

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

Modelamiento de la banda 6 GHz en la ciudad de Guayaquil

PROYECTO INTEGRADOR

Previo a la obtención del Título de:

INGENIERO EN TELECOMUNICACIONES

Autores:

Renatto Carlo Centanaro Vega

Dixon Henry León Masache

GUAYAQUIL-ECUADOR

Año: 2022

DEDICATORIA

A mis padres Renatto Centanaro Vela y Verónica Vega Moreira que me han dado el apoyo incondicional durante esta etapa de mi vida.

A mis hermanas Nicolle y Vanny por alentarme y confiar en mí.

Renatto Carlo Centanaro Vega.

A mis padres, Holanda y Dixon, quienes nunca bajaron los brazos y sirvieron de guía y camino para estar donde estoy.

A mi mejor amigo, que lamentablemente no podrá acompañarme en esta etapa.

Dixon León M.

AGRADECIMIENTOS

Agradezco a mis padres por confiar en mí, por inculcarme principios, enseñanzas, y por sobre todo el consejo que nunca me ha faltado.

Agradezco a mis hermanas por ser mi apoyo y motivación incluso en los momentos difíciles.

Agradezco a mis docentes y tutores por toda la enseñanza y conocimiento impartido en esta etapa fundamental de mi vida, en especial a mi profesor de tesis, el PhD. Francisco Novillo, por la paciencia, y direccionarme en el desarrollo de este proyecto.

Agradezco a mis familiares, amigos y compañeros que estuvieron a lo largo de esta fase.

Renatto Carlo Centanaro Vega

Agradezco a mis padres, los mayores promotores en este proceso, por brindarme invaluable enseñanzas a través de los años y que servirán para toda la vida.

Agradezco a mis formadores, docentes y tutores, por todo el conocimiento y la paciencia invertida, en especial a mi profesor de tesis, el PhD. Francisco Novillo, por la constante retroalimentación a lo largo del proyecto.

Agradezco a mis amigos, quienes me brindaron su compañía en momentos complicados a lo largo de la carrera.

Dixon León M.

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Renatto Carlo Centanaro Vega y Dixon Henry León Masache 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”



Renatto Carlo Centanaro Vega

Dixon Henry León Masache

EVALUADORES



PhD. Francisco Vicente Novillo Parales

PROFESOR DE LA MATERIA

MSc. Alfredo José Núñez Unda

PROFESOR TUTOR

ABREVIATURAS

ARCOTEL:	Agencia de Regulación y Control de las Telecomunicaciones
ASCII:	American Standard Code for Information Interchange
FCC:	Federal Communications Commission
GPIB:	General Purpose Interface Bus
IEEE:	Institute of Electrical and Electronics Engineers
ISM:	Industrial, Scientific and Medical
ITU:	International Telecommunication Union
MSE:	Mean Squared Error
NLOS:	Non-line of Sight
OLOS:	Obstructed Line of Sight
RMSE:	Root Mean Squared Error
SCPI:	SAP Cloud Platform Integration
VISA:	Virtual Instrument Software Architecture
VNA:	Vector Network Analyzer
WiFi:	Wireless Fidelity
WLAN:	Wireless Local Area Network

RESUMEN

El presente trabajo propone un modelo empírico de pérdidas de propagación para el segmento bajo de la banda de 6 GHz desarrollado exclusivamente en Python. Se efectuó una caracterización experimental del comportamiento del canal inalámbrico acorde a las campañas de medición llevadas a cabo en un laboratorio. Las mediciones fueron realizadas entre la frecuencia 5.8 GHz y 6.4 GHz en condiciones de línea de vista durante tres días en horario laboral. Los resultados muestran que el modelo obtenido posee un coeficiente de determinación de 0.92; los coeficientes β_0 y β_1 hallados fueron de 53.2251 y 2.2197 respectivamente. Cabe mencionar que los coeficientes del modelo fueron obtenidos mediante técnicas de regresión lineal sobre los datos de medición usando dos scripts desarrollados en Python. Por otra parte, en términos del error cuadrático medio, los modelos semi-empíricos de la literatura difieren significativamente de los valores reales obtenidos. Las pérdidas de propagación a estas frecuencias en entornos similares se pueden describir con el modelo obtenido, sin embargo, se considera necesario realizar más campañas de medición con el fin de mejorar la base datos y disminuir el sobredimensionamiento o sub-dimensionamiento en futuros despliegues de WLANs.

Palabras claves: modelo empírico, canal inalámbrico, pérdidas de propagación, Python, regresión lineal.

ABSTRACT

This work proposes an empirical propagation loss model for the low segment of the 6 GHz band developed exclusively in Python. An experimental characterization of the behavior of the wireless channel was performed according to the measurement campaigns carried out in a laboratory. The measurements were performed between the 5.8 GHz and 6.4 GHz frequencies in line-of-sight conditions during three days during working hours. The results show that the model obtained has a coefficient of determination of 0.92; the coefficients found were 53.2251 and 2.2197 respectively. It is worth mentioning that the model coefficients were obtained by linear regression techniques on the measurement data using two scripts developed in Python. On the other hand, in terms of the mean square error, the semi-empirical models in the literature differ significantly from the actual values obtained. Propagation losses at these frequencies in similar environments can be described with the obtained model, however, it is considered necessary to perform more measurement campaigns in order to improve the database and decrease over- or undersizing in future deployments of WLANs.

Key words: empirical model, wireless channel, propagation loss, Python, linear regression.

ÍNDICE

DEDICATORIA	I
AGRADECIMIENTOS	III
DECLARACIÓN EXPRESA	V
EVALUADORES	VI
RESUMEN	VIII
ABSTRACT	IX
ÍNDICE	X
CAPÍTULO 1	15
1. Introducción	15
1.1. Definición del Problema	16
1.2. Justificación del Problema	16
1.3. Objetivo General	17
1.4. Propuesta de Solución	18
1.5. Metodología	18
CAPÍTULO 2	19
2. Marco Teórico	19
2.1. Bandas ISM	19
2.2. Modelos de propagación para interiores	19
2.3. Estándar IEEE 488.2	24
CAPÍTULO 3	25
3. Desarrollo de la solución	25
3.1. Equipo Utilizado	25
3.2. Escenario y Ruta de Medición	26
3.3. Procedimiento de Medición	28
3.4. Captura y Procesamiento de Mediciones	29
3.5. Pruebas realizadas	31
3.5.1. Experimento 1	33
3.5.2. Experimento 2	34
CAPÍTULO 4	36
4. Resultados	36

4.1. Pérdidas de Propagación Medidas	36
4.2. Desarrollo de Modelo de Propagación	39
4.3. Comparación de Resultados	41
CAPÍTULO 5	44
5. Conclusiones y Recomendaciones	44
5.1. Conclusiones.....	44
5.2. Recomendaciones	45
BIBLIOGRAFÍA	46
ANEXOS	48

ÍNDICE DE TABLAS

Tabla 3.1. Parámetros del esquema de medición	26
Tabla 4.1. RMSE (dB) de las mediciones reales respecto a los modelos simulados	43

ÍNDICE DE FIGURAS

Figura 3.1. Plano del laboratorio de Radiocomunicaciones	27
Figura 3.4. Rutas de mediciones para el experimento 1 y 2	32
Figura 3.5. Medición de pérdidas para el primer experimento	33
Figura 3.6. Medición de pérdidas de propagación en madera	34
Figura 3.7. Medición de pérdidas de propagación en vidrio.....	35
Figura 4.1. Gráfico de dispersión de Pérdidas de propagación vs distancia.....	38
Figura 4.2. Regresión lineal sobre las variables L(dB) vs distancia (m)	40
Figura 4.3. Comparación de modelos de estimación de pérdidas.....	41
Figura 4.4. Comparación de modelos de estimación de pérdidas en escala logarítmica	42

ÍNDICE DE ANEXOS

Anexo 1. Código para obtener mediciones del VNA (rutinaVNA.py)	48
Anexo 2. Código para procesar los archivos generados (rutinaVNA_procesamiento.py).....	52
Anexo 3. Hoja de datos de las antenas utilizadas.....	69
Anexo 4. Antenas usadas en las mediciones (vista frontal)	70
Anexo 5. Antenas usadas en las mediciones (vista lateral).....	70
Anexo 6. Vista Real de la Configuración del esquema de medición	71
Anexo 7. Laboratorio de Radiocomunicaciones	71
Anexo 8. Panel Frontal del VNA	72
Anexo 9. Cargas utilizadas para calibrar los 2 puertos	72
Anexo 10. MSE y RMSE (dB) mostrados en consola	73
Anexo 11. Base de datos generadas	73

CAPÍTULO 1

1. Introducción

En la actualidad una de las causas de problemas cuando se tiene enlaces en bandas sin licencias son las interferencias [1]. El crecimiento de redes inalámbricas de área local (WLAN), desde su estandarización en la década de los 90, gracias a los bajos costos y gran facilidad de implementación debido su capacidad de operar en bandas de frecuencias no licenciadas, las han convertido en la opción más factible para transmitir datos [2]. La tecnología que predomina actualmente en la banda de 2.4 GHz es Wireless Fidelity (WiFi) bajo el estándar IEEE 802.11 que opera en las bandas *Industrial, Scientific and Medical* (ISM); en esta banda predomina la interferencia debido a la gran concentración de redes en la misma locación y la poca disponibilidad de canales sin solapamiento [3]. Por ello, la Comisión Federal de Comunicaciones (FCC por sus siglas en inglés) liberó en 2014 la banda de los 5GHz para WiFi y otros usos no licenciados. Y posteriormente, en 2020 liberaron la banda de 6 GHz para el mismo propósito.

Garantizar el óptimo funcionamiento de estos sistemas requiere de previa planificación y dimensionamiento, por ende, uno de los requisitos es conocer el comportamiento del canal inalámbrico para estos entornos. Según [4], para modelar un canal, se requiere caracterizar el entorno, esto implica ver cómo influyen los obstáculos presentes en el medio, además de la frecuencia del canal y la distancia entre el transmisor y el receptor.

En el presente proyecto se propone un modelo empírico para la banda de 6 GHz en un entorno *indoor* a partir de las campañas de mediciones que se realizaron en un lapso de una semana en el laboratorio de Radiocomunicaciones del campus Gustavo Galindo en la ciudad de Guayaquil.

Primero se lleva a cabo una revisión del marco teórico acerca de los modelos de propagación ya existentes para ambientes interiores. De forma seguida, se describe la

configuración de los dispositivos para la toma de datos utilizado para las mediciones y se especifican los ambientes y rutas seleccionados. Posteriormente se presentan los resultados obtenidos, se desarrolla el modelo mediante técnicas de regresión lineal y otras herramientas estadísticas, y finalmente, se exponen las conclusiones obtenidas.

1.1. Definición del Problema

Debido al carácter libre de las bandas ISM, el crecimiento acelerado de WLANs demanda más espectro radioeléctrico para asegurar una comunicación efectiva [5]. Reconocer y predecir el comportamiento del canal inalámbrico resulta fundamental para el diseño eficiente de aquellos sistemas [6]. Con base en esto, los modelos de propagación para interiores empíricos usados actualmente, si bien son precisos para la estimación de pérdidas de propagación, son dependientes a las condiciones propias del entorno (externo a nuestro país) en los que fueron desarrollados [7]. En Ecuador, todavía no existen modelos empíricos *indoor* para WLAN desarrollados con una base de datos ampliable en entornos interiores específicos en las frecuencias de la banda ISM, por lo tanto, dificulta la planificación y despliegue de tecnologías que actualmente operan a nivel internacional en la banda en tendencia tal como la banda de 6 GHz que la FCC declaró en 2020 para WiFi y otros usos no licenciados [8].

1.2. Justificación del Problema

Los estudios de pérdidas de propagación permiten la toma de decisiones, en este caso, por parte de la Agencia de Regulación y Control de las Telecomunicaciones (ARCOTEL) para la gestión, aprovechamiento y optimización del espectro radioeléctrico [9].

Actualmente, existen herramientas para estimar cobertura y caracterizar el canal de propagación en ambientes interiores tales como WinProp de Altair y U-NET de Huawei, pero aquellos poseen licencias costosas y no se puede realizar ajustes necesarios al modelo desarrollado

debido a su naturaleza de código cerrado, lo que imposibilita ejecutar cálculos específicos. En ese contexto, un modelo desarrollado en su totalidad en Python permite que cualquier diseñador de sistemas de comunicaciones, estudiantes o interesados en la materia que requieran complementar su conocimiento acerca de propagación o diseñar proyectos relacionados al tema, puedan replicar los resultados o extraer bloques de código al ser este un lenguaje multiplataformas, de código abierto y con las librerías de acceso gratuito.

También cabe mencionar, que los modelos empíricos de pérdida de camino son de gran importancia ya que aumentan la fiabilidad al momento de planificar y desplegar una nueva WLAN, o en su defecto, evaluar las redes ya existentes en entornos similares [10].

1.3. Objetivo General

Implementar un modelo empírico en el segmento bajo de la banda de 6GHz (5.8 GHz – 6.4 GHz) mediante una herramienta creada íntegramente en Python para la estimación de pérdidas de la señal al propagarse en un entorno *indoor*.

Entre los objetivos específicos se tiene:

- Diseñar el esquema de mediciones y diagrama de conexiones de los equipos.
- Realizar campañas de mediciones automatizadas en medios *indoor*.
- Implementar el algoritmo en Python para la captación de las mediciones y procesamiento de estas mediante técnicas estadísticas.
- Comparar los resultados obtenidos con el modelo desarrollado y los modelos existentes mediante el Error Cuadrático Medio (MSE) para la validación de la expresión calculada.

1.4. Propuesta de Solución

Se propone un modelo empírico para WLAN desarrollado en su totalidad en Python que permita predecir las pérdidas de propagación en el canal inalámbrico en la banda de los 6 GHz aplicando técnicas de regresión lineal en un entorno *indoor* típico de una universidad.

1.5. Metodología

Se comenzó el proyecto con la etapa de recopilación de datos y antecedentes sobre la problemática en cuestión. Seguido de la selección de los instrumentos disponibles necesarios para las mediciones. Se comprendió conceptos como el protocolo SCPI [11] para la automatización de las mediciones, y procesos como la calibración de dos puertos del Vectorial Network Analyzer (VNA) para descartar las atenuaciones debida a los cables. Posteriormente, se recopiló los datos ejecutando un *script* hecho en Python por cada metro a lo largo del laboratorio; aquellos datos se leyeron, limpiaron y procesaron con otro *script* en Python para elaborar las gráficas necesarias. Finalmente, se presentaron los resultados obtenidos y las respectivas conclusiones.

CAPÍTULO 2

2. Marco Teórico

En el siguiente apartado se plasman los fundamentos teóricos del presente trabajo, tanto como conceptos, principios y elementos utilizados necesarios para comprender y predecir el comportamiento de la propagación de una onda electromagnética.

2.1. Bandas ISM

Son bandas de frecuencias reservadas internacionalmente para propósitos no relacionados a las telecomunicaciones (industriales, científicos y médicos). Inicialmente, esta parte del espectro radioeléctrico estaba reservada para dispositivos que emitían ondas de radio, como los microondas o los calentadores industriales. En 1985, la FCC designó las redes locales inalámbricas como aplicación para estas bandas. Y posteriormente, añadió la banda de 5 GHz para el mismo propósito [12].

2.2. Modelos de propagación para interiores

Se presenta de forma general cinco modelos utilizados para cálculos de pérdida de propagación en ambientes interiores consultados en este proyecto. Existen diversos modelos y se clasifican principalmente en modelos deterministas y semi-empíricos. Los primeros son más precisos, pero están limitados principalmente a la capacidad de cómputo y a las condiciones específicas en las cuales fueron desarrollados. Por otro lado, los modelos semi-empíricos proponen una facilidad en la simpleza de su naturaleza, debido a la adaptabilidad presentada frente a ambientes con poca variedad de obstáculos frente al uso para obtener el modelo [13].

2.2.1. Modelo Espacio Libre

Este modelo parte de la ecuación de Friis y se utiliza generalmente como una aproximación para estimar pérdidas de potencia en enlaces exteriores con línea de vista entre el transmisor y el receptor. Es comúnmente usado para enlaces satelitales. El modelo se obtiene a partir de la ecuación [14]:

$$P_r(d) = \frac{P_t G_t G_r}{L} \left(\frac{\lambda}{4\pi d} \right)^2 \quad (1)$$

donde P_t es la potencia transmitida en Watts y G_t y G_r son las ganancias (adimensionales) de las antenas transmisora y receptora respectivamente. $L \geq 1$ son pérdidas no relacionadas a la propagación. λ es la longitud de onda de la señal y d es la distancia en metros. Las pérdidas de propagación son obtenidas como una magnitud positiva en dB y está definida como la relación entre la Potencia transmitida y la Potencia recibida. Reemplazando los datos de la tabla 1 se obtiene que

$$L(dB) = 48.162 + 20 \log_{10}(d) \quad (2)$$

2.2.2. Modelo Log-Distance

Es un modelo clásico utilizado tanto para interiores como exteriores, también conocido como el modelo de espacio libre extendido. Se deriva como consecuencia de que la potencia de la señal recibida en un receptor disminuye de forma logarítmica y depende de un exponente n , llamado exponente de pérdida de paso, que cambia según el ambiente. La ecuación está dada por [15]:

$$L(db) = L(d_0) + 10n \log \left(\frac{d}{d_0} \right) + X_\sigma \quad (3)$$

Donde n es típicamente 2 para espacio libre, y conforme aumenten las condiciones del entorno aumenta n . El término d_0 es una distancia de referencia medida en km, que para canales inalámbricos en ambientes interiores es generalmente 0.001 km. Y X_σ es una variable gaussiana con unidades en decibelios que denota la variación de potencia promedio recibida [16]. Este modelo depende del exponente encontrado y de la precisión de la desviación estándar calculada en X_σ con base en las mediciones hechas. Mientras menor se requiere que sea la desviación estándar, más condiciones ambientales se deben tener en cuenta [17].

2.2.3. Modelo de ITU-R para interiores

Es un modelo único para ambientes interiores propuesto por la Unión Internacional de Telecomunicaciones (ITU por sus siglas en inglés) que comprende las frecuencias entre 900 MHz y 410 GHz. El modelo presentado en la ecuación (4) toma en cuenta tres coeficientes asociados al decremento de potencia debido a la distancia (α), al decremento de potencia debido a desplazamientos (β) y al decremento debido a la frecuencia (γ). El modelo se determina a partir de [18]:

$$L(d, f) = 10\alpha \log_{10}(d) + \beta + 10\gamma \log_{10}(f) \quad (4)$$

Donde f es la frecuencia dada en GHz, la distancia es dada en metros entre el transmisor y el receptor y los coeficientes α , β y γ están dados por el cuadro 2 [18], y equivalen a 1.46, 34.62 y 2.03 respectivamente. Lo que resulta en

$$L(dB) = 50.985 + 14.6 \log_{10}(d) \quad (5)$$

2.2.4. Modelo Multiparedes

También llamado modelo Motley-Keenan, es un modelo de predicción empírico diseñado para interiores que toma en cuenta las estructuras presentes del entorno como paredes, techos y pisos. El modelo original publicado en 1988 contemplaba desde 900 MHz hasta 1700MHz y consideraba únicamente un tipo de pared. La atenuación por pérdida puede ser descrita como [19]:

$$L = L_F + L_a \quad (6)$$

Donde L_F es la pérdida de espacio libre y L_a es una pérdida adicional debida a la influencia de la estructura del entorno y se desglosa de la siguiente manera:

$$L_a = L_c + \sum_{i=1}^I k_{f,i} L_{f,i} + \sum_{j=1}^J k_{w,j} L_{w,j} \quad (7)$$

Y L_c es una constante empírica, $k_{f,i}$ es el término asociado al tipo de paredes existentes penetradas, $k_{w,j}$ al número de pisos traspasados por la señal y el tipo de techo que posee el entorno de propagación. Cabe mencionar que en la creación de este modelo solo se consideró un tipo de

pared y piso, y la constante L_c fue obtenida a partir de un algoritmo de regresión lineal con base en las mediciones sobre las paredes y pisos de aquel entorno [19]. Se reemplaza los datos con una frecuencia máxima de 1700 MHz y atenuaciones de pisos y suelos iguales a cero debido a que el estudio considera línea de vista y se obtiene

$$L(dB) = 37 + 20\log_{10}(d) \quad (8)$$

2.2.5. Modelo COST 231

Este modelo de propagación diseñado para interiores se diferencia del anterior al incluir una componente no lineal para describir la atenuación de la señal al traspasar varios pisos. No obstante, mantiene una componente lineal para predecir la pérdida de energía debido a las paredes. La ecuación que describe el modelo está dada por [13]:

$$L_T = L_F + L_C + \sum_{i=1}^w L_{wi} n_{wi} + L_f n_f^{\left(\frac{(n_f+2)}{(n_f+1)} - b\right)} \quad (9)$$

Donde L_F es la pérdida de espacio libre entre el transmisor y el receptor, n_{wi} es el número de paredes atravesadas por la señal, w es el número de paredes de distinto tipo, L_{wi} es la atenuación debida a las paredes de tipo i . n_f es la cantidad de pisos y L_f es la atenuación debida al piso. Los valores típicos para un escenario interior de oficina son $L_c=37$, $b=0.46$, $n=4$ y $L_f=18.3$ [7].

2.3. Estándar IEEE 488.2

Fue lanzado en 1975 como un protocolo de comunicación que definió los aspectos mecánicos de una interfaz de bus de propósito general (GPIB por sus siglas en inglés) conocido también por su nombre IEEE 488.1. Más tarde, en 1991, se lanzó los comandos estándares para un instrumento programable (SCPI por sus siglas en inglés) como una capa añadida al protocolo IEEE 488.2. Los comandos SCPI son cadenas de tipo ASCII enviadas al instrumento que se desea controlar de forma remota a través de la capa física y forman parte del estándar VISA (Virtual Instrument Software Architecture) que surgió por la dificultad de encontrar librerías independientes para cada sistema de interfaz de un determinado dispositivo, y sirve para configurar sistemas de instrumentación que comprenden interfaces como GPIB, VXI, PXI, Serial, Ethernet y USB [20]. Fueron diseñados con el objetivo de reducir el tiempo de programación de pruebas automáticas mediante una serie de comandos versátiles y universales, independientes de los fabricantes de dispositivos [11]. Actualmente, existe una librería en el conocido lenguaje de programación Python llamada PyVisa que ofrece una capa extra para facilitar la programación de rutinas automatizadas usando comandos SCPI.

CAPÍTULO 3

3. Desarrollo de la solución

En el presente capítulo se describe cómo se desarrolló el proceso de medición y captación de los datos, esto incluye la lista detallada de instrumentos, el tipo de datos obtenidos y la forma de guardarlos.

3.1. Equipo Utilizado

Como transmisor y receptor se utilizaron dos antenas tipo bocina modelo ANT5158D18P-DP con ganancia de 18 dBi y polarización horizontal de la marca LANBOWAN debido principalmente a su disponibilidad y a que su coeficiente de reflexión se mantiene mucho menor a -10 dB en el rango de frecuencias requerido a pesar de que sus especificaciones indique que el rango óptimo termina en 5.8 GHz. Además, este tipo de antena es altamente directiva, con un ancho de haz horizontal de 32 grados, y uno vertical de 22 grados. De esa manera, aseguramos alta directividad con línea de vista como las pruebas requieren para un modelo de estimación de pérdidas básicas fiable. Las antenas se encuentran a una altura de 1.62 m respecto al suelo, conectadas a un analizador vectorial de redes (VNA) marca Agilent E5071B configurado como indica en la Tabla 3.1 para medir el parámetro S_{21} mediante dos cables con conector de tipo N de 7.75 m cada uno al puerto 1 y al puerto 2 respectivamente; y una laptop conectada al VNA mediante un cable USB-GPIB con 2 programas y dos librerías instaladas: La librería 'IO LibSuite_18_2_28014', Anaconda y Visual Studio Code con Python 3.9, y la librería PyVisa.

Tabla 3.1. Parámetros del esquema de medición

Parámetro	Valor
Potencia de salida VNA	0 dBm
Ancho de banda VNA	600 MHz
Frecuencia central VNA	6.1 GHz
Ancho de banda IF VNA	10 kHz
Ganancia de cada antena	18 dBi
Altura de antenas	1.62 m
Longitud de cada cable	7.75 m
Scattering	S21

Nota. No se añaden las atenuaciones por cable porque se realiza calibración de 2 puertos.

3.2. Escenario y Ruta de Medición

El ambiente es un laboratorio del campus Gustavo Galindo en la ciudad de Guayaquil, considerado de tipo suburbano, ubicado dentro del ala nueva de laboratorios de la Facultad de Ingeniería en Electricidad y Computación. En la Figura 3.1 se observa el plano del escenario escogido, así como la ruta de las diez mediciones realizadas dentro del laboratorio. El laboratorio posee unas dimensiones de 10.87 metros de largo y 7.06 metros de ancho. Cuenta con un cubículo hecho de vidrio de cuyas dimensiones son 4.79 x 2 metros, y dentro de este descansan dos escritorios con sus respectivas computadoras. Existe un mesón que recorre todo el laboratorio donde reposan cinco computadoras y ocho sillas altas. Cerca de la ruta de medición se encuentran dos mesas altas en las que están dispuestos diversos dispositivos activos de radiofrecuencia.

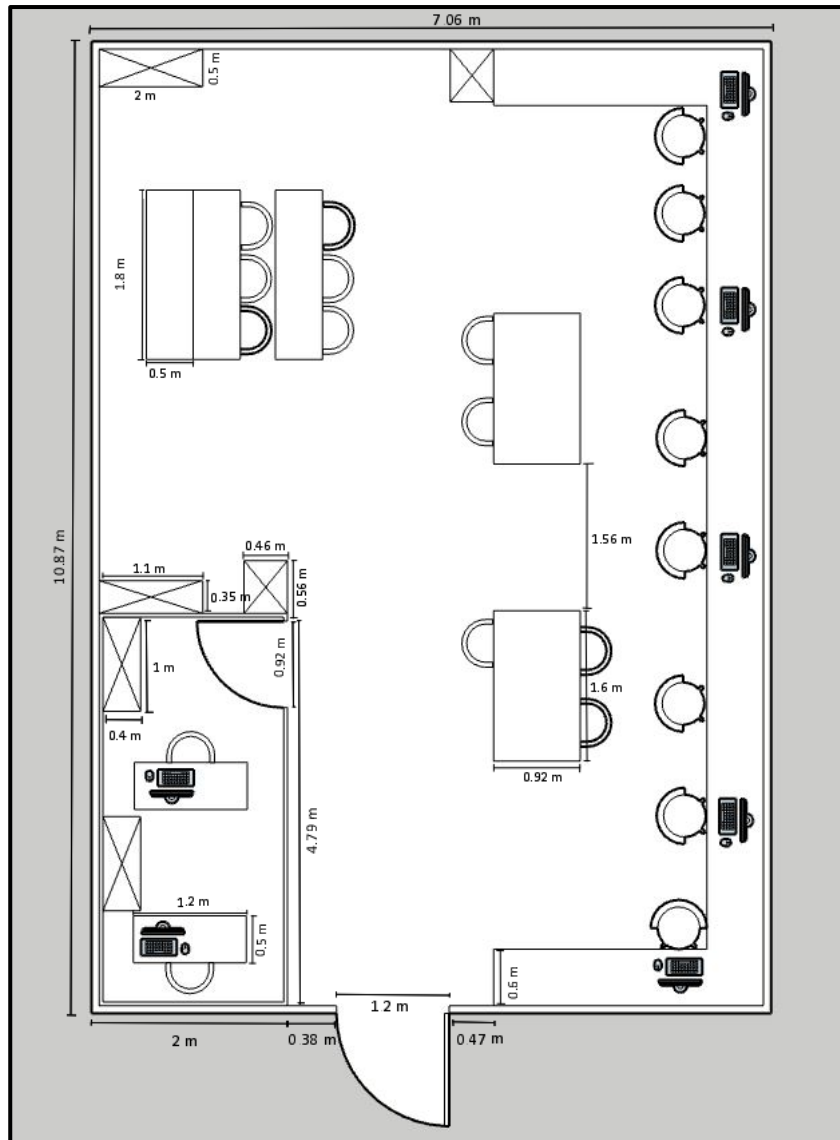


Figura 3.1. Plano del laboratorio de Radiocomunicaciones

3.3. Procedimiento de Medición

El proceso inició configurando el parámetro S21 y las frecuencias de operación (5.8 GHz y 6.4 GHz) en el VNA. Posteriormente se realizó una calibración de 2 puertos para descartar las pérdidas de energía debida a los cables. Una vez realizado este proceso, se conectan los cables a las antenas y se colocan los trípodes en el sitio mostrado en la Figura 3.2.

En la Figura 3.2 se muestra la conexión de los equipos utilizados en la medición con línea de vista en la banda de frecuencia comprendida entre 5.8 GHz a 6.4 GHz. Por el puerto 1 se transmite una señal continua interna desde el VNA con una potencia de 0 dBm por la Antena TX; que se recibe en la Antena RX y es capturado y guardado en un archivo '.txt' mediante el código escrito en Python.

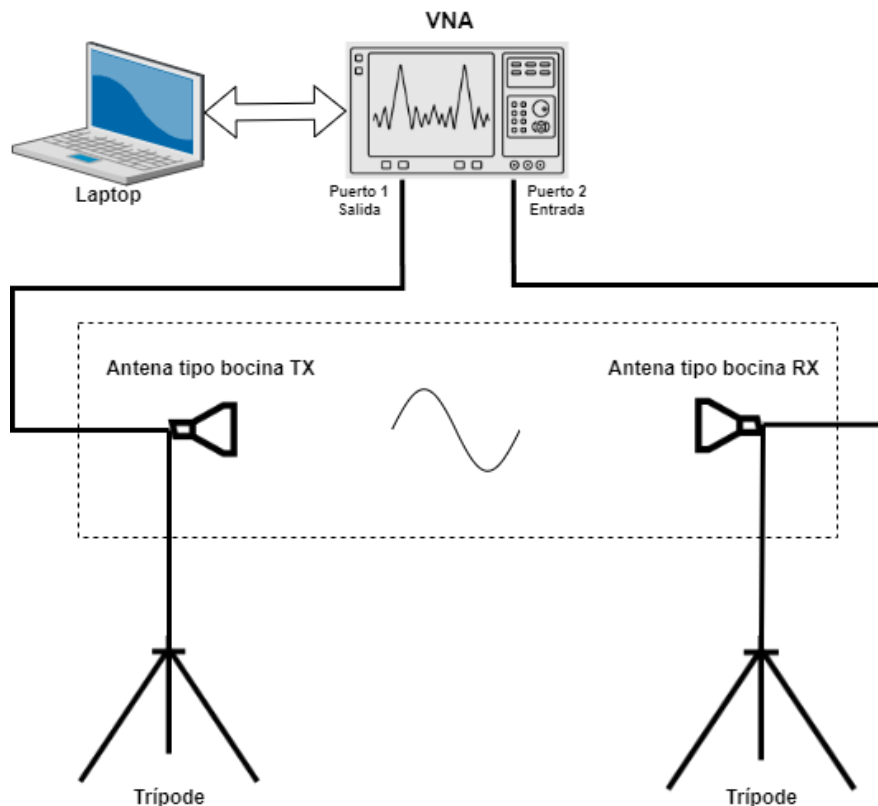


Figura 3.2. Diagrama de medición

3.4. Captura y Procesamiento de Mediciones

El diagrama de flujo mostrado en la Figura 3.3 describe de forma secuencial los pasos generales que se siguió para la obtención de los coeficientes de la regresión lineal y los gráficos correspondientes. Una vez calibrado los dos puertos del VNA, se procedió a colocar las antenas en el primer metro, y se ejecutó el script “rutinaVNA.py” con un nombre específico, consecuentemente se generó el archivo .txt correspondiente a ese metro. En caso de presentarse un error en el archivo .txt obtenido, se volvió a realizar la medición. Seguido, se procedió a mover el receptor RX al segundo metro y se repitió el proceso. De la misma forma hasta completar los nueve metros a lo largo de la línea descrita en la Figura 3.4 cambiando cada vez el nombre del archivo para que no se sobrescriba.

Posteriormente, con los archivos de preferencia en el mismo directorio, se establecieron los valores para las variables número de puntos y cantidad de muestras, a continuación, se reasignaron los valores de las variables referentes a las matrices con los nombres correctos de los archivos generados, y se ejecutó el script “rutinaVNA_procesamiento.py”. Este procedió a leer los archivos, eliminarles los espacios en blanco, los saltos de línea, almacenar aquellos valores en matrices y realizar los cálculos pertinentes.

Las secciones 3.4.1 y 3.4.2 describen a detalle el funcionamiento de estos dos scripts realizados explicando las funciones involucradas y procesos seguidos.

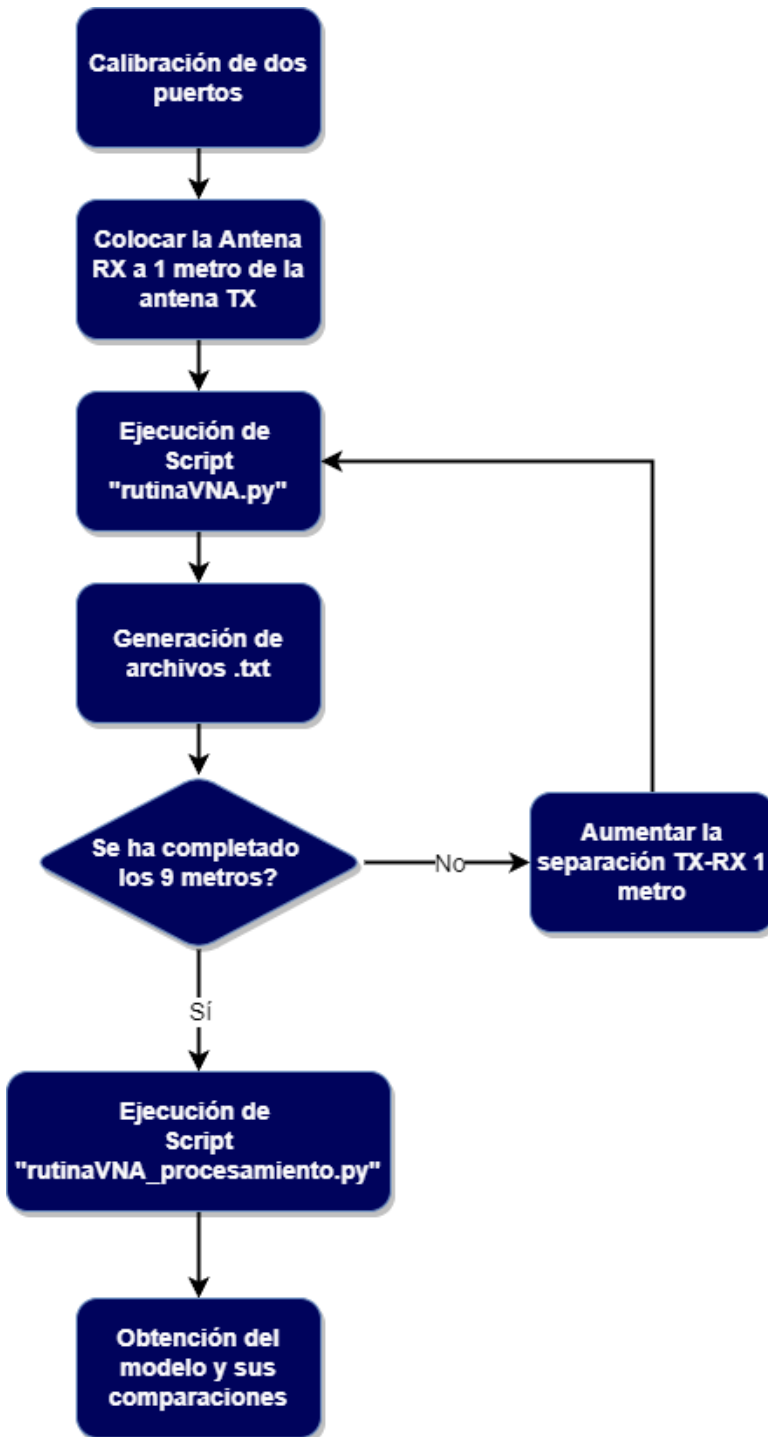


Figura 3.3. Diagrama de bloques del proceso general de medición

3.5. Pruebas realizadas

En este apartado, se describen los dos experimentos llevados a cabo para la verificación de los algoritmos y procedimientos definidos en 3.1, 3.2, 3.3 y 3.4. Ambos fueron ejecutados en un ambiente *indoor* con condiciones de escaso personal (2 personas) a lo largo de un mes y una semana en total, donde se verificó la calidad de los datos obtenidos y se desearon los resultados que presentaban errores en el formato recibido hasta obtener las bases de datos. La Figura 3.4 muestra los dos experimentos realizados. La línea negra representa la ruta del experimento uno, en el cual se recorrió la antena receptora RX a lo largo de nueve metros. La línea amarilla representa la ruta en la cual se realizó la prueba de medición de atenuación en madera, la línea azul representa la ruta de la prueba sobre la atenuación con vidrio y la línea amarilla es la ruta de la prueba realizada donde se consideró como obstáculo una pared de hormigón.

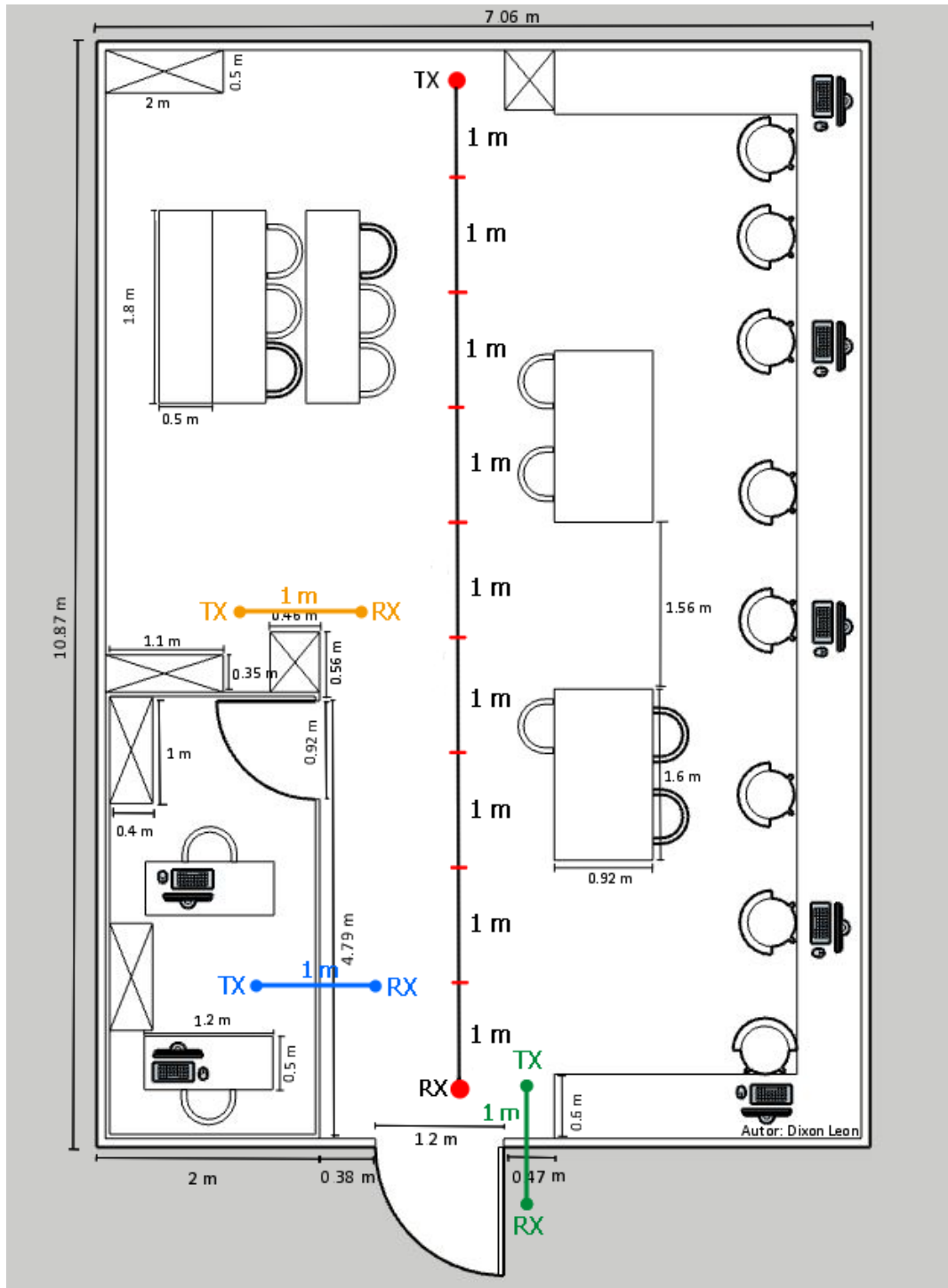


Figura 3.4. Rutas de mediciones para el experimento 1 y 2

3.5.1. Experimento 1

Fue llevado a cabo durante un mes y consistía en aumentar la separación TX-RX un metro en condiciones de línea de vista partiendo de la hipótesis que a esas frecuencias (Tabla 1), las pérdidas de camino no poseían un comportamiento similar a la ecuación de Friis (ecuación 2). Y posteriormente registrar en cada metro el parámetro S21 censado por el VNA.

En la Figura 3.5 se aprecia una de las nueve pruebas realizadas en el escenario, donde las antenas estaban ubicadas en condiciones de línea de vista con una separación de 2 metros entre las antenas. Bajo esas condiciones, se recopilaron en total 270270 muestras correspondientes al parámetro S21 que posteriormente se procesaron para la estimación del modelo.



Figura 3.5. Medición de pérdidas para el primer experimento

3.5.2. Experimento 2

El segundo experimento constó de las mediciones de atenuaciones de potencia en condiciones NLOS (sin línea de vista) a un metro de distancia entre TX y RX de dos distintos materiales comunes presentes en el laboratorio.

Para la primera prueba mostrada en la Figura 3.6, se colocó en medio madera de aproximadamente 2 cm de grosor y se procedió a extraer los datos del VNA, los cuales fueron un total de 15000 muestras para su posterior procesamiento.



Figura 3.6. Medición de pérdidas de propagación en madera

De la misma manera, se realizó la toma de mediciones colocando las antenas con una separación de un metro, y se dispuso como obstáculo una sección de vidrio en el medio del camino entre el transmisor y el receptor con el objetivo de censar las pérdidas causadas por dicho material como se observa en la Figura 3.7. Se recopilaron 15000 muestras en esta etapa del experimento. Adicional a las dos pruebas con madera y vidrio, se realizó la medición colocando como objeto obstructor una pared de concreto.



Figura 3.7. Medición de pérdidas de propagación en vidrio

CAPÍTULO 4

4. Resultados

En el presente capítulo se analizan los resultados obtenidos a partir de las bases de datos obtenidas en formato .txt y se describe el proceso que se siguió para la obtención del modelo empírico, su correspondiente comparación y validación con otros modelos existentes mediante las simulaciones respectivas y el error cuadrático medio. Adicionalmente, se incluyen las atenuaciones medias encontradas en los experimentos descritos en la sección 3.5.2

4.1. Pérdidas de Propagación Medidas

Cada muestra contenida en los archivos .txt generados a partir de las mediciones realizadas corresponden a la pérdida de inserción dadas por la ecuación

$$L_{insertion}(db) = -10 \log_{10} \left(\frac{P_t}{P_r} \right) \quad (10)$$

donde L y P_r se pueden tratar como arreglos de dos dimensiones y P_t equivale a un arreglo de dos dimensiones donde todos sus valores son 0 dbm (véase Tabla 1). Mediante el reemplazo de valores y el manejo de la aritmética la ecuación 10 resultante es

$$P_r(db) = L_{insertion} - 30db \quad (11)$$

Ahora que se dispone de la potencia recibida, se procede a reemplazar los datos en la ecuación de presupuesto de potencia, dada por [21]:

$$L_p = P_T - P_R + G_T + G_R - L_T - L_R \quad (12)$$

Donde:

P_T : es la potencia transmitida en dBm

P_R : es la potencia recibida, que en este caso es $L_{insertion} - 30db$

G_T y G_R son las ganancias de las antenas transmisora y receptora

L_T y L_R son las atenuaciones causadas por los cables en el transmisor y receptor respectivamente

Resolviendo con los datos de la Tabla 2 se obtiene que

$$L(dB) = -L_{insertion} + 36 \quad (13)$$

donde las pérdidas de inserción se refiere las magnitudes contenidas en los archivos extraídos y son arreglos en dos dimensiones en *Numpy*.

A continuación, se muestra en la Figura 4.1 las pérdidas de propagación en función de la distancia para la ruta de medición escogida en un rango de medición de 5.8 GHz a 6.4 GHz y trayectoria directa a modo de gráfico de dispersión. También se puede observar que con la configuración actual de la Tabla 3.1, los valores medidos prácticamente no distan del valor promedio en todo el rango de frecuencia analizados por cada metro.

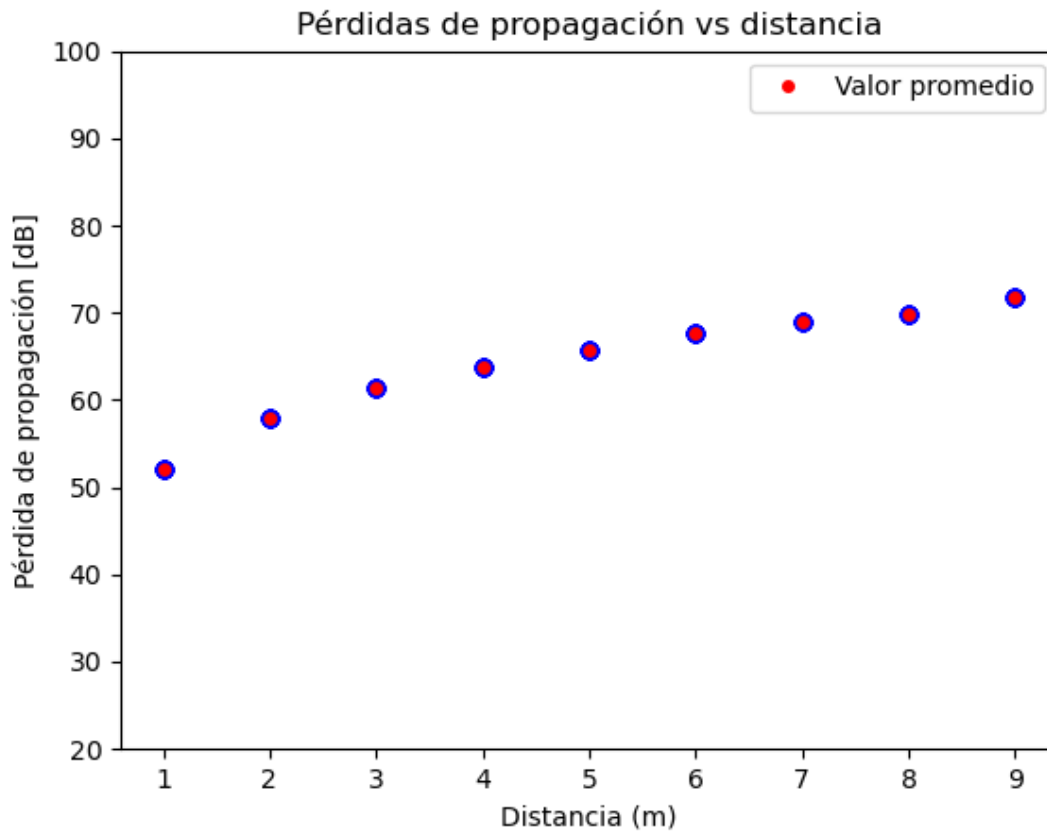


Figura 4.1. Gráfico de dispersión de Pérdidas de propagación vs distancia

Nota. Los valores medidos se muestran en color azul y se observan levemente debajo de los valores en rojo (valor promedio).

4.2. Desarrollo de Modelo de Propagación

El modelo es construido usando como herramienta la regresión lineal simple, un método estadístico que estudia la relación entre dos variables; en este caso, entre las pérdidas de propagación y la distancia, y permite encontrar los coeficientes de una recta capaz de predecir el comportamiento de la variable respuesta en función de la variable predictora o independiente. El modelo de regresión lineal simple está dado por [22]:

$$Y = \beta_0 + \beta_1 X_1 + \epsilon \quad (14)$$

Donde β_1 es llamado pendiente, β_0 el intercepto y ϵ es el error aleatorio con distribución gaussiana de media cero y varianza σ^2 .

En los resultados de la Figura 4.2 se observa la curva que describe el comportamiento de las mediciones tomadas. La pendiente hallada es 2.2197, el intercepto es 53.2251 y el error cuadrático medio es 2.648. Además, el coeficiente de determinación R² (que va de 0 a 1) es 0.92, lo que indica que la proporción de variación de los resultados que puede explicarse con el modelo es bastante baja, es decir, el modelo encontrado se ajusta en un 92% a las mediciones reales.

En consecuencia, y considerando que ambos lados de la ecuación deben estar en escala logarítmica, el modelo obtenido con los parámetros encontrados está dado por la siguiente expresión

$$L(dB) = 10(2.2197)\log_{10}(d) + (53.2251) - 2.648 \quad (15)$$

donde d es la separación entre el transmisor y el receptor expresada en metros.

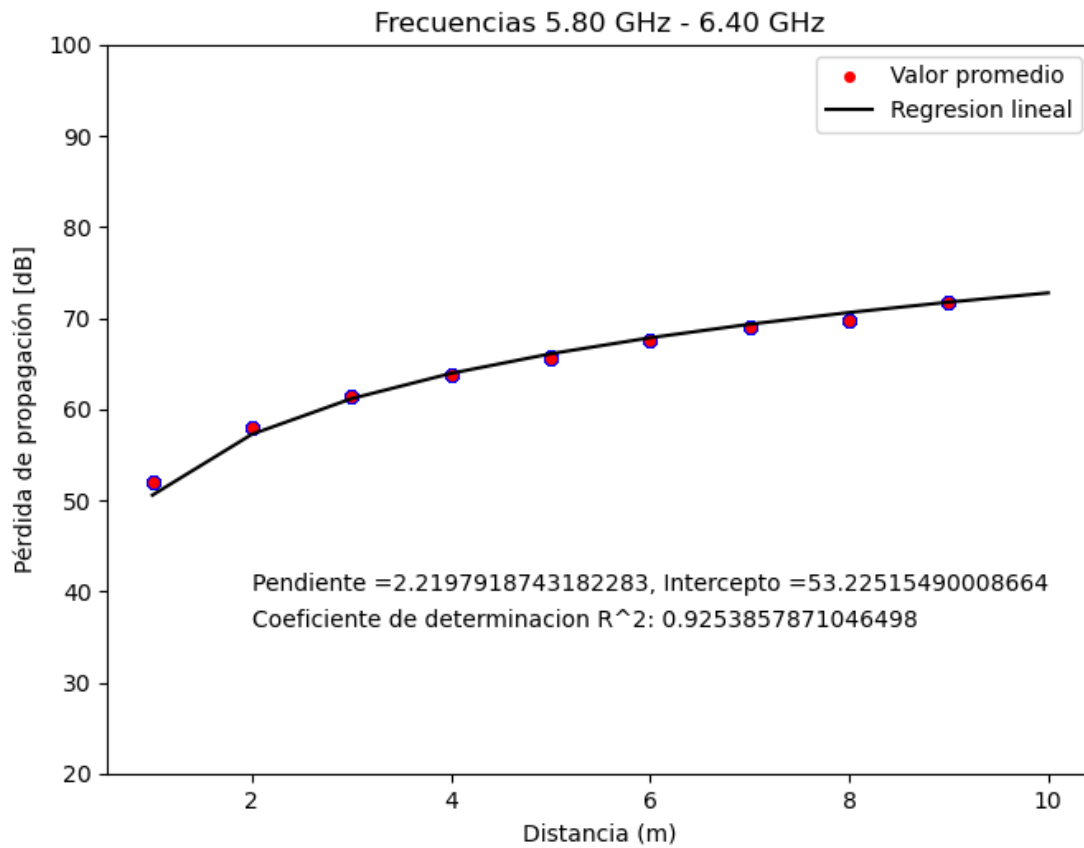


Figura 4.2. Regresión lineal sobre las variables $L(\text{dB})$ vs distancia (m)

Nota. Los valores medidos abarcan todo el rango de frecuencia analizados, y se muestran los valores de la pendiente, intercepto y coeficiente de determinación obtenidos en la parte inferior de la gráfica.

4.3. Comparación de Resultados

Con el fin de comparar el nuevo modelo de estimación de pérdidas, se procede a comparar los resultados de las pérdidas de propagación obtenidas de mediciones reales tanto con el modelo nuevo como con otros modelos ya existentes en la literatura. En la Figura 4.3 se muestra que el modelo nuevo se ajusta de manera eficaz a los datos medidos, mientras que el modelo de Keenan-Motley es el que más se aleja, subestimando las pérdidas de camino. Cabe mencionar que el modelo de espacio libre es el más cercano en cuanto a la forma de la curva subestimando un poco las pérdidas de propagación.

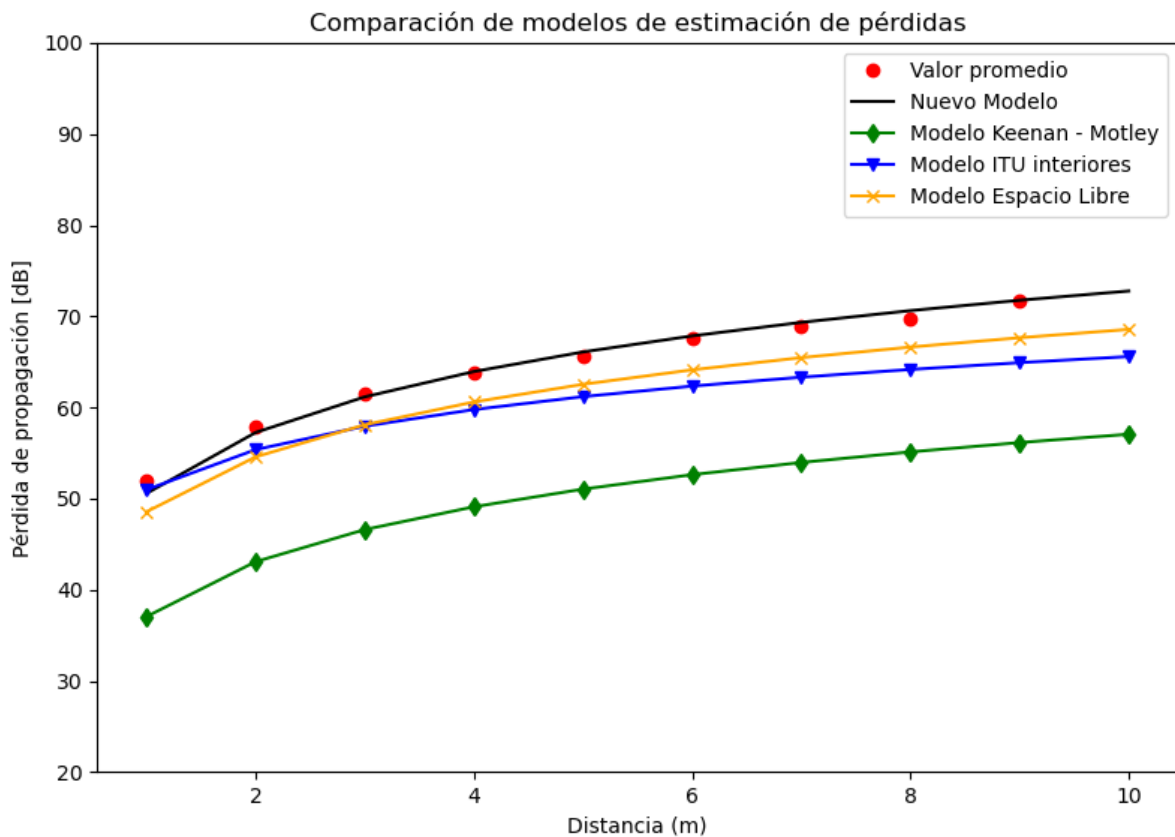


Figura 4.3. Comparación de modelos de estimación de pérdidas

Nota. Se utilizaron las ecuaciones (2), (5), (8) y (15) para las comparaciones entre modelos.

En la Figura 4.4 se muestran graficadas las mismas ecuaciones, pero con la distancia en escala logarítmica. Las pendientes tanto del nuevo modelo como de los modelos Keenan y Espacio Libre mantienen una similitud. Por otra parte, la pendiente del modelo ITU para interiores está más acotada, lo que indica que a medida que la distancia aumente, el modelo comienza a subestimar las pérdidas de camino en condiciones de línea de vista en un entorno estacionario a una frecuencia de transmisión de 6.11 GHz.

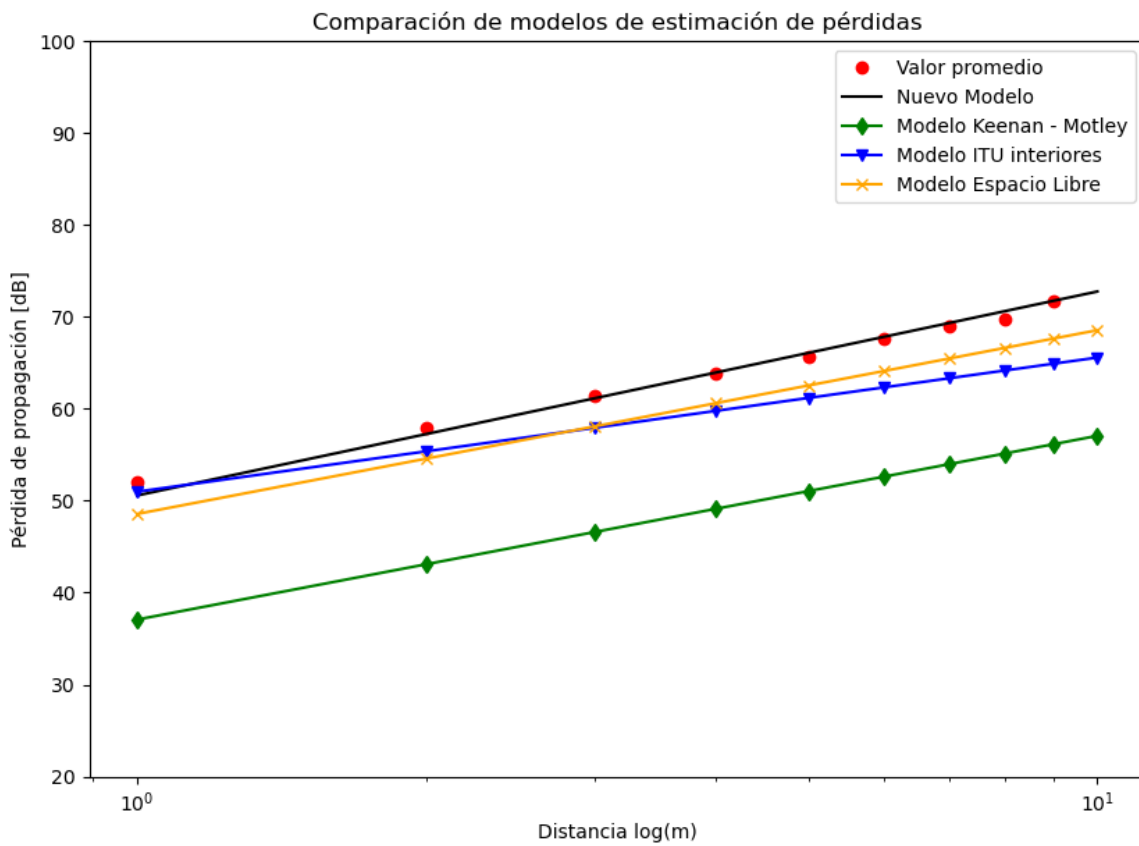


Figura 4.4. Comparación de modelos de estimación de pérdidas en escala logarítmica

Por otra parte, la raíz del error cuadrático medio calculada (RMSE por sus siglas en inglés) a partir de las comparaciones entre los errores obtenidos de los distintos modelos respecto a los valores reales sirve para cuantificar el desempeño de los demás modelos y permiten validar el rendimiento del modelo nuevo [23]. La Tabla 4.1 indica los distintos valores de raíces del error cuadrático medio donde se observa que el mayor valor obtenido corresponde al modelo recomendado por la ITU para interiores con línea de vista con un valor de 21.46, lo cual indica que el modelo subestima las pérdidas reales. El nuevo modelo, extraído a partir de las condiciones reales del entorno, presenta un RMSE de 2.64.

Tabla 4.1. RMSE (dB) de las mediciones reales respecto a los modelos simulados

Modelo	RMSE (dB)
Keenan-Motley (Multiparedes)	14.90
ITU-R interiores	21.46
Espacio Libre	11.66
Nuevo	2.64

CAPÍTULO 5

5. Conclusiones y Recomendaciones

En este capítulo se detallan las conclusiones derivadas del análisis de resultados mostrado en el capítulo anterior. Además, se presentan recomendaciones con base en la experiencia adquirida a lo largo del desarrollo del proyecto, y sugerencias respecto a la posible mejoría tanto del modelo implementado como el programa desarrollado.

5.1. Conclusiones

En el presente proyecto se desarrolló un modelo empírico para predecir las pérdidas de propagación entre los 5.8 GHz y 6.4 GHz en un laboratorio de universidad ubicado en la ciudad de Guayaquil mediante una herramienta creada únicamente en Python. Se contempló como variable predictora a la separación entre el transmisor y receptor, y como variable dependiente a la pérdida de inserción obtenida a través de la medición del parámetro de *scattering* S21, que relaciona la potencia de la señal incidente con la potencia de la señal transmitida [24].

Se implementó el algoritmo escrito únicamente en Python y dividido en dos *scripts* que fue capaz de comunicarse con el VNA, y en el cual se captó y guardó un total de 270270 muestras correspondientes al nivel de potencia recibido repartidos en 9 archivos en formato '.txt', que fueron procesados correctamente.

El rendimiento del modelo nuevo obtenido para la estimación de pérdidas de propagación con base en la raíz del error cuadrático medio, en contraste con los modelos ITU R para interiores, Keenan-Motley reducido y el modelo de Espacio Libre, fue significativamente superior en los casos revisados.

5.2. Recomendaciones

La importancia de obtener modelos para la predicción de cobertura en ambientes interiores de forma automática y versátil radica en el uso de un lenguaje de código abierto, multiplataforma, con soporte constante de parte de la comunidad para mantener librerías actualizadas y totalmente modificable por el usuario, lo que convierte a esta herramienta en una gran opción para el aprendizaje efectivo, por ello se sugiere añadir su uso en programas de estudio de materias relacionadas al tópico tal como Propagación.

La fiabilidad de los modelos empíricos depende en gran parte de la base de datos del ambiente analizado [13]. Por ello, es recomendable realizar estudios en otros tipos de ambientes interiores, como viviendas, cafeterías, oficinas, entre otros añadiendo condiciones específicas como línea de vista obstruida (OLOS) o sin línea de vista (NLOS).

El modelo obtenido depende únicamente de la separación entre transmisor y receptor. Existen otros parámetros que permiten complementar la caracterización de un canal de propagación tal como la fase de la señal, la altura del transmisor y/o receptor y las atenuaciones causadas por pisos y techos. Por lo tanto, se considera necesario extender las campañas de medición con el fin de robustecer el modelo y tomar en cuenta los parámetros antes mencionados.

BIBLIOGRAFÍA

- [1] R. C. Guevara C. y É. Serna M., «Una propuesta de solución al problema de la interferencia entre redes WiFi por solapamiento de canales.», *Cienc. E Ing. Neogranadina*, vol. 23, n.º 2, p. 7, dic. 2013, doi: 10.18359/rcin.220.
- [2] «History of IEEE 801-11 Standardization.pdf». Accedido: 2 de agosto de 2022. [En línea]. Disponible en: <https://ece3115.elimu.net/Books/WiMAX/History%20of%20IEEE%20801-11%20Standardization.pdf>
- [3] R. C. G. Calume, «Cuantificación Experimental de la Interferencia entre redes WiFi causado por el solapamiento de canales», *Cuad. Act.*, vol. 5, pp. 23-29, 2013, Accedido: 14 de junio de 2022. [En línea]. Disponible en: <https://ojs.tdea.edu.co/index.php/cuadernoactiva/article/view/125>
- [4] P. Pajusco, «Propagation channel models for mobile communication», *Comptes Rendus Phys.*, vol. 7, n.º 7, pp. 703-714, sep. 2006, doi: 10.1016/j.crhy.2006.07.002.
- [5] R. Z. L. Cristina y T. R. L. Washington, «TÍTULO: “ANÁLISIS DE SATURACIÓN DEL ESPECTRO ELECTROMAGNÉTICO EN LA BANDA NO LICENCIADA DE 5GHZ EN LA ZONA URBANA DE LA CIUDAD DE GUAYAQUIL Y SU IMPACTO EN EL DESPLIEGUE DE EQUIPOS CON TECNOLOGÍA WIFI 802.11AC”», p. 218.
- [6] ITU, «RECOMENDACIÓN UIT-R P.1238-11 - Datos de propagación y métodos de predicción para la planificación de sistemas de radiocomunicaciones en interiores y redes radioeléctricas de área local en la gama de frecuencias de 300 MHz a 450 GHz*», p. 33, sep. 2021.
- [7] J. M. Torres, Á. Pinto-Mangones, M. R. Macea A, N. A. Pérez-García, y L. M. Rujano, «Modelo para la estimación de las pérdidas de propagación en redes WLAN operando en 2,4 ghz y 5,8 ghz, para ambientes interiores de edificios comerciales», *Univ. Cienc. Tecnol.*, vol. 20, n.º 78, pp. 42-53, mar. 2016, Accedido: 26 de julio de 2022. [En línea]. Disponible en: http://ve.scielo.org/scielo.php?script=sci_abstract&pid=S1316-48212016000100004&lng=es&nrm=iso&tlng=es
- [8] «FCC Opens 6 GHz Band to Wi-Fi and Other Unlicensed Uses», *Federal Communications Commission*, 23 de abril de 2020. <https://www.fcc.gov/document/fcc-opens-6-ghz-band-wi-fi-and-other-unlicensed-uses> (accedido 2 de septiembre de 2022).
- [9] Arcotel, «Ley Orgánica de Telecomunicaciones», p. 40.
- [10] T. Chrysikos, G. Georgopoulos, y S. Kotsopoulos, «Wireless channel characterization for a home indoor propagation topology at 2.4 GHz», en *2011 Wireless Telecommunications Symposium (WTS)*, abr. 2011, pp. 1-10. doi: 10.1109/WTS.2011.5960879.
- [11] C. SCPI, «Standard Commands for Programmable Instruments (SCPI)». mayo de 1999. Accedido: 10 de agosto de 2022. [En línea]. Disponible en: <https://www.ivifoundation.org/docs/scpi-99.pdf>
- [12] «Definition of ISM band», *PCMAG*. <https://www.pcmag.com/encyclopedia/term/ism-band> (accedido 3 de agosto de 2022).
- [13] A. Alhamoud *et al.*, «Empirical investigation of the effect of the door's state on received signal strength in indoor environments at 2.4 GHz», en *39th Annual IEEE Conference on Local Computer Networks Workshops*, Edmonton, AB, Canada, sep. 2014, pp. 652-657. doi: 10.1109/LCNW.2014.6927716.

- [14] H. T. Friis, «A Note on a Simple Transmission Formula», *Proc. IRE*, vol. 34, n.º 5, pp. 254-256, may 1946, doi: 10.1109/JRPROC.1946.234568.
- [15] T. S. Rappaport, *Wireless communications: principles and practice*, Second edition, Twentieth impression 2019, Indian subcontinent adaption. Noida: Pearson India Education Services, 2010.
- [16] J. B. Andersen, T. S. Rappaport, y S. Yoshida, «Propagation measurements and models for wireless communications channels», *IEEE Commun. Mag.*, vol. 33, n.º 1, pp. 42-49, ene. 1995, doi: 10.1109/35.339880.
- [17] S. Y. Seidel y T. S. Rappaport, «914 MHz path loss prediction models for indoor wireless communications in multifloored buildings», *IEEE Trans. Antennas Propag.*, vol. 40, n.º 2, pp. 207-217, feb. 1992, doi: 10.1109/8.127405.
- [18] «RECOMENDACIÓN UIT-R P.1238-11 - Datos de propagación y métodos de predicción para la planificación de sistemas de radiocomunicaciones en interiores y redes radioeléctricas de área local en la gama de frecuencias de 300 MHz a 450 GHz*», p. 33.
- [19] J. Peng, *Communications and Networking*. BoD – Books on Demand, 2010.
- [20] M. Dartiailh, «PyVISA: Control your instrument with Python». [En línea]. Disponible en: <https://pyvisa.readthedocs.io/en/latest/>
- [21] A. Rahimian y F. Mehran, «RF link budget analysis in urban propagation microcell environment for mobile radio communication systems link planning», en *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, Nanjing, China, nov. 2011, pp. 1-5. doi: 10.1109/WCSP.2011.6096799.
- [22] A. Carrasquilla-Batista *et al.*, «Regresión lineal simple y múltiple: aplicación en la predicción de variables naturales relacionadas con el crecimiento microalgal», *Rev. Tecnol. En Marcha*, vol. 29, pp. 33-45, dic. 2016, doi: 10.18845/tm.v29i8.2983.
- [23] A. Torres, «Aprendizaje automático: Una introducción al error cuadrático medio y las líneas de regresión». 14 de octubre de 2021. [En línea]. Disponible en: <https://www.freecodecamp.org/espanol/news/aprendizaje-automatico-una-introduccion-al-error-cuadratico-medio-y-las-lineas-de-regresion/>
- [24] D. J. Choma, «Scattering Parameters: Concept, Theory, and Applications», p. 35, 2006.

ANEXOS

Anexo 1. Código para obtener mediciones del VNA (rutinaVNA.py)

```
"""
Created on Tue Jul 12 10:38:57 2022

@author: Dixon Leon
"""
import pyvisa as vi
import numpy as np
import time
import winsound

rm = vi.ResourceManager()
print(rm.list_resources())
ena_inst = rm.open_resource('GPIB0::17::INSTR')
#ena_inst = rm.open_resource('TCPIP0::192.168.0.1::inst0::INSTR')

#Imprime la identificacion del instrumento
print("Datos del dispositivo conectado: " + str(ena_inst.query("*IDN?")))

print("Comenzando las configuraciones...")

#Configurando las condiciones y parametros de medicion
#ena_inst.write(":SENS1:FREQ:STAR 5800E6") #frecuencia inicial (5800 MHz)
#ena_inst.write(":SENS1:FREQ:STOP 6400E6") #frecuencia final (6400 MHz)
ena_inst.write(":DISP:WIND1:ACT") #Seleccionar el canal activo
ena_inst.write(":CALC1:PAR1:SEL") #Habilitar una traza
ena_inst.write(":CALC1:PAR:COUN1") #Configurar el número de trazas
ena_inst.write(":CALC1:PAR1:DEF S21") #Definir el parametro S21
ena_inst.write(":SENS1:SWE:TYPE LIN") #Seleccionar el tipo de barrido
(lineal)
ena_inst.write(":SENS1:SWE:GEN STEP") #Seleccionar el modo de barrido
(escalonado)
ena_inst.write(":SENS1:SWE:POIN 1001") #NUMERO DE PUNTOS
ena_inst.write(":CALC1:FORM MLOG") #Formato de datos (Eje Y en db)
ena_inst.write(":SENS1:BAND 10000") #Modifica el IF Bandwidth a 10 kHz

'''Configuraciones adicionales'''

#ena_inst.write(":SENS1:SWE:TIME:AUTO") #Tiempo de medición (automático)
#ena_inst.write(":DISP:SPL D1") #Modo de despliegue de un canal
#ena_inst.write(":DISP:WIND1:SPL D1") #Modo de despliegue de una traza
#ena_inst.write(":DISP:WIND1:TRAC1:Y:AUTO") #Activar el autoscale
```

```

#ena_inst.write(':DISP:WIND1:TITL:DATA ""S21"') #Título del área del canal 1
"S21"
"""
ena_inst.write(":CALC1:PAR1:DEF?") #Solicita el valor del parametro escogido
param_s=ena_inst.read()

"""

ena_inst.write(":SENS1:FREQ:STAR?") #Solicita la frecuencia inicial escogida
frec_inicial=ena_inst.read()
ena_inst.write(":SENS1:FREQ:STOP?") #Solicita la frecuencia final escogida
frec_final=ena_inst.read()

datos_magnitud=[]
datos_fase=[]
datos_freq=[]
datos_complejos=[]

#El tiempo de medicion esta definido por la multiplicacion del
#numero de iteraciones x tiempo.sleep

for i in range(30):
    ena_inst.write(":CALC1:DATA:FDAT?")
    datos_magnitud.append(ena_inst.read())
    ena_inst.write(":SENS1:FREQ:DATA?")
    datos_freq.append(ena_inst.read())

    #time.sleep(7)

#Guardar la lista de listas generada en los archivos únicos generados de
magnitud y los valores en frecuencia
archivo_mediciones=open("Mediciones corregidas 9 metro magnitud db.txt",
"w+")
datos_magnitud=",".join(datos_magnitud)
archivo_mediciones.write(datos_magnitud)
archivo_frecuencias=open("Frecuencias corregidas 9 metro magnitud db.txt",
"w+")
datos_freq=",".join(datos_freq)
archivo_frecuencias.write(datos_freq)

#Se cambia a fase el VNA
ena_inst.write(":CALC1:FORM PHAS") #Formato de datos (Eje Y en FASE)

```

```

#Obtención de los barridos
for i in range(30):
    ena_inst.write(":CALC1:DATA:FDAT?")
    datos_fase.append(ena_inst.read())

    #time.sleep(7)

#Guardar la lista generada en un archivo con su propio nombre
archivo_mediciones=open("Mediciones corregidas 9 metro fase.txt", "w+")
datos_fase=",".join(datos_fase)
archivo_mediciones.write(datos_fase)

#Barrido para obtener valores complejos no formateados
for i in range(30):
    ena_inst.write(":CALC1:DATA:SDAT?")
    datos_complejos.append(ena_inst.read())

    #time.sleep(7)

archivo_mediciones=open("Mediciones corregidas 9 metro complejo.txt", "w+")
datos_complejos=",".join(datos_complejos)
archivo_mediciones.write(datos_complejos)

#Se cambia a GROUP DELAY el VNA
ena_inst.write(":CALC1:FORM GDEL") #Formato de datos (Eje Y en FASE)

for i in range(cantidad_muestras):
    ena_inst.write(":CALC1:DATA:FDAT?")
    datos_group_delay.append(ena_inst.read())
    ena_inst.write(":SENS1:FREQ:DATA?")
    datos_tiempo_group_delay.append(ena_inst.read())

    #time.sleep(7)

#Guardar la lista de listas generada en los archivos únicos generados de magnitud
y los valores en frecuencia
archivo_mediciones=open("Mediciones corregidas 9 metro group delay.txt", "w+")
datos_group_delay=",".join(datos_group_delay)
archivo_mediciones.write(datos_group_delay)
archivo_frecuencias=open("Tiempo corregidas 9 metro group delay.txt", "w+")
datos_tiempo_group_delay=",".join(datos_tiempo_group_delay)
archivo_frecuencias.write(datos_tiempo_group_delay)

```

```
print(
    "Configuraciones terminadas \n" +
    "Frecuencia inicial: " + str(frec_inicial) + "\n" +
    "Frecuencia final: " + str(frec_final) + "\n"
)

#Cerrar las instancias de los archivos
archivo_mediciones.close()
archivo_frecuencias.close()

#Para saber cuando termina el programa de ejecutarse añadimos un sonido
frecuencia=1500
duracion=1000
winsound.Beep(frecuencia, duracion)
```

Anexo 2. Código para procesar los archivos generados (rutinaVNA_procesamiento.py)

```
from cProfile import label
from pydoc import locate
from shutil import which
from turtle import color
import numpy as np
import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

'''

Programa para leer archivos en formato de datos
MLOG obtenidos del VNA, y obtener una linea de
tendencia mediante regresion lineal

Creado por Dixon Leon Masache

'''

numero_puntos=1001 #Definido en los parámetros de configuración del VNA
(Resolución de la gráfica)
cantidad_muestras=30 #Definido en los parámetros de configuración del VNA
(Bucle for para iteraciones)

def matriz(nombreArchivo):
    datos_medidos=[]
    datos_sin_ceros=[]
    linea_sin_ceros=[]
    #Se abre el archivo y se lee cada linea
    with open(nombreArchivo) as archivo:
        #se lee cada linea y se la limpia de espacios en blanco y saltos de
        linea
        for linea in archivo:
            linea=linea.rstrip()
            linea=linea.lstrip()
            linea=linea.split(sep=',') #creacion de lista usando de separador
            la coma
            if len(linea) == (2*numero_puntos)+1: #Si se crea un elemento de
            lista de solo espacio, se lo quita
                linea.pop(0)
            for i in range(len(linea)): #Se convierte los strings a flotante
                linea[i]=float(linea[i])
            datos_medidos.append(linea)
```

```

archivo.close()

cantidad_iteraciones=len(datos_medidos)
#se eliminan los elementos que contienen ceros
for i in range(cantidad_iteraciones):
    linea_sin_ceros=[j for j in datos_medidos[i] if j != 0]
    datos_sin_ceros.append(linea_sin_ceros)

matriz_metro=np.array(datos_sin_ceros)
matriz_metro_reshape=matriz_metro.reshape(numero_puntos,cantidad_muestras
) #Se reescala para que quede una matriz de 1001x30
return matriz_metro_reshape

#Funcion para graficar
'''
Recibe el arreglo de distancia para el eje X, y para el eje Y recibe
la matriz de perdidas de mediciones, el arreglo de perdidas promedio, y el
titulo
del grafico
'''
def graficar(distancia, distancia_regresion, L_Mediciones, L_Promedio,
ecuacion_reg, titulo_freq, anotacion1, anotacion2):
    plt.figure(figsize=(19.2,10.8))
    plt.plot(distancia, L_Mediciones, "b.", markersize=10)
    plt.plot(distancia, L_Promedio, "r.", markersize=8 ,label='Valor
promedio')
    plt.plot(distancia_regresion, ecuacion_reg, color='black',
label=('Regresion lineal'))
    plt.xlabel("Distancia (m)")
    plt.ylabel("Pérdida de propagación [dB]")
    plt.legend()
    plt.title(titulo_freq)
    plt.text(6,40, anotacion1)
    plt.text(6,36, anotacion2)
    #plt.xlim(0,10)
    plt.ylim(20,100)
    #plt.xscale("log")
    plt.show()

#Se aplica la funcion para leer los archivos de cada metro
matriz1metro=matriz("Mediciones corregidas 1 metro magnitud db.txt")
matriz2metro=matriz("Mediciones corregidas 2 metro magnitud db.txt")
matriz3metro=matriz("Mediciones corregidas 3 metro magnitud db.txt")
matriz4metro=matriz("Mediciones corregidas 4 metro magnitud db.txt")

```

```

matriz5metro=matriz("Mediciones corregidas 5 metro magnitud db.txt")
matriz6metro=matriz("Mediciones corregidas 6 metro magnitud db.txt")
matriz7metro=matriz("Mediciones corregidas 7 metro magnitud db.txt")
matriz8metro=matriz("Mediciones corregidas 8 metro magnitud db.txt")
matriz9metro=matriz("Mediciones corregidas 9 metro magnitud db.txt")

#Vector de distancias para la grafica en plot
dist=np.linspace(1,9,9)

#Vector de frecuencias para saber el incremento entre frecuencias y graficar
la fase
freq=np.linspace(5800000000,6400000000,numero_puntos)

...

Explicacion:
Cada matrizXmetro tiene una matriz de (FILAS=frecuencias,
COLUMNAS=Mediciones)
Y son 1001 frecuencias desde 5.80 GHz hasta 6.40 GHz.
En cada frecuencia se tomaron 30 muestras.
Por lo tanto, cada frecuencia equivale a cada fila de las matricesXmetro.
Dichos valores estan en terminos de Potencia recibida de la ecuacion
de presupuesto de potencia.

Pout = Pt + Gt - Lt - Lfs - Lm + Gr - Lr

Dado que no se consideran las atenuaciones por cable debido a que se
calibró en dos puertos el VNA, y resumiendo Lfs (Perdidas de espacio libre)
y Lm (Perdidas miscelaneas) en una sola L, se obtiene:
Pt = 0 dBm
Gt = 18 dBi
Gr = 18 dBi
L = (Pout - 36 dB)*-1
L = Pout*-1 + 36

...

''''***** INICIO DE LA GESTION CON LA MATRIZ GENERAL
*****''''

...

La idea es promediar las columnas de todas las matricesXmetro,
y de esta forma obtener una sola matriz donde sus filas correspondan
a los metros, o sea 9 filas, y sus columnas sean las 30 mediciones
promediadas

```

```

'''
#Creación de matriz GENERAL
Matriz_general_Loss=np.array([
    (np.mean(matriz1metro,0)*-1 + 36),
    (np.mean(matriz2metro,0)*-1 + 36),
    (np.mean(matriz3metro,0)*-1 + 36),
    (np.mean(matriz4metro,0)*-1 + 36),
    (np.mean(matriz5metro,0)*-1 + 36),
    (np.mean(matriz6metro,0)*-1 + 36),
    (np.mean(matriz7metro,0)*-1 + 36),
    (np.mean(matriz8metro,0)*-1 + 36),
    (np.mean(matriz9metro,0)*-1 + 36),
])

#Calculo del valor medio por cada metro
Matriz_general_loss_mean = np.mean(Matriz_general_Loss,1)

#Calculo de la varianza por cada metro
Matriz_general_Loss_std = np.std(Matriz_general_Loss,1)

#Regresion lineal de la Matriz General de perdidas
regresion_matriz_general=LinearRegression() #instancia de la regresion
regresion_matriz_general.fit(dist.reshape(-1,1), Matriz_general_loss_mean)
pendiente_matriz_general = float(regresion_matriz_general.coef_)
intercepto_matriz_general = float(regresion_matriz_general.intercept_)
anotacion_matriz_general_pendiente_intercepto = 'Pendiente
='+str(pendiente_matriz_general) + ', Intercepto =' +
str(intercepto_matriz_general)
print(anotacion_matriz_general_pendiente_intercepto)
distancia_regresion_matriz_general=np.linspace(1,10,10)

#Obtención del Coeficiente de Determinación R2
R2_matriz_general=regresion_matriz_general.score(dist.reshape(-
1,1),Matriz_general_loss_mean)
anotacion_matriz_general_R2 = 'Coeficiente de determinacion R^2:
'+str(R2_matriz_general)
print(anotacion_matriz_general_R2)

#Obtencion del error cuadratico medio
prediccion_matriz_general = regresion_matriz_general.predict(dist.reshape(-
1,1))

```



```

error_cuadratico_medio_matriz_general =
mean_squared_error(y_true=Matriz_general_loss_mean,
y_pred=prediccion_matriz_general)
raiz_error_cuadratico_medio_matriz_general =
np.sqrt(error_cuadratico_medio_matriz_general)
print('Error cuadratico medio usando metodo predict:
'+str(error_cuadratico_medio_matriz_general))
print('Raiz de Error Cuadratico medio usando metodo predict:
'+str(raiz_error_cuadratico_medio_matriz_general)+ '\n')

#Obtención de la ecuación final obtenida por regresion lineal
ecuacion_regresion_matriz_general=10*pendiente_matriz_general*np.log10(distan
cia_regresion_matriz_general)+intercepto_matriz_general-
error_cuadratico_medio_matriz_general

#Grafico de dispersión solamente
plt.figure(figsize=(19.2,10.8))
plt.plot(dist, Matriz_general_Loss, "b.", markersize=10)
plt.plot(dist, Matriz_general_loss_mean, "r.", markersize=8 ,label='Valor
promedio')
plt.xlabel("Distancia (m)")
plt.ylabel("Pérdida de propagación [dB]")
plt.legend()
plt.title("Pérdidas de propagación vs distancia")
#plt.xlim(0,10)
plt.ylim(20,100)
#plt.xscale("log")
plt.show()

#Representacion grafica de las perdidas de camino desde Frecuencia 5.80 GHz -
6.40 GHz incluyendo el modelo
graficar(
    dist, distancia_regresion_matriz_general, Matriz_general_Loss,
    Matriz_general_loss_mean, ecuacion_regresion_matriz_general,
    "Frecuencias 5.80 GHz - 6.40 GHz",
    anotacion_matriz_general_pendiente_intercepto,
    anotacion_matriz_general_R2
)

'''***** INICIO COMPARACIÓN DE MODELOS *****'''

#Modelos para comparar los resultados obtenidos

```

```

#Modelo obtenido
ecuacion_regresion_matriz_general=10*pendiente_matriz_general*np.log10(distan
cia_regresion_matriz_general)+intercepto_matriz_general-
error_cuadratico_medio_matriz_general
#Obtencion del error cuadratico medio del modelo obtenido respecto a las
mediciones reales
prediccion_modelo_obtenido =
10*pendiente_matriz_general*np.log10(dist)+intercepto_matriz_general-
error_cuadratico_medio_matriz_general
error_cuadratico_medio_nuevo =
mean_squared_error(y_true=Matriz_general_loss_mean,
y_pred=prediccion_modelo_obtenido)
raiz_error_cuadratico_medio_nuevo = np.sqrt(error_cuadratico_medio_nuevo)
print('Error cuadratico medio Modelo Obtenido:
'+str(error_cuadratico_medio_nuevo))
print('Raiz de Error Cuadratico medio Modelo Obtenido:
'+str(raiz_error_cuadratico_medio_nuevo)+ '\n')

#Modelo Keenan-Motley
modelo_keenan = 37.07 + 20*np.log10(distancia_regresion_matriz_general)
#Obtencion del error cuadratico medio del modelo Keenan respecto a las
mediciones reales
prediccion_modelo_keenan = 37.07 + 20*np.log10(dist)
error_cuadratico_medio_keenan =
mean_squared_error(y_true=Matriz_general_loss_mean,
y_pred=prediccion_modelo_keenan)
raiz_error_cuadratico_medio_keenan = np.sqrt(error_cuadratico_medio_keenan)
print('Error cuadratico medio Keenan: '+str(error_cuadratico_medio_keenan))
print('Raiz de Error Cuadratico medio Keenan:
'+str(raiz_error_cuadratico_medio_keenan)+ '\n')

#Modelo ITU interiores
modelo_ITU = 50.985 + 14.6*np.log10(distancia_regresion_matriz_general)
#Obtencion del error cuadratico medio del modelo ITU interiores respecto a
las mediciones reales
prediccion_modelo_itu = 50.985 + 14.6*np.log10(dist)
error_cuadratico_medio_itu =
mean_squared_error(y_true=Matriz_general_loss_mean,
y_pred=prediccion_modelo_itu)
raiz_error_cuadratico_medio_itu = np.sqrt(error_cuadratico_medio_itu)
print('Error cuadratico medio ITU interiores:
'+str(error_cuadratico_medio_itu))
print('Raiz de Error Cuadratico medio ITU interiores:
'+str(raiz_error_cuadratico_medio_itu)+ '\n')

```

```

#Modelo Espacio Libre
modelo_free_space= 48.565 + 20*np.log10(distancia_regresion_matriz_general)
#Obtencion del error cuadratico medio del modelo Espacio Libre respecto a las
mediciones reales
prediccion_modelo_free_space = 48.565 + 20*np.log10(dist)
error_cuadratico_medio_free_space =
mean_squared_error(y_true=Matriz_general_loss_mean,
y_pred=prediccion_modelo_free_space)
raiz_error_cuadratico_medio_free_space =
np.sqrt(error_cuadratico_medio_free_space)
print('Error cuadratico medio Espacio Libre:
'+str(error_cuadratico_medio_free_space))
print('Raiz de Error Cuadratico medio Espacio Libre:
'+str(raiz_error_cuadratico_medio_free_space))

#En escala lineal la distancia
plt.figure(figsize=(19.2,10.8))
plt.plot(dist, Matriz_general_Loss, "b.", markersize=8)
plt.plot(dist, Matriz_general_loss_mean, "ro", label='Valor promedio')
plt.plot(distancia_regresion_matriz_general,
ecuacion_regresion_matriz_general, 'k-', label=('Nuevo Modelo'))
plt.plot(distancia_regresion_matriz_general,modelo_keenan, marker='d',
color='green', label='Modelo Keenan - Motley')
plt.plot(distancia_regresion_matriz_general, modelo_ITU, marker='v',
color='blue', label='Modelo ITU interiores')
plt.plot(distancia_regresion_matriz_general, modelo_free_space, marker='x',
color='orange', label='Modelo Espacio Libre')
plt.xlabel("Distancia (m)")
plt.ylabel("Pérdida de propagación [dB]")
plt.legend()
plt.title('Comparación de modelos de estimación de pérdidas')
#plt.xlim(0,10)
plt.ylim(20,100)
plt.show()

#En escala logaritmica la distancia
plt.figure(figsize=(19.2,10.8))
plt.plot(dist, Matriz_general_Loss, "b.", markersize=8)
plt.plot(dist, Matriz_general_loss_mean, "ro", label='Valor promedio')
plt.plot(distancia_regresion_matriz_general,
ecuacion_regresion_matriz_general, 'k-', label=('Nuevo Modelo'))
plt.plot(distancia_regresion_matriz_general,modelo_keenan, marker='d',
color='green', label='Modelo Keenan - Motley')

```

```

plt.plot(distancia_regresion_matriz_general, modelo_ITU, marker='v',
color='blue', label='Modelo ITU interiores')
plt.plot(distancia_regresion_matriz_general, modelo_free_space, marker='x',
color='orange', label='Modelo Espacio Libre')
plt.xlabel("Distancia log(m)")
plt.ylabel("Pérdida de propagación [dB]")
plt.legend()
plt.title('Comparación de modelos de estimación de pérdidas')
#plt.xlim(0,10)
plt.ylim(20,100)
plt.xscale("log")
plt.show()

'''*****INICIO DE MEDICION DE PERDIDA EN MATERIALES*****'''
numero_puntos=30 #Definido en los parámetros de configuración del VNA (Resolución
de la gráfica)
cantidad_muestras=500 #Definido en los parámetros de configuración del VNA (Bucle
for para iteraciones)

matriz1metro_MDF=matriz("Mediciones 1 metro Division MDF.txt")
matriz1metro_MDF=(matriz1metro_MDF)*-1 + 36 - 52
matriz1metro_MDF_mean=np.mean(matriz1metro_MDF,1)
matriz1metro_MDF_mean_mean=np.mean(matriz1metro_MDF_mean)

matriz1metro_pared=matriz("Mediciones 1 metro Division Pared.txt")
matriz1metro_pared=(matriz1metro_pared)*-1 + 36 - 52
matriz1metro_pared_mean=np.mean(matriz1metro_pared,1)
matriz1metro_pared_mean_mean=np.mean(matriz1metro_pared_mean)

matriz1metro_vidrio=matriz("Mediciones 1 metro Division Vidrio.txt")
matriz1metro_vidrio=(matriz1metro_vidrio)*-1 + 36 - 52
matriz1metro_vidrio_mean=np.mean(matriz1metro_vidrio,1)
matriz1metro_vidrio_mean_mean=np.mean(matriz1metro_vidrio_mean)

print(matriz1metro_MDF_mean_mean)
print(matriz1metro_pared_mean_mean)
print(matriz1metro_vidrio_mean_mean)

#Vector de distancias para la grafica en plot
dist=np.linspace(1,9,9)
#Vector de frecuencias para saber el incremento entre frecuencias y graficar la
fase
freq=np.linspace(5800000000,6400000000,numero_puntos)

#Grafico de dispersión solamente

```

```

#plt.figure(figsize=(19.2,10.8))
plt.plot(freq, matriz1metro_MDF, "b.", markersize=2)
plt.plot(freq, matriz1metro_MDF_mean, "ro", markersize=4 ,label='Valor promedio')
plt.xlabel("Frecuencia")
plt.ylabel("Atenuacion [dB]")
plt.legend()
plt.title("Atenuación a 1 metro con madera")
#plt.xlim(0,10)
plt.ylim(0,20)
#plt.xscale("log")
plt.show()

#Grafico de dispersión solamente
#plt.figure(figsize=(19.2,10.8))
plt.plot(freq, matriz1metro_pared, "b.", markersize=2)
plt.plot(freq, matriz1metro_pared_mean, "ro", markersize=4 ,label='Valor
promedio')
plt.xlabel("Frecuencia")
plt.ylabel("Atenuacion [dB]")
plt.legend()
plt.title("Atenuación a 1 metro con pared")
#plt.xlim(0,10)
plt.ylim(0,30)
#plt.xscale("log")
plt.show()

#Grafico de dispersión solamente
#plt.figure(figsize=(19.2,10.8))
plt.plot(freq, matriz1metro_vidrio, "b.", markersize=2)
plt.plot(freq, matriz1metro_vidrio_mean, "ro", markersize=4 ,label='Valor
promedio')
plt.xlabel("Frecuencia")
plt.ylabel("Atenuacion [dB]")
plt.legend()
plt.title("Atenuación a 1 metro con vidrio")
#plt.xlim(0,10)
plt.ylim(0,20)
#plt.xscale("log")
plt.show()

...

#Leer los archivos de FASE
#Se aplica la funcion para leer los archivos de cada metro
matriz1metro_fase=matriz("Mediciones corregidas 1 metro fase.txt")
matriz2metro_fase=matriz("Mediciones corregidas 2 metro fase.txt")

```

```

matriz3metro_fase=matriz("Mediciones corregidas 3 metro fase.txt")
matriz4metro_fase=matriz("Mediciones corregidas 4 metro fase.txt")
matriz5metro_fase=matriz("Mediciones corregidas 5 metro fase.txt")
matriz6metro_fase=matriz("Mediciones corregidas 6 metro fase.txt")
matriz7metro_fase=matriz("Mediciones corregidas 7 metro fase.txt")
matriz8metro_fase=matriz("Mediciones corregidas 8 metro fase.txt")
matriz9metro_fase=matriz("Mediciones corregidas 9 metro fase.txt")

#Creación de matriz FASE
Matriz_general_fase=np.array([
    (np.mean(matriz1metro_fase,1)),
    (np.mean(matriz2metro_fase,1)),
    (np.mean(matriz3metro_fase,1)),
    (np.mean(matriz4metro_fase,1)),
    (np.mean(matriz5metro_fase,1)),
    (np.mean(matriz6metro_fase,1)),
    (np.mean(matriz7metro_fase,1)),
    (np.mean(matriz8metro_fase,1)),
    (np.mean(matriz9metro_fase,1)),
])

plt.figure(figsize=(19.2,10.8))
plt.plot(freq, Matriz_general_fase[0])
plt.xlabel("Frecuencias")
plt.ylabel("Fase")
plt.title("Fase a 1 metro")
plt.ylim(-179,179)
plt.show()

'''
'''

''''***** FIN COMPARACIÓN DE MODELOS *****'''

''''***** FIN DE LA GESTION CON LA MATRIZ GENERAL *****'''

''''***** INICIO DEL ANALISIS EN CADA FRECUENCIA *****'''

'''
'''

```

```

#FRECUENCIA 5.80 GHz (FRECUENCIA INICIAL DEL MODELO)

#La potencia recibida (RSSI)
Pout_freq_580=np.array([
    matriz1metro[0],
    matriz2metro[0],
    matriz3metro[0],
    matriz4metro[0],
    matriz5metro[0],
    matriz6metro[0],
    matriz7metro[0],
    matriz8metro[0],
    matriz9metro[0]])
#Las perdidas de camino
L_freq_580=np.array([
    (matriz1metro[0]-6)*-1,
    (matriz2metro[0]-6)*-1,
    (matriz3metro[0]-6)*-1,
    (matriz4metro[0]-6)*-1,
    (matriz5metro[0]-6)*-1,
    (matriz6metro[0]-6)*-1,
    (matriz7metro[0]-6)*-1,
    (matriz8metro[0]-6)*-1,
    (matriz9metro[0]-6)*-1])

# Calculo del valor medio de las muestras
L_freq_580_mean=np.mean(L_freq_580,1)

#calculo de la varianza por cada 500 muestras
L_freq_580_std=np.std(L_freq_580,1)

#Regresion lineal
regresion_580=LinearRegression() #instancia de la regresion
regresion_580.fit(dist.reshape(-1,1), L_freq_580_mean)
pendiente_580 = float(regresion_580.coef_)
intercepto_580 = float(regresion_580.intercept_)
anotacion_580='pendiente =' +str(pendiente_580) + ', intercepto =' +
str(intercepto_580)
#print(anotacion_580)
distancia_regresion_580=np.linspace(0,10,10)
ecuacion_regresion_580=pendiente_580*distancia_regresion_580+intercepto_580

#Obtención del Coeficiente de Determinación R2
R2_580=regresion_580.score(dist.reshape(-1,1),L_freq_580_mean)
anotacion2_580='Coeficiente R^2: ' +str(R2_580)

```

```

#print(anoacion2_580)

#Representacion grafica de las perdidas de camino
graficar(
    dist, distancia_regresion_580,L_freq_580,
    L_freq_580_mean, ecuacion_regresion_580, "Frecuencia 5.80 GHz",
    anoacion_580, anoacion2_580
)

#Comparacion de modelos de estimacion de perdidas en la frecuencia 5.80 GHz

modelo_keenan = 37 + 2*distancia_regresion_580
modelo_ITU = 47.72 + 2*distancia_regresion_580
modelo_free_space = 48.2 + 2*distancia_regresion_580
#Grafico con distancia en escala lineal
plt.figure(figsize=(19.2,10.8))
plt.plot(dist, L_freq_580, "b.", markersize=3)
plt.plot(dist, L_freq_580_mean, "ro", label='Valor promedio')
plt.plot(distancia_regresion_580, ecuacion_regresion_580, 'k-', label=('Nuevo
Modelo'))
plt.plot(distancia_regresion_580,modelo_keenan, marker='d',
color='darkgreen', label='Modelo Keenan - Motley')
plt.plot(distancia_regresion_580, modelo_ITU, marker='v', color='blue',
label='Modelo ITU interiores')
plt.xlabel("Distancia (m)")
plt.ylabel("Pérdida de propagación [dB]")
plt.legend()
plt.title('Comparación de modelos de estimación de pérdidas')
#plt.xlim(0,10)
plt.ylim(0,100)
plt.show()

#Grafico con distancia en escala logaritmica
plt.figure(figsize=(19.2,10.8))
plt.plot(dist, L_freq_580, "b.", markersize=3)
plt.plot(dist, L_freq_580_mean, "ro", label='Valor promedio')
plt.plot(distancia_regresion_580, ecuacion_regresion_580, 'k-', label=('Nuevo
Modelo'))
plt.plot(distancia_regresion_580,modelo_keenan, marker='d',
color='darkgreen', label='Modelo Keenan - Motley')
plt.plot(distancia_regresion_580, modelo_ITU, marker='v', color='blue',
label='Modelo ITU interiores')
plt.xlabel("Distancia log(m)")
plt.ylabel("Pérdida de propagación [dB]")
plt.legend()

```



```

plt.title('Comparación de modelos de estimación de pérdidas')
#plt.xlim(0,10)
plt.ylim(0,100)
plt.xscale("log")
plt.show()

#FRECUENCIA 6.11 GHz (FRECUENCIA CENTRAL DEL BARRIDO)

#La potencia recibida (RSSI)
Pout_freq_611=np.array([
    matriz1metro[500],
    matriz2metro[500],
    matriz3metro[500],
    matriz4metro[500],
    matriz5metro[500],
    matriz6metro[500],
    matriz7metro[500],
    matriz8metro[500],
    matriz9metro[500]])
#Las perdidas de camino
L_freq_611=np.array([
    (matriz1metro[500]-6)*-1,
    (matriz2metro[500]-6)*-1,
    (matriz3metro[500]-6)*-1,
    (matriz4metro[500]-6)*-1,
    (matriz5metro[500]-6)*-1,
    (matriz6metro[500]-6)*-1,
    (matriz7metro[500]-6)*-1,
    (matriz8metro[500]-6)*-1,
    (matriz9metro[500]-6)*-1])

# Calculo del valor medio de las muestras
L_freq_611_mean=np.mean(L_freq_611,1)

#Calculo de la varianza por cada 500 muestras
L_freq_611_std=np.std(L_freq_611,1)

#Regresion lineal
regresion_611=LinearRegression() #instancia de la regresion
regresion_611.fit(dist.reshape(-1,1), L_freq_611_mean)

```

```

pendiente_611 = float(regresion_611.coef_)
intercepto_611 = float(regresion_611.intercept_)
anotacion_611='pendiente =' +str(pendiente_611) + ', intercepto =' +
str(intercepto_611)
#print('pendiente =' +str(pendiente_611) + ', intercepto =' +
str(intercepto_611))
distancia_regresion_611=np.linspace(0,10,10)
ecuacion_regresion_611=pendiente_611*distancia_regresion_611+intercepto_611

#Obtención del Coeficiente de Determinación R2
R2_611=regresion_611.score(dist.reshape(-1,1),L_freq_611_mean)
anotacion2_611='Coeficiente R^2: ' +str(R2_611)
#print('Coeficiente R^2: ' +str(R2_611))

#Representacion grafica de las perdidas de camino
graficar(
    dist, distancia_regresion_611,L_freq_611,
    L_freq_611_mean, ecuacion_regresion_611, "Frecuencia 6.11 GHz",
    anotacion_611, anotacion2_611
)

#Comparacion de modelos de estimacion de perdidas en la frecuencia 6.11 GHz

modelo_keenan = 37 + 2*distancia_regresion_611
modelo_ITU = 47.72 + 2*distancia_regresion_611
modelo_free_space = 48.2 + 2*distancia_regresion_611

#Grafico con distancia en escala lineal
plt.figure(figsize=(19.2,10.8))
plt.plot(dist, L_freq_611, "b.", markersize=3)
plt.plot(dist, L_freq_611_mean, "ro", label='Valor promedio')
plt.plot(distancia_regresion_611, ecuacion_regresion_611, 'k-', label=('Nuevo
Modelo'))
plt.plot(distancia_regresion_611,modelo_keenan, marker='d',
color='darkgreen', label='Modelo Keenan - Motley')
plt.plot(distancia_regresion_611, modelo_ITU, marker='v', color='blue',
label='Modelo ITU interiores')
plt.xlabel("Distancia (m)")
plt.ylabel("Pérdida de propagación [dB]")
plt.legend()
plt.title('Comparación de modelos de estimación de pérdidas')
#plt.xlim(0,10)
plt.ylim(0,100)
plt.show()

```

```

#Grafico con distancia en escala logaritmica
plt.figure(figsize=(19.2,10.8))
plt.plot(dist, L_freq_611, "b.", markersize=3)
plt.plot(dist, L_freq_611_mean, "ro", label='Valor promedio')
plt.plot(distancia_regresion_611, ecuacion_regresion_611, 'k-', label=('Nuevo
Modelo'))
plt.plot(distancia_regresion_611,modelo_keenan, marker='d',
color='darkgreen', label='Modelo Keenan - Motley')
plt.plot(distancia_regresion_611, modelo_ITU, marker='v', color='blue',
label='Modelo ITU interiores')
plt.xlabel("Distancia log(m)")
plt.ylabel("Pérdida de propagación [dB]")
plt.legend()
plt.title('Comparación de modelos de estimación de pérdidas')
#plt.xlim(0,10)
plt.ylim(0,100)
plt.xscale("log")
plt.show()

#FRECUENCIA 6.40 GHz (FRECUENCIA FINAL DEL BARRIDO)

#La potencia recibida (RSSI)
Pout_freq_640=np.array([
    matriz1metro[1000],
    matriz2metro[1000],
    matriz3metro[1000],
    matriz4metro[1000],
    matriz5metro[1000],
    matriz6metro[1000],
    matriz7metro[1000],
    matriz8metro[1000],
    matriz9metro[1000]])
#Las perdidas de camino
L_freq_640=np.array([
    (matriz1metro[1000]-6)*-1,
    (matriz2metro[1000]-6)*-1,
    (matriz3metro[1000]-6)*-1,
    (matriz4metro[1000]-6)*-1,
    (matriz5metro[1000]-6)*-1,
    (matriz6metro[1000]-6)*-1,
    (matriz7metro[1000]-6)*-1,
    (matriz8metro[1000]-6)*-1,
    (matriz9metro[1000]-6)*-1])

```

```

# Calculo del valor medio de las muestras
L_freq_640_mean=np.mean(L_freq_640,1)

#Calculo de la varianza por cada 500 muestras
L_freq_640_std=np.std(L_freq_640,1)

#Regresion lineal
regresion_640=LinearRegression() #instancia de la regresion
regresion_640.fit(dist.reshape(-1,1), L_freq_640_mean)
pendiente_640 = float(regresion_640.coef_)
intercepto_640 = float(regresion_640.intercept_)
anotacion_640='pendiente =' +str(pendiente_640) + ', intercepto =' +
str(intercepto_640)
#print(anotacion_640)
distancia_regresion_640=np.linspace(0,10,10)
ecuacion_regresion_640=pendiente_640*distancia_regresion_640+intercepto_640

#Obtención del Coeficiente de Determinación R2
R2_640=regresion_640.score(dist.reshape(-1,1),L_freq_640_mean)
anotacion2_640='Coeficiente R^2: ' +str(R2_640)
#print(anotacion2_640)

#Representacion grafica de las perdidas de camino
graficar(
    dist, distancia_regresion_640,L_freq_640,
    L_freq_640_mean, ecuacion_regresion_640, "Frecuencia 6.40 GHz",
    anotacion_640, anotacion2_640
)

#Comparacion de modelos de estimacion de perdidas en la frecuencia 6.40 GHz

modelo_keenan = 37 + 2*distancia_regresion_640
modelo_ITU = 47.72 + 2*distancia_regresion_640
modelo_free_space = 48.2 + 2*distancia_regresion_640
#Grafico con distancia en escala lineal
plt.figure(figsize=(19.2,10.8))
plt.plot(dist, L_freq_640, "b.", markersize=3)
plt.plot(dist, L_freq_640_mean, "ro", label='Valor promedio')
plt.plot(distancia_regresion_640, ecuacion_regresion_640, 'k-', label=('Nuevo
Modelo'))
plt.plot(distancia_regresion_640, modelo_keenan, marker='d',
color='darkgreen', label='Modelo Keenan - Motley')
plt.plot(distancia_regresion_640, modelo_ITU, marker='v', color='blue',
label='Modelo ITU interiores')
plt.xlabel("Distancia (m)")

```

```

plt.ylabel("Pérdida de propagación [dB]")
plt.legend()
plt.title('Comparación de modelos de estimación de pérdidas')
#plt.xlim(0,10)
plt.ylim(0,100)
plt.show()

#Grafico con distancia en escala logaritmica
plt.figure(figsize=(19.2,10.8))
plt.plot(dist, L_freq_640, "b.", markersize=3)
plt.plot(dist, L_freq_640_mean, "ro", label='Valor promedio')
plt.plot(distancia_regresion_640, ecuacion_regresion_640, 'k-', label=('Nuevo
Modelo'))
plt.plot(distancia_regresion_640, modelo_keenan, marker='d',
color='darkgreen', label='Modelo Keenan - Motley')
plt.plot(distancia_regresion_640, modelo_ITU, marker='v', color='blue',
label='Modelo ITU interiores')
plt.xlabel("Distancia log(m)")
plt.ylabel("Pérdida de propagación [dB]")
plt.legend()
plt.title('Comparación de modelos de estimación de pérdidas')
#plt.xlim(0,10)
plt.ylim(0,100)
plt.xscale("log")
plt.show()
'''

```

Anexo 3. Hoja de datos de las antenas utilizadas



Photograph of Antenna (天线外观照片)



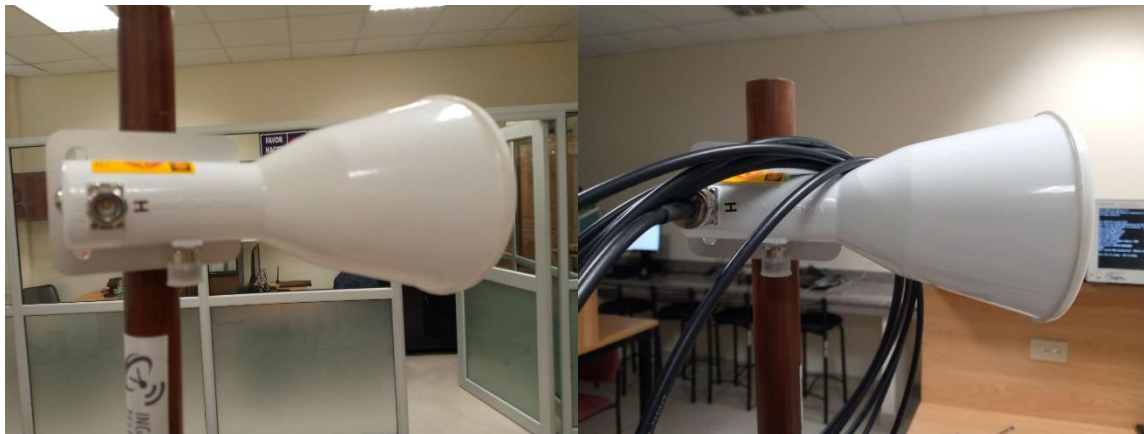
Antenna Electrical Specification (天线性能指标)

Freq. Range (频率)	5150-5850 MHz
Gain (增益)	18dBi
Horizontal Beamwidth (水平面波瓣宽度)	32°
Vertical Beamwidth (垂直面波瓣宽度)	22°
V.S.W.R (驻波比)	<1.5type; Max 2.0
Isolation(隔离)	>25dB
F/B Ratio (前后比)	>23dB
Nominal Impedance (输入阻抗)	50 ohm
Polarization (极化方式)	Vertical and Horizontal
Max Power (最大功率)	50 W
Connector (接口形式)	N Female
Size (外观尺寸)	φ 120x215mm
Weight (重量)	1kg
Radome Material (外罩材料)	ABS
Dia. Of Pole (抱杆直径)	Φ30-50mm

Anexo 4. Antenas usadas en las mediciones (vista frontal)



Anexo 5. Antenas usadas en las mediciones (vista lateral)



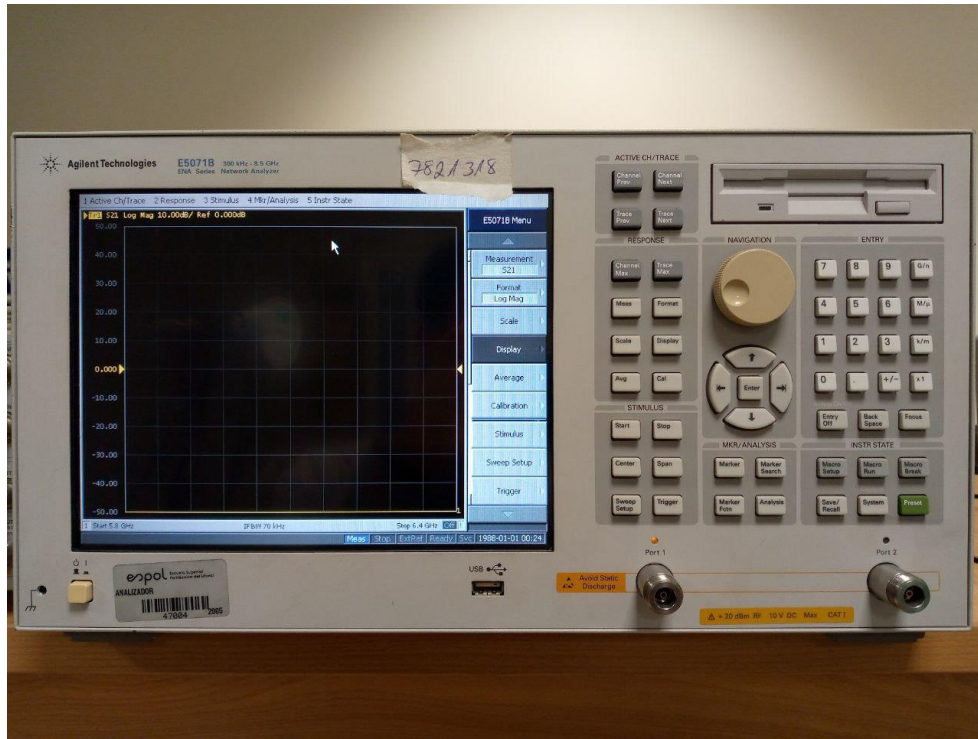
Anexo 6. Vista Real de la Configuración del esquema de medición



Anexo 7. Laboratorio de Radiocomunicaciones



Anexo 8. Panel Frontal del VNA



Anexo 9. Cargas utilizadas para calibrar los 2 puertos



Anexo 10. MSE y RMSE (dB) mostrados en consola

```
Pendiente =2.2197918743182283, Intercepto =53.22515490008664
Coeficiente de determinacion R^2: 0.9253857871046498
Error cuadratico medio usando metodo predict: 2.6486952492981115
Raiz de Error Cuadratico medio usando metodo predict: 1.6274812592770804

Error cuadratico medio Modelo Obtenido: 0.4174546633628699
Raiz de Error Cuadratico medio Modelo Obtenido: 0.6461073156704464

Error cuadratico medio Keenan: 222.0599489948353
Raiz de Error Cuadratico medio Keenan: 14.901676046500116

Error cuadratico medio ITU interiores: 21.46474959889917
Raiz de Error Cuadratico medio ITU interiores: 4.633006539915433

Error cuadratico medio Espacio Libre: 11.665002724157192
Raiz de Error Cuadratico medio Espacio Libre: 3.415406670391857
```

Anexo 11. Base de datos generadas

https://drive.google.com/drive/folders/117jWC5fi_1DTAG3c3SPKP1og77u6IGm0?usp=sharing