

    % remove DC component
    rtlsdr_data = rtlsdr_data - mean(rtlsdr_data);

    % find fft [ +ve , -ve ]
    rtlsdr_data_fft = abs(fft(rtlsdr_data,nfft));

    % rearrange fft [ -ve , +ve ] and keep only overlap data
    fft_reorder(frm,( 1 : (overlap*nfft/2) )) = rtlsdr_data_fft(
(overlap*nfft/2)+(nfft/2)+1 : end ); % -ve
    fft_reorder(frm,( (overlap*nfft/2)+1 : end )) = rtlsdr_data_fft( 1 :
(overlap*nfft/2) ); % +ve

end

% process the fft data down to [1 x nfft*overlap/data_decimate] from
[nfrmhold x nfft*overlap/data_decimate]
if strcmp(fft_hold,'avg')
    % if set to average, find mean
    fft_reorder_proc = mean(fft_reorder);

elseif strcmp(fft_hold,'max')
    % if set to max order hold, find max
    fft_reorder_proc = max(fft_reorder);

end

% decimate data to smooth and store in spectrum matrix
fft_dec(1,:) = step(obj_decntr,fft_reorder_proc)';

```

```

% show progress if at an n10% value
if floor(1*10/1) ~= tune_progress;
    tune_progress = floor(1*10/1);
    disp(['    progress = ',num2str(tune_progress*10),'%']);
end

% REORDER INTO ONE MATRIX
fft_masterreshape = reshape(fft_dec',1,1*nfft*overlap/dec_factor);

y_data = fft_masterreshape*1e-3;
y_data_dbm = 10*log10(((y_data.^2)/50))+30;

reg=[run_time y_data_dbm];
xlswrite(['Data Tuner Frequency ',num2str(rtlsdr_tunerfreq/1e6),'
MHz.xlsx'],reg,'Hoja1',['A',num2str(i)]);

i=i+1;

run_time=toc;
end
% PLOT DATA

[pot, fc]=findpeaks(y_data_dbm,freq_axis,'MinPeakDistance',0.19);

plot(h_spectrum.axes1,freq_axis,y_data_dbm,fc,pot,'o','Color',h_spectrum.line_blue,
'linewidth',1.25);

plot(h_spectrum.axes2,freq_axis,y_data,'Color',h_spectrum.line_orange,'linewidth'
,1.25);

% STOP TIMER
disp(' ');
disp(['    run time = ',num2str(toc),'s']);
disp(' ');

end

ruta = ['Data Tuner Frequency ',num2str(rtlsdr_tunerfreq/1e6),' MHz.xlsx'];

end

```

## reporte\_final.m

Algoritmo que permite generar un reporte con todos los parámetros técnicos para las emisoras que el usuario desee, el mismo que debe mostrar dichos resultados, si cumplen o no, de darse el caso de un incumplimiento se debe mostrar el porcentaje de desviación respecto a la norma técnica de la ARCOTEL.

```
function varargout = reporte_final(varargin)
```

```
% REPORTE_FINAL MATLAB code for reporte_final.fig
%   REPORTE_FINAL, by itself, creates a new REPORTE_FINAL or raises the
%   existing
%   singleton*.
%
%   H = REPORTE_FINAL returns the handle to a new REPORTE_FINAL or the
%   handle to
%   the existing singleton*.
%
%   REPORTE_FINAL('CALLBACK',hObject,eventData,handles,...) calls the
%   local
%   function named CALLBACK in REPORTE_FINAL.M with the given input
%   arguments.
%
%   REPORTE_FINAL('Property','Value',...) creates a new REPORTE_FINAL or
%   raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before reporte_final_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to reporte_final_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help reporte_final

% Last Modified by GUIDE v2.5 26-Aug-2018 21:36:52

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @reporte_final_OpeningFcn, ...
                  'gui_OutputFcn', @reporte_final_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
```

```

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before reporte_final is made visible.
function reporte_final_OpeningFcn(hObject, eventdata, handles, varargin)
global i x eWorkbookReporteFinal ruta_actual controlexcel;

    i=3;
    x = actxserver('Excel.Application');
    eWorkbookReporteFinal = x.Workbooks.Add;
    x.Visible = 1;
    ruta_actual = [];

fiec = imread('com-logo-fiec.jpg');
axes(handles.axes1);
image(fiec);
axis off;
gicom = imread('com-logo-gicom.jpg');
axes(handles.axes3);
image(gicom);
axis off;
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to reporte_final (see VARARGIN)

% Choose default command line output for reporte_final
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes reporte_final wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

```

```

function varargout = reporte_final_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Get default command line output from handles structure
varargout{1} = handles.output;

```

```

% --- 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 filename fullpathname pathname i;
filename = 0;

```

```

[filename pathname] = uigetfile({'Reporte Emisora Not Aberrant*.xlsx', 'Reporte
Emisora'});

```

```

if isequal(filename,0)
    set(handles.text3, 'String', 'Cargue un archivo válido');
    return

```

```

else
    fullpathname = strcat(pathname,filename);
    set(handles.text3, 'String', filename);
end

```

```

% --- Executes on button press in agregar_reportefinal.

```

```

function agregar_reportefinal_Callback(hObject, eventdata, handles)
% hObject handle to agregar_reportefinal (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

global filename fullpathname pathname i x eWorkbookReporteFinal ruta_actual;

```

```

if isequal(ruta_actual,filename)
    msgbox('El archivo ya ha sido agregado','Mensaje');
    return

```

```

end
if isequal(filename,0)
    set(handles.text3, 'String', 'Cargue un archivo válido');
    return

```

```

else
    set(handles.text3, 'String', 'Si carga');

    e = actxserver('Excel.Application');

```

```
eWorkbooks = e.Workbooks;
```

```
% Hacer Excel visible
```

```
e.Visible = 1;
```

```
%Abrir Excel de Reporte Emisora Cargado
```

```
WorkbookReporteEmisoraFinal = eWorkbooks.Open(fullpathname);
```

```
%Activar Sheet numero 1 del archivo Excel cargado
```

```
eSheets = e.ActiveWorkBook.Sheets;  
eSheet1 = eSheets.get('Item',1);  
eSheet1.Activate;
```

```
xSheets = x.ActiveWorkBook.Sheets;  
xSheet1 = xSheets.get('Item',1);  
xSheet1.Activate;
```

```
xActivesheetRange = x.Activesheet.get('Range','A1:A2');  
xActivesheetRange.MergeCells = 1;  
xActivesheetRange.Value = 'Frecuencia Autorizada [MHz]';  
xActivesheetRange.Select;  
invoke(x.Selection.Columns,'AutoFit');
```

```
xActivesheetRange = x.Activesheet.get('Range','B1:B2');  
xActivesheetRange.MergeCells = 1;  
xActivesheetRange.Value = 'Frecuencia Central [MHz]';  
xActivesheetRange.Select;  
invoke(x.Selection.Columns,'AutoFit');
```

```
xActivesheetRange = x.Activesheet.get('Range','C1:C2');  
xActivesheetRange.MergeCells = 1;  
xActivesheetRange.Value = 'Ancho de Banda [kHz]';  
xActivesheetRange.Select;  
invoke(x.Selection.Columns,'AutoFit');
```

```
xActivesheetRange = x.Activesheet.get('Range','D1:D2');  
xActivesheetRange.MergeCells = 1;  
xActivesheetRange.Value = 'Intensidad de Campo [dBuV/m]';  
xActivesheetRange.Select;  
invoke(x.Selection.Columns,'AutoFit');
```

```
xActiveshetRange = x.Activesheet.get('Range','E1:E2');
xActiveshetRange.MergeCells = 1;
xActiveshetRange.Value = 'Comentarios';
xActiveshetRange.Select;
invoke(x.Selection.Columns,'AutoFit');
```

```
xActiveshetRange = x.Activesheet.get('Range',['A',num2str(i)]);
xActiveshetRange.Value = e.Activesheet.get('Range','D2');
```

```
xActiveshetRange = x.Activesheet.get('Range',['B',num2str(i)]);
xActiveshetRange.Value = e.Activesheet.get('Range','D6');
xActiveshetRange.Select;
invoke(x.Selection.Columns,'AutoFit');
```

```
xActiveshetRange = x.Activesheet.get('Range',['C',num2str(i)]);
xActiveshetRange.Value = e.Activesheet.get('Range','D10');
```

```
xActiveshetRange = x.Activesheet.get('Range',['D',num2str(i)]);
xActiveshetRange.Value = e.Activesheet.get('Range','D14');
```

```
xActiveshetRange = x.Activesheet.get('Range',['E',num2str(i)]);
xActiveshetRange.Value = e.Activesheet.get('Range','A16');
xActiveshetRange.Select;
invoke(x.Selection.Columns,'AutoFit');
```

```
Close(eWorkbooks);
Quit(e);
delete(e);
```

```
cancelar=0;
if cancelar==0
set(handles.text4, 'String', num2str(i));
i=i+1;
end
ruta_actual = filename;
disp('Hola')
```

```
end
```

```
% --- Executes on button press in pushbutton3.
```

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

```
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global eWorkbookReporteFinal x
```

```
%date= datetime('now','Format','d-MMM-y HH:mm');
```

```

strdate = datestr(now,'m d, yyyy HH:MM AM');
d = strrep(strdate,':','h');
SaveAs(eWorkbookReporteFinal,['Reporte Final ',d,'.xlsx']);
eWorkbookReporteFinal.Saved = 1;
Close(eWorkbookReporteFinal);
Quit(x);
delete(x);
close(reporte_final);
ejemplo;

```

## ejemplo.m

Algoritmo cuya función permite crear la interfaz gráfica para el usuario, permite seleccionar la emisora y el tiempo a medir.

```
function varargout = ejemplo(varargin)
```

```

% EJEMPLO MATLAB code for ejemplo.fig
%   EJEMPLO, by itself, creates a new EJEMPLO or raises the existing
%   singleton*.
%
%   H = EJEMPLO returns the handle to a new EJEMPLO or the handle to
%   the existing singleton*.
%
%   EJEMPLO('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in EJEMPLO.M with the given input arguments.
%
%   EJEMPL O('Property','Value',...) creates a new EJEMPLO or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before ejemplo_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to ejemplo_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help ejemplo

% Last Modified by GUIDE v2.5 03-Sep-2018 21:18:13

% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...

```

```

        'gui_Singleton', gui_Singleton, ...
        'gui_OpeningFcn', @ejemplo_OpeningFcn, ...
        'gui_OutputFcn', @ejemplo_OutputFcn, ...
        'gui_LayoutFcn', [], ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});

end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});

end
% End initialization code - DO NOT EDIT

% --- Executes just before ejemplo is made visible.
function ejemplo_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to ejemplo (see VARARGIN)

% Choose default command line output for ejemplo

handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

fiec = imread('com-logo-fiec.jpg');
axes(handles.axes1);
image(fiec);
axis off;
gicom = imread('com-logo-gicom.jpg');
axes(handles.axes2);
image(gicom);
axis off;

% UIWAIT makes ejemplo wait for user response (see UIRESUME)
% uiwait(handles.figure1);

```

```

% --- Outputs from this function are returned to the command line.
function varargout = ejemplo_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

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)

% Hints: get(hObject,'String') returns contents of edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in medirbutton.
function medirbutton_Callback(hObject, eventdata, handles)
% hObject handle to medirbutton (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
emisoras = get(handles.popupmenu2, 'String');

simtime = get(handles.popupmenu3, 'String');

numemisora = get(handles.popupmenu2, 'Value');

```

```

numtime = get(handles.popupmenu3, 'Value');
sim_time = simtime(numtime);
obj_rtlsdr = comm.SDRRTLReceiver('0');
if ~isempty(sdrinfo(obj_rtlsdr.RadioAddress))
else

    msgbox('Fallo en el RTL-SDR. Por favor verifique la conexión usando el
comando en MATLAB "sdrinfo",'Error','error');
    %return;
end
rtlsdr_tunerfreq = emisoras(numemisoras);

```

```

guardardatos(rtlsdr_tunerfreq,sim_time);

```

```

close(ejemplo);

```

```

% --- Executes on selection change in popupmenu2.

```

```

function popupmenu2_Callback(hObject, eventdata, handles)

```

```

% hObject handle to popupmenu2 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2 contents as
cell array

```

```

% contents{get(hObject,'Value')} returns selected item from popupmenu2

```

```

% --- Executes during object creation, after setting all properties.

```

```

function popupmenu2_CreateFcn(hObject, eventdata, handles)

```

```

% hObject handle to popupmenu2 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.

```

```

% See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),

```

```

get(0,'defaultUicontrolBackgroundColor'))

```

```

    set(hObject,'BackgroundColor','white');

```

```

end

```

```

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

close(ejemplo);

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu3 contents as
cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu3

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function reporte_final_Callback(hObject, eventdata, handles)
% hObject    handle to reporte_final (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(ejemplo);
reporte_final;

% -----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% --- Executes during object creation, after setting all properties.
function text7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
msgbox('Esta aplicación permite analizar el espectro FM mediante la medición y
la caracterización de este, tomando en consideración parámetros técnicos;
frecuencia central, ancho de banda e intensidad de campo eléctrico; generando
un reporte que detalla que emisoras cumplen e incumplen la norma
técnica.', 'Bienvenido', 'Help');

```

### **guardardatos.m**

Algoritmo que permite guardar los datos para luego ser cargados cuando se genere el reporte.

```

function varargout = guardardatos(varargin)
% GUARDARDATOS MATLAB code for guardardatos.fig
%   GUARDARDATOS, by itself, creates a new GUARDARDATOS or raises the
existing
%   singleton*.
%
%   H = GUARDARDATOS returns the handle to a new GUARDARDATOS or
the handle to
%   the existing singleton*.
%
%   GUARDARDATOS('CALLBACK',hObject,eventData,handles,...) calls the
local
%   function named CALLBACK in GUARDARDATOS.M with the given input
arguments.
%
%   GUARDARDATOS('Property','Value',...) creates a new GUARDARDATOS or
raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before guardardatos_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to guardardatos_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help guardardatos

```

```
% Last Modified by GUIDE v2.5 05-Aug-2018 18:12:02
```

```
% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;  
gui_State = struct('gui_Name',    mfilename, ...  
                  'gui_Singleton', gui_Singleton, ...  
                  'gui_OpeningFcn', @guardardatos_OpeningFcn, ...  
                  'gui_OutputFcn', @guardardatos_OutputFcn, ...  
                  'gui_LayoutFcn', [], ...  
                  'gui_Callback', []);
```

```
if nargin && ischar(varargin{1})  
    gui_State.gui_Callback = str2func(varargin{1});  
end
```

```
if narginout  
    [varargout{1:narginout}] = gui_mainfcn(gui_State, varargin{:});  
else  
    gui_mainfcn(gui_State, varargin{:});  
end
```

```
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before guardardatos is made visible.
```

```
function guardardatos_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
% This function has no output args, see OutputFcn.
```

```
% hObject    handle to figure
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% varargin   command line arguments to guardardatos (see VARARGIN)
```

```
set(handles.tunerfreq, 'String', varargin{1});
```

```
set(handles.simtime, 'String', varargin{2});
```

```
%varargin{1} = handles.output;
```

```
%varargin{2} = handles.output;
```

```
% Choose default command line output for guardardatos
```

```
handles.output = hObject;
```

```
% Update handles structure
```

```
guidata(hObject, handles);
```

```
% UIWAIT makes guardardatos wait for user response (see UIRESUME)
```

```
% uiwait(handles.figure1);
```

```
fiec = imread('com-logo-fiec.jpg');
```

```
axes(handles.axes2);
```

```
image(fiec);
```

```
axis off;
```

```

gicom = imread('com-logo-gicom.jpg');
axes(handles.axes3);
image(gicom);
axis off;

% --- Outputs from this function are returned to the command line.
function varargout = guardardatos_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global rtlcdr_tunerfreq sim_time
% 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)
freqemisora = get(handles.tunerfreq, 'String');
tim = get(handles.simtime, 'String');
rtlsdr_tunerfreq = str2num(cell2mat(freqemisora))*1e6;
time = cell2mat(tim);
sim_time = str2num(time(1:2))*60;
msgbox('Por favor espere, se estan tomando los datos...','Midiendo...','help');
route = rtlcdr_rx_singlesweep(rtlsdr_tunerfreq,sim_time);

global ruta rutaemisora rutaemisoranotaberrant tuner_freq medianfc meanfc stdfc
frecuenciascentrales medianbw meanbw stdbw bws medianic meanic stdic
fieldintensity n...
    rutanotaberrant medianbwna meanbwna stdbwna bwsna...
    medianfcna meanfcna stdfcna frecuenciascentralesna nna...
    medianicna meanicna stdicna fieldintensityna

% Analisis de data con datos aberrantes
ruta=route;
tuner_freq = (regexp(ruta, '\S+(?=.)', 'match'));
tuner_freq = str2double(tuner_freq{4});
[medianfc, meanfc, stdfc, frecuenciascentrales, n] = median_fc(ruta,tuner_freq);
[medianbw, meanbw, stdbw, bws] = median_bw(ruta,tuner_freq);
[medianic, meanic, stdic, fieldintensity] = median_ic(ruta,tuner_freq);

rutaemisora=reporte_emisora(tuner_freq,medianfc,meanfc,stdfc,medianbw,meanb
w, stdbw, medianic, meanic, stdic, n, 0);
%normaldistr(meanfc, stdfc, frecuenciascentrales, 'frecuencia',
num2str(tuner_freq));

```

**%Análisis de data sin datos aberrantes**

```
[rutanotaberrant, medianbwna, meanbwna, stdbwna, bwsna] =  
data_no_aberrante(ruta,tuner_freq);  
[medianfcna, meanfcna, stdfcna, frecuenciascentralesna, nna] =  
median_fc(rutanotaberrant,tuner_freq);  
[medianicna, meanicna, stdicna, fieldintensityna] =  
median_ic(rutanotaberrant,tuner_freq);
```

```
rutaemisoranotaberrant =  
reporte_emisora(tuner_freq,medianfcna,meanfcna,stdfcna,medianbwna,meanbwn  
a,stdlibbwna,medianicna,meanicna,stdlibcna,nna,1);  
close(guardardatos(rtlsdr_tunerfreq,sim_time));  
generar_reporte;
```

**% --- Executes on button press in pushbutton2.**

```
function pushbutton2_Callback(hObject, eventdata, handles)  
global rtlsdr_tunerfreq sim_time  
% 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(guardardatos(rtlsdr_tunerfreq,sim_time));  
ejemplo;
```

### **normaldistr.m**

Algoritmo que genera imágenes del comportamiento del conjunto de datos, pretende demostrar que se obtiene una campana de Gauss cuando se estandarizan, este proceso se realiza para cada parámetro técnico.

```
function normaldistr(mean, std, datos, texto, tuner_freq)
```

```
z=(datos-mean)./std;
```

```
c=sort(z,'ascend');
```

```
y=normpdf(c,0,1);
```

```

figure;
plot(c,y);

if (strcmp(texto,'frecuencia'))
    title(['Distribución Normal Frecuencia Central: ',tuner_freq,' MHz'],'FontSize',
    12);
elseif (strcmp(texto,'ancho de banda'))
    title('Distribución Normal para Ancho de Banda','FontSize', 12);

else
    title('Distribución Normal para Intensidad de Campo','FontSize', 12);

end
end

```

### **median\_bw**

Algoritmo que permite obtener la media, mediana y desviación estándar del ancho de banda.

```
function [medianbw, meanbw, stdbw, bws] = median_bw(ruta,tuner_freq)
```

```

%ruta = 'Data Tuner Frequency 88.1 MHz.xlsx';
datos=xlsread(ruta);
%tunerfreq=88.1*1e6;
tunerfreq=tuner_freq*1e6;
x=tunerfreq-0.03e6;
y=tunerfreq+0.03e6;
freq_axis=(datos(1,(2:end)))*1e6;
bws=[];
s = size(datos);

```

```
for i=2:1:s(1)
```

```

    pxx_dbm=datos(i,(2:end));
    pxx_db=pxx_dbm-30;
    pxx=10.^(pxx_db/10);
    %[pot, posfc]=max(pxx_dbm);
    %fc=freq_axis(posfc);

```

```

    bw=powerbw(pxx,freq_axis,[x y]);
    %plot(freq_axis,pxx_dbm);
    bws=[bws bw];
    %z=1;

```

end

```
b=sort(bws,'ascend');  
meanbw=mean(b,'omitnan');  
medianbw=median(b,'omitnan');  
stdbw=std(b,'omitnan');
```

end

### Median\_fc.m

Algoritmo que permite obtener la media, mediana y desviación estándar para la frecuencia central.

```
function [medianfc, meanfc, stdfc, frecuenciascentrales, n] =  
median_fc(ruta,tuner_freq)
```

```
datos=xlsread(ruta);  
%tuner_freq=106.1;  
freq_axis=(datos(1,(2:end)));  
frecuenciascentrales=[];  
s=size(datos);
```

```
for i=2:1:s(1)
```

```
    pxx_dbm=datos(i,(2:end));
```

```
    %pcentral=find(abs(freq_axis-tuner_freq)<0.001); %256  
    pcentral = 256;  
    %freqaxisround=round(freq_axis,2);  
    %liminf=tuner_freq-0.1;  
    %panterior=find(abs(freqaxisround-liminf)<0.001);  
    %pant=panterior(1); %144  
    pant = 144;
```

```
    %limsup=tuner_freq+0.1;  
    %psiguiente=find(abs(freqaxisround-limsup)<0.001);  
    %psig=psiguiente(end); %370  
    psig = 370;
```

```
sub_arreglo_pot1=(pxx_dbm(1,(pant:pcentral)));
sub_arreglo_pot2=(pxx_dbm(1,(pcentral:psig)));
```

```
value1=min(sub_arreglo_pot1); %posicion del menor valor en el sub arreglo
```

```
value2=min(sub_arreglo_pot2); % posicion del mayor valor en el sub arreglo
```

```
pos1=find(pxx_dbm==value1);
pos2=find(pxx_dbm==value2);
```

```
freq_axis_emisora=freq_axis(1,(pos1:pos2));
sub_arreglo_pxx=pxx_dbm(1,(pos1:pos2));
```

```
[pot, posfc]=max(sub_arreglo_pxx);
fc=freq_axis_emisora(posfc);
frecuenciascentrales=[frecuenciascentrales fc];
```

```
clear pcentral;
end
```

```
a=sort(frecuenciascentrales,'ascend');
meanfc=mean(a,'omitnan');
medianfc=median(a,'omitnan');
stdfc=std(a,'omitnan');
```

```
n=s(1);
```

```
end
```

## median\_ic

Algoritmo que permite obtener la media, mediana y desviación estándar para la intensidad de campo eléctrico.

```
function [medianic, meanic, stdic, fieldintensity] = median_ic(ruta,tuner_freq)
```

```
datos=xlsread(ruta);  
freq_axis=(datos(1,(2:end)));
```

```
s=size(datos);  
fieldintensity=[];
```

```
for i=2:1:s(1)
```

```
    pxx_dbm=datos(i,(2:end));
```

```
    %pcentral=find(freq_axis==tuner_freq);  
    pcentral = 256;  
    %freqaxisround=round(freq_axis,2);  
    %liminf=tuner_freq-0.1;  
    %panterior=find(abs(freqaxisround-liminf)<0.001);  
    %pant=panterior(1);  
    pant = 144;
```

```
    %limsup=tuner_freq+0.1;  
    %psiguiente=find(abs(freqaxisround-limsup)<0.001);  
    %psig=psiguiente(end);  
    psig = 370;
```

```
    sub_arreglo_pot1=(pxx_dbm(1,(pant:pcentral)));  
    sub_arreglo_pot2=(pxx_dbm(1,(pcentral:psig)));
```

```
    value1=min(sub_arreglo_pot1); %posicion del menor valor en el sub arreglo
```

```
    value2=min(sub_arreglo_pot2); % posicion del mayor valor en el sub arreglo
```

```
    pos1=find(pxx_dbm==value1);  
    pos2=find(pxx_dbm==value2);
```

```
    freq_axis_emisora=freq_axis(1,(pos1:pos2));  
    sub_arreglo_pxx=pxx_dbm(1,(pos1:pos2));
```

```
[pot, posfc]=max(sub_arreglo_pxx);
```

```

fc=freq_axis_emisora(posfc);
dBu=pot+(20*log10(fc))+77.2-6;

fieldintensity=[fieldintensity dBu];

%plot(freq_axis_emisora,sub_arreglo_pxx);

```

end

```

a=sort(fieldintensity);
medianic=median(a,'omitnan');
meanic=mean(a,'omitnan');
stdic=std(a,'omitnan');

```

end

## Anexo C

```

%RTL-SDR Spectrum Sweep
% - You can use this script to sweep and record the RF spectrum with your
%   RTL-SDR
% - Change the "location" parameter (line 22) to something that identifies
%   your location, eg Glasgow, New York or Sydney
% - You may change range that the RTL-SDR will sweep over by changing the
%   values of "start_freq" and "stop_freq" (lines 23 and 24)
% - If you wish, you can also change the RTL-SDR sampling rate by changing
%   "rtlsdr_fs", and the tuner gain by modifying "rtlsdr_gain" (lines 26
%   and 27)
% - At the end of the simulation, the recorded data will be processed and
%   plotted in a popup figure
% - This figure will be saved to the MATLAB 'current folder' for later
%   viewing
% - NOTE: to end simulation early, use |Ctrl| + |C|

```

**function** rtlsdr\_rx\_specsweep

```

% PARAMETERS (can change)
location      = 'Guayaquil'; % location used for figure name
start_freq    = 88e6;        % sweep start frequency
stop_freq     = 108e6;       % sweep stop frequency
rtlsdr_id     = '0';         % RTL-SDR stick ID
rtlsdr_fs     = 1.3e6;       % RTL-SDR sampling rate in Hz

```

```

rtlsdr_gain      = 20;          % RTL-SDR tuner gain in dB
rtlsdr_frmrlen  = 4096;        % RTL-SDR output data frame size
rtlsdr_datatype = 'single';    % RTL-SDR output data type
rtlsdr_ppm      = 0;          % RTL-SDR tuner parts per million correction
% PARAMETERS (can change, but may break code)
nfrmhold        = 20;          % number of frames to receive
fft_hold        = 'avg';       % hold function "max" or "avg"
nfft            = 16384;       % number of points in FFTs (2^something)
dec_factor      = 32;          % output plot downsample
overlap         = 0.5;         % FFT overlap to counter rolloff
nfrmdump        = 100;         % number of frames to dump after retuning (to clear
buffer)
run_time=0;
sim_time=10;
% CALCULATIONS
rtlsdr_tunerfreq = start_freq:rtlsdr_fs*overlap:stop_freq; % range of tuner
frequency in Hz
if( max(rtlsdr_tunerfreq) < stop_freq ) % check the whole range is
covered, if not, add an extra tuner freq
    rtlsdr_tunerfreq(length(rtlsdr_tunerfreq)+1) =
max(rtlsdr_tunerfreq)+rtlsdr_fs*overlap;
end
nretunes = length(rtlsdr_tunerfreq); % calculate number of
retunes required
freq_bin_width = (rtlsdr_fs/nfft); % create axis
freq_axis = (rtlsdr_tunerfreq(1)-rtlsdr_fs/2*overlap : freq_bin_width*dec_factor :
(rtlsdr_tunerfreq(end)+rtlsdr_fs/2*overlap)-freq_bin_width)/1e6;

% create spectrum figure
h_spectrum = create_spectrum;

% run capture and plot
capture_and_plot;

% make spectrum visible
h_spectrum.fig.Visible = 'on';

% save data
filename =
['rtlsdr_rx_specsweep_',num2str(start_freq/1e6),'MHz_',num2str(stop_freq/1e6),'M
Hz_',location, '.fig'];
savefig(filename);

%% FUNCTION to create spectrum window
function h_spectrum = create_spectrum

% colours
h_spectrum.line_blue = [0.0000 0.4470 0.7410]; % spectrum analyzer blue

```

```

h_spectrum.line_orange = [1.0000 0.5490 0.0000]; % spectrum analyzer
orange
h_spectrum.window_grey = [0.95 0.95 0.95]; % background light grey
h_spectrum.axes_grey = [0.1 0.1 0.1]; % dark grey for axes titles etc
h_spectrum.plot_white = [1 1 1]; % white for plot background

% sizes
fig_w = 1200;
fig_h = 600;
scnsize = get(0,'ScreenSize'); % find monitor 1 size
if scnsize(3) < fig_w % if monitor is not fig_w wide
    fig_w = scnsize(3); % reduce fig_w
end
if scnsize(4) < fig_h % if monitor is not fig_h tall
    fig_h = scnsize(h); % reduce fig_h
end
fig_pos = [(scnsize(3)-fig_w)/2 (scnsize(4)-fig_h)/2 fig_w fig_h]; % set to
open in middle of monitor 1

% create new figure
h_spectrum.fig = figure(...
    'Color',h_spectrum.window_grey,...
    'Position',fig_pos,...
    'SizeChangedFcn',@resize_spectrum,...
    'Name',['RTL-SDR Spectrum Sweep: ',location],...
    'Visible', 'off');
h_spectrum.fig.Renderer = 'painters';

% subplot 1
h_spectrum.axes1 = axes(...
    'Parent',h_spectrum.fig,...
    'YGrid','on','YColor',h_spectrum.axes_grey,...
    'XGrid','on','XColor',h_spectrum.axes_grey,...
    'GridLineStyle','--',...
    'Color',h_spectrum.plot_white);
box(h_spectrum.axes1,'on');
hold(h_spectrum.axes1,'on');
xlabel(h_spectrum.axes1,'Frequency (MHz)');
ylabel(h_spectrum.axes1,'Power Ratio (dBm) [relative to 50 \Omega load] ');
xlim(h_spectrum.axes1,[start_freq/1e6,stop_freq/1e6]);
%xlim(h_spectrum.axes1,[freq_axis(1),freq_axis(end)]);

% subplot 2
h_spectrum.axes2 = axes(...
    'Parent',h_spectrum.fig,...
    'YGrid','on','YColor',h_spectrum.axes_grey,...
    'XGrid','on','XColor',h_spectrum.axes_grey,...
    'GridLineStyle','--',...
    'Color',h_spectrum.plot_white);
box(h_spectrum.axes2,'on');

```

```

hold(h_spectrum.axes2,'on');
xlabel(h_spectrum.axes2,'Frequency (MHz)');
ylabel(h_spectrum.axes2,'Relative Power (Watts)');
xlim(h_spectrum.axes2,[start_freq/1e6,stop_freq/1e6]);

% figure title
title(h_spectrum.axes1,['RTL-SDR Spectrum Sweep || Range =
',num2str(start_freq/1e6),'MHz to ',...
num2str(stop_freq/1e6),'MHz || Bin Width =
',num2str(freq_bin_width*dec_factor/1e3),...
'kHz || Number of Bins = ',num2str(length(freq_axis)), ' || Number of
Retunes = ',...
num2str(nretunes)]);

% position axes
axes_position(fig_w,fig_h);

% link plots together for zooming
linkaxes([h_spectrum.axes1,h_spectrum.axes2],'x');

end

%% FUNCTION to calculate axes positions
function axes_position(fig_w,fig_h)

h_spectrum.axes1.Position = [... % dBm axes
70/fig_w,... % 70px from left
(fig_h/2)/fig_h,... % at centre line
(fig_w-100)/fig_w,... % 100px from right
(fig_h/2-30)/fig_h]; % 80px from top

h_spectrum.axes2.Position = [... % Watts axes
70/fig_w,... % 70px from left
50/fig_h,... % 50px from bottom
(fig_w-100)/fig_w,... % 100px from right
(fig_h/2-100)/fig_h]; % 100px below centre line

end

%% FUNCTION (callback) to resize axes in spectrum window
function resize_spectrum(hObject,callbackdata)

% find current sizes
fig_w = h_spectrum.fig.Position(3);
fig_h = h_spectrum.fig.Position(4);

% update axes positions
axes_position(fig_w,fig_h);

```

end

%% FUNCTION to capture data from the RTL-SDR and plot it  
function capture\_and\_plot

% START TIMER

tic;  
disp(' ');

% SYSTEM OBJECTS

% RTL-SDR system object

obj\_rtlsdr = comm.SDRRTLReceiver(...  
 rtlsdr\_id,...  
 'CenterFrequency', rtlsdr\_tunerfreq(1),...  
 'EnableTunerAGC', false,...  
 'TunerGain', rtlsdr\_gain,...  
 'SampleRate', rtlsdr\_fs, ...  
 'SamplesPerFrame', rtlsdr\_frmrlen,...  
 'OutputDataType', rtlsdr\_datatype ,...  
 'FrequencyCorrection', rtlsdr\_ppm );

% FIR decimator

obj\_decntr = dsp.FIRDecimator(...  
 'DecimationFactor', dec\_factor,...  
 'Numerator', fir1(300,1/dec\_factor));

% CALCULATIONS (others)

rtlsdr\_data\_fft = zeros(1,nfft); % fullsize matrix to hold calculated  
fft [1 x nfft]  
fft\_reorder = zeros(length(nfrmhold),nfft\*overlap); % matrix with overlap  
compensation to hold re-ordered ffts [navg x nfft\*overlap]  
fft\_dec = zeros(nretunes,nfft\*overlap/dec\_factor); % matrix with overlap  
compensation to hold all ffts [ntune x nfft\*overlap/data\_decimate]

% SIMULATION

% check if RTL-SDR is active

if ~isempty(sdrinfo(obj\_rtlsdr.RadioAddress))  
else  
 error(['RTL-SDR failure. Please check connection to ',...  
 'MATLAB using the "sdrinfo" command.']);  
end

% create progress variable

tune\_progress = 0;

% for each of the tuner values

for ntune = 1:1:nretunes;

```

% tune RTL-SDR to new centre frequency
obj_rtlsdr.CenterFrequency = rtlsdr_tunerfreq(ntune);

% dump frames to clear software buffer
for frm = 1:1:nfrmdump
    % fetch a frame from the rtlsdr stick
    rtlsdr_data = step(obj_rtlsdr);
end

% display current centre frequency
disp(['          fc = ',num2str(rtlsdr_tunerfreq(ntune)/1e6),'MHz']);

% loop for nfrmhold frames
for frm = 1:1:nfrmhold

    % fetch a frame from the rtlsdr stick
    rtlsdr_data = step(obj_rtlsdr);

    % remove DC component
    rtlsdr_data = rtlsdr_data - mean(rtlsdr_data);

    % find fft [ +ve , -ve ]
    rtlsdr_data_fft = abs(fft(rtlsdr_data,nfft));

    % rearrange fft [ -ve , +ve ] and keep only overlap data
    fft_reorder(frm,( 1 : (overlap*nfft/2) )) = rtlsdr_data_fft(
(overlap*nfft/2)+(nfft/2)+1 : end ); % -ve
    fft_reorder(frm,( (overlap*nfft/2)+1 : end )) = rtlsdr_data_fft( 1 :
(overlap*nfft/2) ); % +ve

end

% process the fft data down to [1 x nfft*overlap/data_decimate] from
[nfrmhold x nfft*overlap/data_decimate]
if strcmp(fft_hold,'avg')
    % if set to average, find mean
    fft_reorder_proc = mean(fft_reorder);

elseif strcmp(fft_hold,'max')
    % if set to max order hold, find max
    fft_reorder_proc = max(fft_reorder);

end

% decimate data to smooth and store in spectrum matrix
fft_dec(ntune,:) = step(obj_decmtr,fft_reorder_proc);

% show progress if at an n10% value
if floor(ntune*10/nretunes) ~= tune_progress;
    tune_progress = floor(ntune*10/nretunes);

```

```

        disp(['    progress = ',num2str(tune_progress*10),'%']);
    end

end

% REORDER INTO ONE MATRIX
fft_masterreshape = reshape(fft_dec',1,ntune*nfft*overlap/dec_factor);

y_data = fft_masterreshape;
y_data_dbm = 10*log10((fft_masterreshape.^2)/50000);

% PLOT DATA

plot(h_spectrum.axes1,freq_axis,y_data_dbm,'Color',h_spectrum.line_blue,'linewidth',1.25);

plot(h_spectrum.axes2,freq_axis,y_data,'Color',h_spectrum.line_orange,'linewidth',1.25);

% STOP TIMER
disp(' ');
disp(['    run time = ',num2str(toc),'s']);
disp(' ');

end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Software, Simulation Examples and Design Exercises Licence Agreement
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This license agreement refers to the simulation examples, design
% exercises and files, and associated software MATLAB and Simulink
% resources that accompany the book:
%
% Title: Software Defined Radio using MATLAB & Simulink and the RTL-SDR

```

% Published by Strathclyde Academic Media, 2015  
% Authored by Robert W. Stewart, Kenneth W. Barlee, Dale S.W. Atkinson,  
% and Louise H. Crockett  
%  
% and made available as a download from [www.desktopSDR.com](http://www.desktopSDR.com) or variously  
% acquired by other means such as via USB storage, cloud storage, disk or  
% any other electronic or optical or magnetic storage mechanism. These  
% files and associated software may be used subject to the terms of  
% agreement of the conditions below:  
%  
% Copyright © 2015 Robert W. Stewart, Kenneth W. Barlee,  
% Dale S.W. Atkinson, and Louise H. Crockett. All rights reserved.  
%  
% Redistribution and use in source and binary forms, with or without  
% modification, are permitted provided that the following conditions are  
% met:  
%  
% (1) Redistributions of source code must retain the above copyright  
% notice, this list of conditions and the following disclaimer.  
%  
% (2) Redistributions in binary form must reproduce the above copyright  
% notice, this list of conditions and the following disclaimer in  
% the documentation and/or other materials provided with the  
% distribution.  
%  
% (3) Neither the name of the copyright holder nor the names of its  
% contributors may be used to endorse or promote products derived  
% from this software without specific prior written permission.  
%  
% (4) In all cases, the software is, and all modifications and  
% derivatives of the software shall be, licensed to you solely for  
% use in conjunction with The MathWorks, Inc. products and service  
% offerings.  
%  
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND  
% CONTRIBUTORS  
% "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT  
% NOT  
% LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND  
% FITNESS FOR  
% A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE  
% COPYRIGHT  
% HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,  
% INCIDENTAL,  
% SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,  
% BUT NOT  
% LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
% LOSS OF USE,  
% DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED  
% AND ON ANY

