

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ESTUDIO DEL HANDOVER EN REDES LTE MEDIANTE MEDICIONES DE CAMPO

PROPUESTA DE UN ALGORITMO BÁSICO DE PREDICCIÓN PARA EL PROCESO DE HANDOVER

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TELECOMUNICACIONES**

JHON RAMIRO CANDO PUENAYAN

jhon.cando@epn.edu.ec

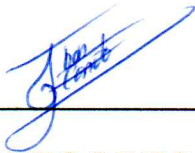
DIRECTOR: Ph.D. PABLO ANÍBAL LUPERA MORILLO

pablo.lupera@epn.edu.ec

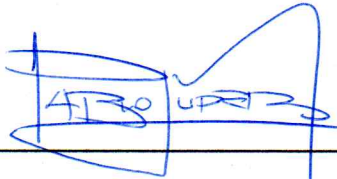
DMQ, abril 2023

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.




JHON RAMIRO CANDO PUENAYAN



PABLO ANÍBAL LUPERA MORILLON

CERTIFICACIONES

Yo, JHON RAMIRO CANDO PUENAYAN declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



JHON RAMIRO CANDO PUENAYAN

Certifico que el presente trabajo de integración curricular fue desarrollado por JHON RAMIRO CANDO PUENAYAN, bajo mi supervisión.



PABLO ANÍBAL LUPERA MORILLO
DIRECTOR

DEDICATORIA

Dedico esta tesis a mi familia, quienes siempre me han brindado un apoyo absoluto en todas las etapas de mi vida y nunca dejaron de creer en mí. Sin ellos este gran logro personal y profesional no lo habría sido posible, también dedico este trabajo a mi abuelita Rosa María Arteaga, que a pesar de que hoy en día ya no está con nosotros, siempre la llevo en mi corazón.

Jhon Cando

AGRADECIMIENTO

Primeramente, quiero expresar mi agradecimiento a Dios por haberme brindado la salud, la fortaleza y la sabiduría necesarias para culminar con esta etapa de mi vida. Su guía y protección han sido fundamentales, ya que me han llevado siempre a tomar las mejores decisiones en la persecución de mis objetivos.

Le agradezco a mi extraordinaria madre, Cecilia Puenayan, quien con su ejemplo de constancia y esfuerzo diario motivó mis ganas de crecer día a día para no rendirme por más turbias que se presentaran las cosas. También agradezco a mi padre, Ramiro Cando, quien con su atención y cuidado me ha permitido superar las circunstancias difíciles y continuar con mi camino.

Agradezco a mi hermano Alex y su familia, por el apoyo y confianza que me ofrecieron en todo momento. A mi hermano Julián, por preocuparse por mí y darme los ánimos suficientes para seguir adelante. A mi hermano David, por sus sabios consejos que supieron convertir posibles malas decisiones en momentos de reflexión.

Agradezco a mi director de tesis. El Ph.D. Pablo Lupera, por su valiosa orientación y apoyo durante esta etapa de investigación. Su retroalimentación, comentarios y sugerencias me ayudaron a mejorar mi tesis con éxito.

Agradezco a mis compañeros de investigación, Jonathan Villarreal y Patricio Aguagallo, por el tiempo y esfuerzo que dedicaron participando en el progreso de este proyecto.

También quiero agradecer a mis amigos Anderson, Graciela y Mauricio, quienes fueron el pilar más importante en el inicio de esta etapa académica. David, Jandry y Dario, con quienes compartí y perseguí el mismo objetivo, demostraron que las segundas familias sí existen y pueden estar conformadas por foráneos. A Lenin, Diego, Andrés, Alejandro, Moe, Jona y Leonardo, quienes me brindaron su apoyo y amistad incondicional e hicieron que todo este trayecto sea ameno.

Finalmente, quiero agradecer a aquellas personas que en el pasado formaron parte de mi vida, ya que tuvieron una gran influencia en mi personalidad y en la forma en la que percibo la vida.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VI
ABSTRACT.....	VII
1 INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS	2
1.3 ALCANCE	2
1.4 MARCO TEÓRICO	3
1.4.1 HERRAMIENTAS UTILIZADAS PARA LA RECOLECCIÓN DE DATOS.....	3
1.4.2 CARACTERÍSTICAS DEL TERMINAL MÓVIL.....	7
1.4.3 HERRAMIENTAS DE ANÁLISIS DE DATOS.....	9
2 METODOLOGÍA.....	12
2.1 DEFINICIÓN DE LOS DATOS A RECOLECTAR Y PROCESAR.....	12
2.1.1 OBJETIVOS DE LA RECOLECCIÓN DE DATOS.	12
2.1.2 IDENTIFICACIÓN Y DEFINICIÓN DE LOS PARÁMETROS QUE SE VAN A RECOLECTAR.....	12
2.2 DEFINICIÓN DE LA ZONA Y RUTAS A CONSIDERAR	14
2.2.1 DETERMINACIÓN DE LA UBICACIÓN GEOGRÁFICA DE LAS ESTACIONES BASE SELECCIONADAS	17
2.3 RUTAS CONSIDERADAS	19
2.3.1 PRIMERA RUTA – SECTOR SAN JUAN.....	19
2.3.2 SEGUNDA RUTA – SECTOR SAN JUAN.....	20

2.4	CRONOGRAMA DE LA RECOLECCIÓN DE DATOS	20
2.5	PROCESAMIENTO DE DATOS.....	21
2.5.1	GENERACIÓN DEL ATRIBUTO HAND_OVER	24
2.5.2	GENERACIÓN DEL ATRIBUTO DISTANCIA_BS	27
2.5.3	GENERACIÓN DEL ATRIBUTO AZIMUT	28
2.5.4	GENERACIÓN DEL ATRIBUTO SE ACERCA O SE ALEJA.....	30
2.6	MACHINE LEARNING	31
2.6.1	APRENDIZAJE SUPERVISADO	31
2.6.2	APRENDIZAJE NO SUPERVISADO	33
2.7	TÉCNICA DE CLASIFICACIÓN DE MACHINE LEARNING SELECCIONADA.....	34
2.8	DESCRIPCIÓN DE LA ETAPA PARA EL ANÁLISIS DE DATOS.....	34
2.8.1	PARÁMETROS ESCOGIDOS COMO VARIABLES DE ENTRADA.....	35
2.8.2	PARÁMETRO ESCOGIDO COMO VARIABLE DE SALIDA.....	35
2.8.3	DESCRIPCIÓN DE LA CODIFICACIÓN EN RSTUDIO.....	36
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	41
3.1	RESULTADOS.....	41
3.1.1	SECTOR SAN JUAN – RUTA 1.....	41
3.1.2	SECTOR SAN JUAN – RUTA 2.....	44
3.1.3	ALGORITMO BÁSICO DEL HANDOVER	46
3.2	CONCLUSIONES	50
3.3	RECOMENDACIONES	50
5	ANEXOS	55

RESUMEN

En este documento se realizan mediciones de parámetros de radiofrecuencia en el sector San Juan de Quito. Los datos recolectados a través de las herramientas de medición compatibles con Android: CellMapper, G-NetTrack y Net Monitor Lite, son analizados y procesados con el fin de proponer un algoritmo básico para predecir con algún nivel de probabilidad la celda hacia la que se dirige el terminal de usuario. El algoritmo básico deducido se evalúa con los mismos datos recopilados y se determina su efectividad en función de los resultados. Debe tenerse en cuenta que la capacidad del algoritmo para garantizar que se ejecuten los traspasos está condicionada a la cantidad de datos que se puedan recopilar y de las capacidades de las aplicaciones de medición.

Para la selección del algoritmo de traspaso, se deben tener en cuenta muchas métricas, como las condiciones de la red, las condiciones del terminal móvil y las condiciones climáticas, ya que al utilizar un canal inalámbrico el proceso de handover está sujeto a fenómenos como el ping-pong y fallos como traspasos de celdas fallidos, que degradan la calidad de la comunicación.

Se utiliza la técnica de aprendizaje supervisado Machine Learning ya que este tipo de técnica se aplica para obtener un modelo predictivo. Este enfoque se convierte en una solución efectiva para mejorar el rendimiento del traspaso de celdas celulares y aumentar la utilización de la red.

PALABRAS CLAVE: Machine Learning, handover, modelo.

ABSTRACT

In this document, radiofrequency parameters measurements are made in the San Juan sector from Quito. The data collected through the measurement tools compatible with Android: CellMapper, G-NetTrack, and Net Monitor Lite, are analyzed and processed in order to propose a basic algorithm to predict with some level of probability towards the cell to which it is directed the user terminal. The deduced basic algorithm is evaluated with the same data collected, and its effectiveness is determined based on the results. It must be taken into account that the ability of the algorithm to guarantee that handovers are executed depends on the amount of data that can be collected and the capabilities of the measurement applications.

For the selection of the handover algorithm, many metrics must be taken into account, such as network conditions, mobile terminal conditions, and weather conditions, since the handover process when using a wireless channel is subject to phenomena, such as ping-pong. And glitches, such as failed handovers, which degrade communication quality.

The Machine Learning supervised learning technique is used since this type of technique is applied to obtain a predictive model. This approach becomes an effective solution to improve cellular cell handover performance and increase network utilization.

KEYWORDS: Machine Learning, handover, model.

1 INTRODUCCIÓN

Las redes móviles se han vuelto esenciales para el funcionamiento de la sociedad actual, ya que permiten la comunicación instantánea y la transmisión de datos desde cualquier lugar y en cualquier momento. Estas redes móviles se han vuelto cada vez más importantes en la actualidad, ya que la mayoría de las personas poseen dispositivos móviles y dependen de ellos para mantenerse conectados a la red y así poder realizar sus tareas diarias [1].

Para mantener constante la transmisión de datos y voz entre estaciones móviles y estaciones base, se debe asegurar la continuidad del servicio sin interrupciones o caídas a través del proceso handover. Este proceso es una operación crítica en las redes de comunicaciones móviles, donde se espera que los usuarios puedan moverse sin problemas de una estación base a otra, sin pérdida de calidad en la señal y en la comunicación.

Las tecnologías en las que se involucra el handover son: GSM (Global System for Mobile Communications), CDMA (Code Division Multiple Access), LTE (Long-Term Evolution) y 5G (quinta generación de redes móviles). Cada tecnología tiene sus propias características y beneficios, y es utilizada por diferentes proveedores de servicios móviles en todo el mundo [2].

Debido a que el handover es un proceso crítico, es posible que lleve una mala gestión del mismo y por ende provocar la interrupción del servicio, la pérdida de calidad de la señal y la insatisfacción del usuario. Por tanto, es necesario contar con sistemas eficientes de handover para asegurar una transición fluida y sin problemas. En este sentido, se han desarrollado diferentes técnicas para optimizar el handover, incluyendo la monitorización constante de la calidad de la señal, la selección de la estación base adecuada y la predicción del handover basado en el movimiento del usuario y la calidad de la señal [3].

Existen diferentes tipos de handover, que se utilizan en diferentes situaciones. El handover suave es el más común, y se utiliza para cambiar de un punto de acceso a otro sin interrupciones en el servicio. El handover duro, por otro lado, puede interrumpir temporalmente el servicio mientras se produce el cambio de punto de acceso. Otras variantes incluyen el handover horizontal, aquel que se produce entre diferentes puntos de acceso en la misma red, y el handover vertical, que se produce entre diferentes redes [4].

Para la optimización del handover utilizamos Machine Learning, ya que es una herramienta que actualmente tiene gran impacto, gracias a esta herramienta podemos determinar modelos predictivos en este caso regresión lineal, a partir de las posibles relaciones

encontradas entre variables de entrada y variables de salida en el proceso de aprendizaje [5].

1.1 OBJETIVO GENERAL

Diseñar un algoritmo básico para predecir con algún nivel de probabilidad la celda hacia la que se dirige el terminal de usuario.

1.2 OBJETIVOS ESPECÍFICOS

1. Conocer el funcionamiento de las herramientas de recolección de datos para obtener con finura los parámetros de radio frecuencia a medir.
2. Plantear distintas rutas de medición de datos dentro del sector San Juan de Quito para la etapa de recolección de datos.
3. Depurar los datos brutos de las rutas para su posterior análisis.
4. Conocer el funcionamiento de las herramientas utilizadas para el análisis del conjunto de datos como R y R Studio, para implementar un algoritmo básico de predicción aplicando la técnica de Machine Learning de aprendizaje supervisado.

1.3 ALCANCE

En este proyecto se propone implementar en base a los datos de radio frecuencia recolectados en varios puntos geográficos de la ciudad de Quito, un algoritmo básico para predecir con algún nivel de probabilidad la celda hacia la que el terminal móvil se está trasladando y no necesariamente a la celda vecina con señal más fuerte como se ejecuta el proceso actualmente. El algoritmo básico se evaluará con los mismos datos recolectados y su capacidad será proporcional a la cantidad de datos que se pueda recolectar con las aplicaciones compatibles para Android.

Para realizar la recolección de datos, se utilizó las aplicaciones: CellMapper, G-NetTrack y Net Monitor Lite, las cuales otorgan diferentes datos de la señal móvil cada segundo, ya que ese es el valor de tiempo predeterminado para cada una de ellas. Además, es importante mencionar que las aplicaciones indicadas no mostraban información detallada de las estaciones base, pero si parámetros de radio frecuencia de la estación móvil como: rssi_strongest, rssi y rsrq [8] [10] [11].

En la primera fase, también conocida como recolección de datos, se han establecido dos rutas dentro del sector de San Juan para llevar a cabo la recolección de datos. Estos datos serán recolectados caminando a una velocidad promedio de aproximadamente 4.125 km/h. Se debe tener en cuenta los diferentes factores que afectan la recolección de datos, como

la capacidad de procesamiento del dispositivo móvil, ya que, si se tiene varias aplicaciones ejecutándose al mismo tiempo (en segundo plano), estas no recolectaran datos efectivamente y presentaran mayores errores en las mediciones [6].

Para la etapa de análisis y procesamiento de los datos recolectados, primero se ordenaron y depuraron los datos brutos medidos de las dos rutas. Posteriormente, para el modelo de predicción se empleará la herramienta de programación con enfoque estadístico Rstudio, en el cual se realizará la programación del modelo de regresión lineal, ya que es una técnica estadística importante en el campo del Machine Learning, pues se utiliza comúnmente en modelos de aprendizaje supervisado. En esta técnica se entrena un modelo de aprendizaje automático utilizando un conjunto de datos de entrenamiento que contiene valores de variables de entrada y salida con el fin de obtener un modelo para predecir la salida correspondiente a una entrada no vista [7].

En este trabajo, específicamente, se modelará la relación entre las variables independientes rssi_strongest, rssi y rsrq y variable dependiente hand_over, para predecir valores futuros de un handover Exitoso o Fallido. Mediante este modelo se obtendrá los coeficientes de una ecuación lineal que satisficará la relación entre las variables mencionadas. Seguidamente se evaluará la ecuación lineal obtenida con la herramienta Matlab para estimar el valor de datos predichos correctamente y datos predichos incorrectamente.

Finalmente, se incluye la codificación que determina la celda candidata más cercana a la que se puede conectar el terminal móvil en función de las distancias entre el terminal móvil y las estaciones base.

1.4 MARCO TEÓRICO

1.4.1 HERRAMIENTAS UTILIZADAS PARA LA RECOLECCIÓN DE DATOS

Para la recolección de datos de radio frecuencia utilizamos 3 aplicaciones:

- CellMapper
- G-NetTrack Lite
- Net Monitor Lite

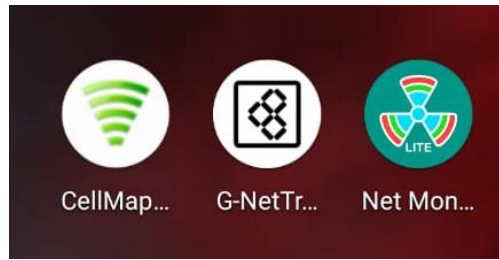


Figura 1.1: Aplicaciones que se van a utilizar

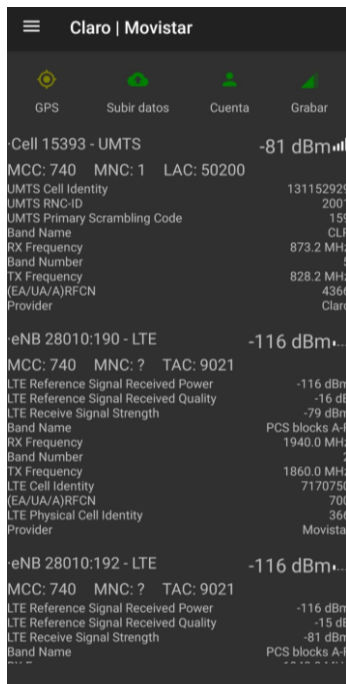
1.4.1.1 CELLMAPPER

Es una aplicación que permite al usuario recolectar parámetros de radio frecuencia en tiempo real, se caracteriza por localizar estaciones base de diferentes tecnologías, tales como: 2G, 3G, 4G, 4G+. Se puede descargar desde las aplicaciones de Android o de la página web de CellMapper [8]. Se debe tener en cuenta que para la recolección de datos la versión de Android del dispositivo debe ser mayor a 7 [8].

La aplicación comienza a recopilar datos una vez que, al ejecutarse, todos sus iconos estén de color verde y finalmente los datos se almacenan en un archivo csv. Los íconos de la aplicación son:

- GPS: Indica al usuario si el sistema de navegación está en correcto funcionamiento.
- Subir datos: Indica al usuario si está o no habilitada la subida de datos.
- Cuenta: Indica al usuario si el inicio de sesión fue correcto o no.
- Grabar: Indica al usuario si la grabación de los datos esta activa o no.
- En las siguientes gráficas se describen los principales menús y los datos que se reportan.

Parámetros de radio frecuencia:



Celda: Identificador de celda, se utiliza para diferenciar las celdas entre sí.

MCC: Código numérico de identificación de país.

MNC: Código numérico usado para identificar al operador de telefonía móvil.

RSRQ: Calidad de la señal recibida.

SINR: Relación señal interferencia más ruido.

RSRP: Potencia de señal recibida de referencia.

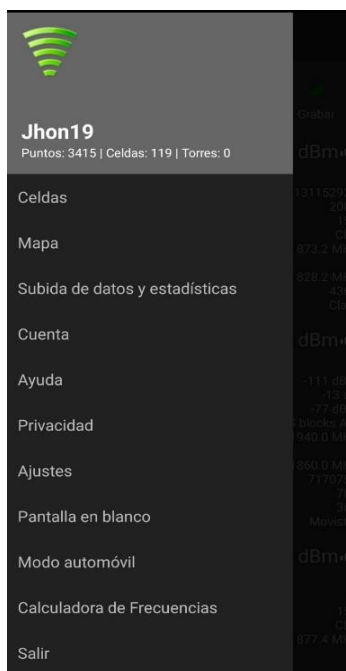
RSSI: Intensidad de la señal recibida.

TAC: Sección del código IMEI es empleada con el objetivo de identificar dispositivos inalámbricos.

LAC: Código de área de localización, identifica la zona de ubicación.

Figura 1.2: Pantalla principal de CellMapper y descripción de los principales parámetros de Radio Frecuencia [8]

Descripción del Menú:



Mapa: Permite ver la ubicación visual de todas las estaciones base en el mapa.

Subida de datos y estadísticas: Carga los datos de la estación base ya medida.

Cuenta: Permite rastrear cuántas torres ha registrado y ubica esas torres en el mapa web.

Ayuda: Despliega la guía con recomendaciones para la aplicación.

Privacidad: La aplicación recopila datos personales de los usuarios.

Ajustes: Permite realizar modificaciones en los ajustes principales.

Modo automóvil: Permite visualizar parámetros de movimiento.

Calculadora de Frecuencias: Permite calcular la frecuencia exacta a partir de los números de canal para las distintas redes.

Figura 1.3: Menú de CellMapper [8]

Las bandas de frecuencia disponibles en el Ecuador son las siguientes [9]:

- Ecuador: B4 - 1700 MHz en Uplink y 2100 en Downlink.
- Movistar: B2 - 1900 MHz.

1.4.1.2 G-NETTRACK LITE

Es una aplicación del sistema Android diseñada para dispositivos móviles, que permite monitorear la información de radio frecuencia de las celdas vecinas y el servicio de la red móvil para redes 2G, 3G, 4G, 5G. Esta aplicación a diferencia de otras permite hacer una representación visual de la ruta recorrida.

Para el funcionamiento de la aplicación la versión de Android puede ser desde 2.2 en adelante, pero se debe tener en cuenta que, para versiones anteriores de Android, la carga del archivo es de hasta al menos 128KB [10].

En esta aplicación la recolección de datos no es automática, ya que hay que iniciar el registro y posteriormente finalizarlo. Los datos se guardan en archivo txt y kml.

Descripción de las pestañas:



Célula: Muestra la información geográfica y las características de la red.

Vecinas: Indica la información sobre las mediciones de las celdas vecinas.

Mapa: Muestra una vista geográfica de las estaciones base de la red.

Info: Muestra información visual de los parámetros de radio frecuencia.

Coche: Muestra la información de las celdas con letras de fuente de mayor tamaño.

Figura 1.4: Pantalla principal, vecinas y mapa de G-NetTrack Lite [10]

1.4.1.3 NET MONITOR LITE

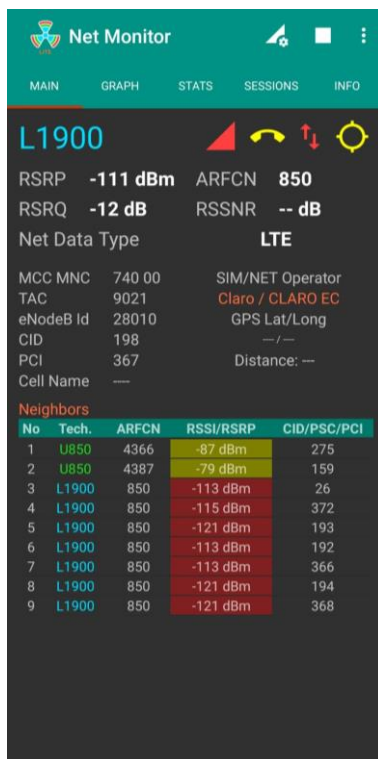
Esta aplicación se utiliza para monitorear parámetros técnicos de las redes celulares 2G, 3G, 4G en tiempo real. Dentro de sus funcionalidades permite recolectar la intensidad de la señal con su respectiva marca de tiempo y también parámetros para el análisis de redes celulares tales como RSSI, RSRQ y RSSNR.

Se considera una de las aplicaciones más completas ya que recolecta varios parámetros de radio frecuencia en comparación con las otras aplicaciones.

Para su funcionamiento debe manejar una versión mínima de Android de 4.3.

La recolección de los datos se realiza de manera automática al ejecutarse la aplicación y el archivo que se genera es de tipo csv [11].

Descripción de las pestañas:



Main: En la ventana principal podemos observar datos como: RSRP, ARFCN, RSRQ, RSSNR, tipo de red, MCC, eNodeB, CID entre otros.

Graph: Permite apreciar gráficamente la información de la celda.

Stats: Presenta gráficamente las estadísticas del nivel de las señales.

Sessions: Muestra las sesiones registradas en la aplicación con características como: Tiempo de inicio, duración, salto y área.

Info: Muestra la información general de la aplicación.

Figura 1.5: Pantalla principal de Net Monitor Lite [11]

1.4.2 CARACTERÍSTICAS DEL TERMINAL MÓVIL

El dispositivo empleado para la recolección de datos es: Redmi Note 9s con las siguientes características:

Tabla 1.1: Características del terminal móvil [12] [13]



Figura 1.6: Dispositivo móvil Redmi Note 9s

Características del modelo		
Nombre del modelo	Xiaomi Note 9S (M2003J6A1G Global version)	
Marca	Xiaomi	
Dimensiones y peso	166.9 x 76 x 8.8 mm - 209 gramos	
Sistema Operativo		
Versión del sistema	Android 10	
Actualización firmware	Android 11	
Interfaz de usuario	MIUI 11/12	
Procesador		
Chipset	64bits: Qualcomm Snapdragon 720G (SM7125)	
CPU	Octa-core, 2 procesadores: 2x 2.3GHz Kryo 465 Gold (Dual-core) 6x 1.8GHz Kryo 465 Silver (Hexa-core)	
Memoria		
Memoria RAM	6GB	
Memoria interna	128GB	
Memoria expansible	Hasta 512GB microSD	
Red móvil		
Doble SIM	Dual-SIM Standby	
Tarjeta SIM	2 ranuras (Doble-SIM) nano-SIM (4FF)	
Uplink /Downlink máxima	800/150 Mbps	
Redes móviles	2G, 3G, 4G	
Redes		
GSM	GSM 850 MHz (B5) GSM 900 MHz (B8)	GSM 1800 MHz (B3) GSM 1900 MHz (B2)
CDMA	CDMA 800 MHz (BC0)	
W-CDMA	W-CDMA 850 MHz (B5) W-CDMA 900 MHz (B8)	W-CDMA 1900 MHz (B2) W-CDMA 2100 MHz (B1)
LTE	LTE-FDD 850 MHz (B5) LTE-FDD 900 MHz (B8) LTE-FDD 1800 MHz (B3) LTE-FDD 2100 MHz (B1)	LTE-FDD 2600 MHz (B7) LTE-FDD 2300 MHz (B40) LTE-FDD 2500 MHz (B41)

Tecnologías de redes móviles		
Tecnologías de redes móviles	UTMS EDGE GPRS HSPA+	LTE Cat 12 EV-DO rev. A TD-SCDMA TD-HSDPA
Conectividad		
Bluetooth	5.0 + A2DP/LE	
WiFi	802.11 a/b/g/n/ac (2.4Ghz, 5Ghz) + MIMO WiFi Direct, hotspot	
GPS	A-GPS, GeoTagging, GLONASS, GALILEO, BeiDou	
Batería		
Batería	LiPo: polimeros de litio	
Capacidad batería	5020 mAh	
Cargador, watts	18W Carga rápida por cable	
Otras funciones		
Navegación web	HTML, XHTML, HTML5	
Antena	Antena interna	

1.4.3 HERRAMIENTAS DE ANÁLISIS DE DATOS

Las herramientas instaladas en el computador para el análisis del conjunto de datos en este trabajo se presentan en la Figura 1.7.

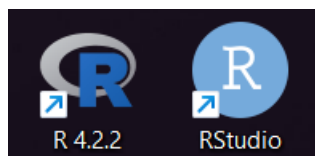


Figura 1.6: Herramientas de análisis de datos

1.4.3.1 Software R versión 4.2.2

Es una herramienta de programación gratuita y de código abierto utilizada principalmente para el estudio y análisis de conjunto de datos, así como para la generación de gráficos. Maneja un entorno sencillo y completo por lo que se considera una herramienta amigable para el usuario. Es compatible con sistemas operativos: Windows, UNIX y MacOS. Además, permite integrarse con distintas bases de datos como Excel.

R utiliza una interfaz de línea de comandos (CLI), que se presenta en la siguiente figura, en lugar de una interfaz gráfica [14] [15] [16].

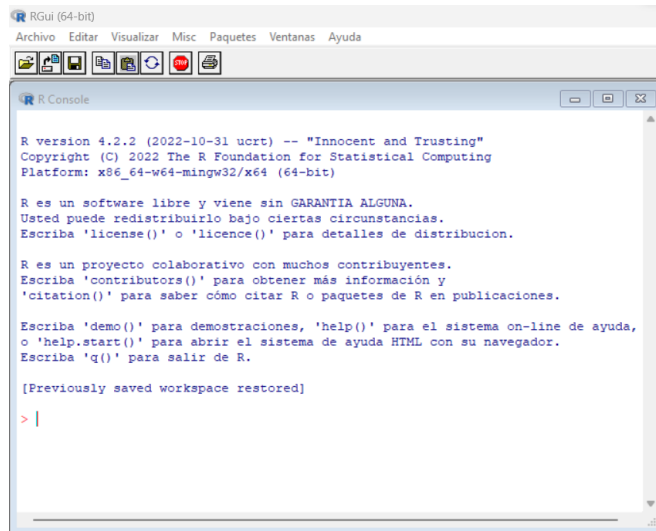


Figura 1.7: Interfaz de línea de comandos de R

1.4.3.2 Software RStudio

Es un entorno de desarrollo integrado (IDE) de código abierto destinado para el manejo del lenguaje de programación R con una interacción más fluida a través una interfaz gráfica. En este software, ofrece la posibilidad de utilizar una amplia variedad de modelos de aprendizaje automático, que abarcan desde la regresión lineal hasta las redes neuronales, al igual que en R [17].

Está conformado por: editor de sintaxis, entorno de trabajo, utilidades y consola como podemos ver en la Figura 10.

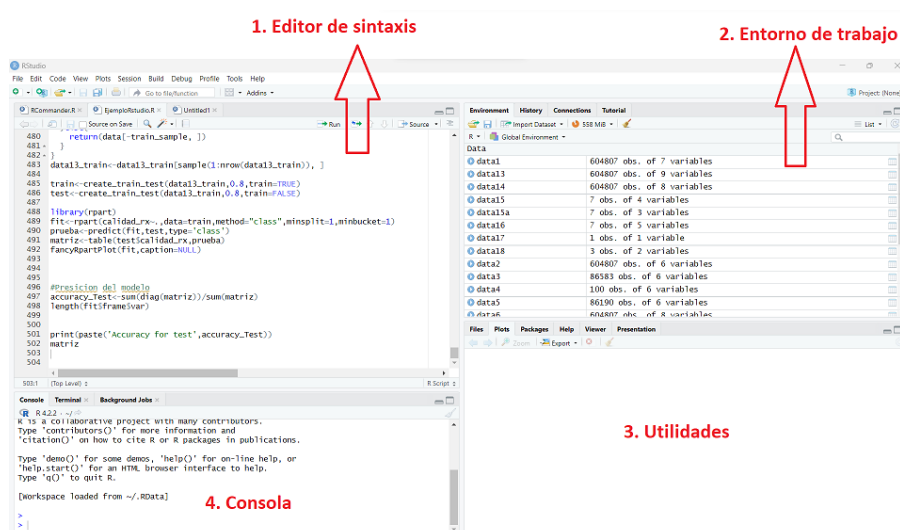


Figura 1.8: Partes del software Rstudio

Editor de sintaxis: Es el panel donde se crean y modifican los scripts para posteriormente ejecutarlos.

Entorno de trabajo: Consiste en el panel donde se visualiza el conjunto de datos y los objetos que se almacenan al ejecutar los análisis.

Utilidades: está conformada de varias subpestañas como:

- **Files:** permite la visualización del registro de archivos previamente accedidos en el directorio actual.
- **Plots:** muestra los gráficos generados.
- **Packages:** permite visualizar, descargar e instalar bibliotecas de software.
- **Help:** permite acceder a la ayuda sobre R.
- **Viewer:** Sirve para gráficas interactivas o animaciones.
- **Presentation:** Crea una nueva presentación R.

Consola: en ella se ejecuta los comandos realizados desde el editor de sintaxis [18] [19].

A continuación, se presentan algunas de las principales tareas que se pueden realizar en RStudio:

- Los usuarios pueden crear scripts, funciones y paquetes de R para realizar análisis estadísticos y manipulación de datos.
- RStudio permite importar datos en archivos CSV, Excel, SPSS, SAS y bases de datos como MySQL y SQLite. También se pueden conectar a fuentes de datos en línea como APIs o servicios web.
- A través de una amplia variedad de paquetes de R, es posible realizar análisis estadísticos y gráficos, desde análisis descriptivos hasta modelos predictivos más complejos.
- RStudio incluye una gran variedad de herramientas para visualizar datos, como gráficos de barras, histogramas, diagramas de dispersión, mapas y gráficos de líneas [20].

2 METODOLOGÍA

2.1 DEFINICIÓN DE LOS DATOS A RECOLECTAR Y PROCESAR

La recopilación de datos es una herramienta fundamental para verificar las hipótesis y suposiciones planteadas con el objetivo de validarlas o refutarlas. En el presente trabajo la etapa de recolección de datos de radio frecuencia por los distintos sectores de Quito ya mencionados, tiene la finalidad recopilar información para posteriormente diseñar un algoritmo básico para predecir con algún nivel de probabilidad la celda hacia la que se dirige el terminal de usuario.

2.1.1 OBJETIVOS DE LA RECOLECCIÓN DE DATOS.

- Interactuar con las distintas herramientas de recolección de datos, con la finalidad de familiarizarnos con su funcionamiento y entender su entorno de almacenamiento de variables.
- Planear distintas rutas por las cuales se realizará la recopilación de los datos de radio frecuencia.
- Realizar un cronograma para la planificación, organización y seguimiento de las rutas por las cuales se recopilará información.
- Realizar el análisis de los datos para el estudio del proceso de handover.

2.1.2 IDENTIFICACIÓN Y DEFINICIÓN DE LOS PARÁMETROS QUE SE VAN A RECOLECTAR.

Los parámetros que las aplicaciones permiten recolectar son:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
1	report	Feha	Hora	net_op	nc_rssi	stron	tech	mcc	lac_tac	node_id	cid	rssi	rsrq	rssnr	accuracy	lat	long
2	0	21/11/2022	14:04:11	Movistar	-104	LTE		740	9021	28235	191	-105	-18	-40	8	-0.20844	-78.5034
3	1	21/11/2022	14:04:12	Movistar	-104	LTE		740	9021	28235	191	-105	-18	-40	7	-0.20845	-78.5034
4	2	21/11/2022	14:04:13	Movistar	-104	LTE		740	9021	28235	191	-105	-18	-40	6	-0.20844	-78.5034
5	3	21/11/2022	14:04:14	Movistar	-101	LTE		740	9021	28235	191	-99	-17	-12	6	-0.20844	-78.5035
6	4	21/11/2022	14:04:15	Movistar	-101	LTE		740	9021	28235	191	-99	-17	-12	5	-0.20843	-78.5035
7	5	21/11/2022	14:04:16	Movistar	-101	LTE		740	9021	28235	191	-99	-17	-12	5	-0.20843	-78.5034
8	6	21/11/2022	14:04:17	Movistar	-99	LTE		740	9021	28235	191	-98	-17	-12	5	-0.20843	-78.5034
9	7	21/11/2022	14:04:18	Movistar	-99	LTE		740	9021	28235	191	-98	-17	-12	5	-0.20843	-78.5034
10	8	21/11/2022	14:04:19	Movistar	-99	LTE		740	9021	28235	191	-98	-17	-10	5	-0.20843	-78.5034
11	9	21/11/2022	14:04:20	Movistar	-99	LTE		740	9021	28235	191	-102	-18	-10	5	-0.20843	-78.5034
12	10	21/11/2022	14:04:21	Movistar	-99	LTE		740	9021	28235	191	-102	-18	-10	4	-0.20843	-78.5033
13	11	21/11/2022	14:04:22	Movistar	-99	LTE		740	9021	28235	191	-102	-18	-10	4	-0.20842	-78.5033
14	12	21/11/2022	14:04:23	Movistar	-101	LTE		740	9021	28235	191	-99	-15	-10	4	-0.20842	-78.5033
15	13	21/11/2022	14:04:24	Movistar	-101	LTE		740	9021	28235	191	-99	-15	-2	4	-0.20841	-78.5033
16	14	21/11/2022	14:04:25	Movistar	-101	LTE		740	9021	28235	191	-99	-15	-2	4	-0.2084	-78.5033
17	15	21/11/2022	14:04:26	Movistar	-99	LTE		740	9021	28235	191	-100	-15	-2	4	-0.2084	-78.5033
18	16	21/11/2022	14:04:27	Movistar	-99	LTE		740	9021	28235	191	-100	-15	-2	4	-0.20839	-78.5033
19	17	21/11/2022	14:04:28	Movistar	-99	LTE		740	9021	28235	191	-100	-15	-2	4	-0.20838	-78.5033
20	18	21/11/2022	14:04:29	Movistar	-99	LTE		740	9021	28235	191	-104	-17	-2	4	-0.20837	-78.5033
21	19	21/11/2022	14:04:30	Movistar	-99	LTE		740	9021	28235	191	-104	-17	0	4	-0.20835	-78.5033
22	20	21/11/2022	14:04:31	Movistar	-99	LTE		740	9021	28235	191	-104	-17	0	4	-0.20833	-78.5033
23	21	21/11/2022	14:04:32	Movistar	-100	LTE		740	9021	28235	191	-100	-20	0	4	-0.20832	-78.5033
24	22	21/11/2022	14:04:33	Movistar	-100	LTE		740	9021	28235	191	-100	-20	0	3	-0.2083	-78.5033

Figura 2.1: Parámetros de recolección de la aplicación

Fecha y Hora: Indica la fecha en la que se lleva a cabo la medición utilizando el formato día/mes/año, así como también la hora exacta en la que se realiza, utilizando el formato hora/minutos/segundos.

net_op_name: el nombre del operador de la red.

rssi_strongest: potencia más alta del vecino.

tech: modo de tecnología de red.

mcc (Mobile Country Code): es un código numérico de tres dígitos utilizado en la telefonía móvil para identificar el país de origen de un dispositivo móvil [21].

lac_tac (Location Area Code): es un código numérico utilizado en las redes móviles para identificar una Área de Localización. Se utiliza como referencia única para la ubicación de un suscriptor móvil [22].

node_id: especifica el identificador del nodo.

cid: es un identificador numérico único que se utiliza para identificar una celda específica en la red [23].

rssi (Received Signal Strength Indication): En español significa Indicador Potencia de la Señal Recibida, es una medida de la potencia de la señal recibida por un receptor. Se expresa en decibelios (dBm) y se utiliza comúnmente en redes wireless, como Wi-Fi, Bluetooth y redes celulares.

Este parámetro nos permite determinar la calidad de la señal y puede utilizarse para conocer es estado de la red. Se debe tener en cuenta que el RSSI no siempre es una medida precisa de la calidad de la señal, ya que puede verse afectado por factores como la interferencia, el ruido y la distancia. Si el valor medido está más cercano a 0 dBm, hay mejor señal, pero si el valor es -110 dBm o más negativo, hay señal débil o sin señal [24].

rsrq (Reference Signal Received Quality): En español significa Calidad de la Señal de Referencia Recibida, se refiere a una medida utilizada en redes celulares para evaluar la calidad de la señal de referencia recibida por un dispositivo móvil en relación con el ruido y la interferencia. Se expresa en decibelios (dB) y se utiliza para evaluar la calidad de la señal recibida en la banda de frecuencia de la red celular [24].

rssnr (Reference Signal Signal-to-Noise Ratio): En español significa Relación Señal-Ruido de la Señal de Referencia, se refiere a una medida de fiabilidad de la señal de referencia en una red móvil LTE. Un valor alto indica una señal de referencia fuerte y una

geográficas de las estaciones base. Por lo que para obtener este tipo de información se recurrirá a un método utilizando Google Maps, que se describirá más adelante.

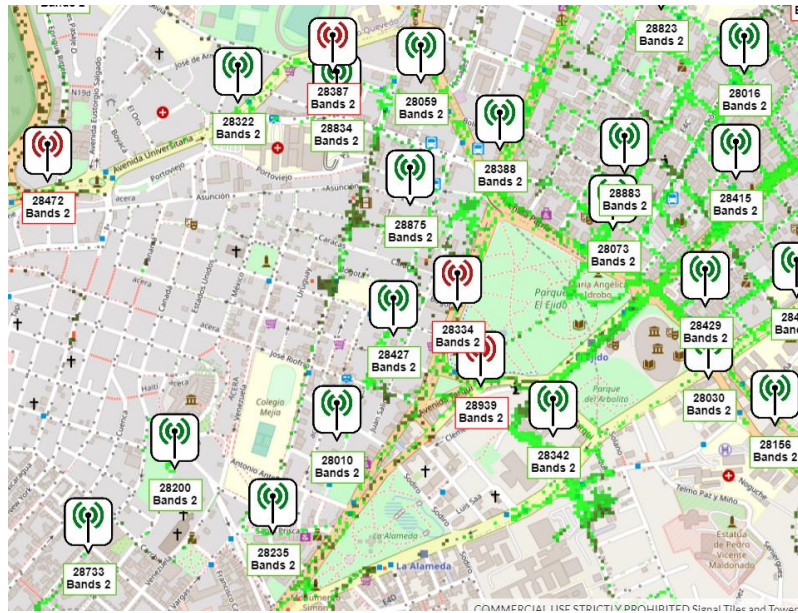


Figura 2.3: Sector San Juan con mapeo de estaciones base - CellMapper

Seguidamente, en la Figura: 2.4, se exhibe la ubicación visual de las ocho estaciones base seleccionadas para la recolección de parámetros de radio frecuencia para su posterior análisis.

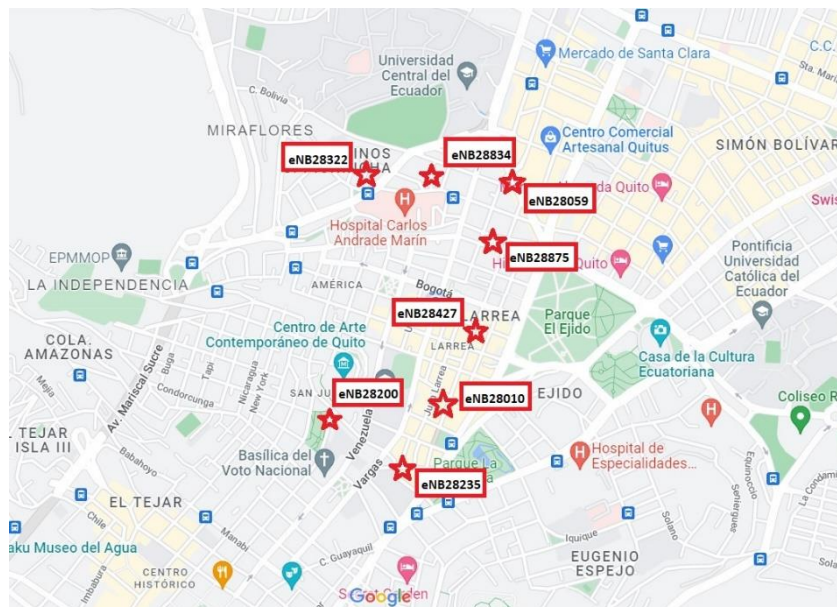










Figura 2.4: Ubicación de las 8 estaciones base de interés – Google Maps

En la Tabla:2, se presentan las imágenes reales de las estaciones base con sus respectivos identificadores que intervienen en el análisis:

Tabla 2: Ubicación y eNB de las estaciones base

Estaciones base	
<p>Estación base 1: 28322 Dirección: Av. Universitaria OE5-302, Quito 170402</p> 	<p>Estación base 5: 28427 Dirección: Manuel Larrea N16-56, Quito 170103</p> 
<p>Estación base 2: 28834 Dirección: 18 de Septiembre y De Soto</p> 	<p>Estación base 6: 28010 Dirección: Manuel Larrea N13-65, Quito 170103</p> 
<p>Estación base 3: 28059 Dirección: 18 de Septiembre y Pérez Guerrero</p>	<p>Estación base 7: 28235 Dirección: Sta. Prisca, Quito 170130</p>

	
<p>Estación base 4: 28875 Dirección: Asunción 272-308, Quito 170103</p>	<p>Estación base 8: 28200 Dirección: Cdla. Mz. 1025 Sl.25 y, 26</p>
	

2.2.1 DETERMINACIÓN DE LA UBICACIÓN GEOGRÁFICA DE LAS ESTACIONES BASE SELECCIONADAS

Para esta parte del trabajo hemos decidido utilizar Google Maps, ya que es una aplicación de mapas utilizada para obtener información detallada sobre una ubicación y su entorno. En este caso la información de interés es la ubicación presentada en términos de latitud y longitud.

Para determinar la latitud y longitud de cada estación base, primero nos dirigimos a la página de CellMapper, y en el Menú seleccionamos la opción “Search”, una vez adentro, en la parte de búsqueda de torre tipeamos el identificador del nodo de la estación a ubicar como se muestra en la Figura: 2. 5.a. Posteriormente en el cuadro de dialogo presentado seleccionamos el resultado Figura: 2.5.b, finalmente el mapa automáticamente se direccionará a la ubicación de la estación base con identificador nodo tipeado Figura: 2.5.c.

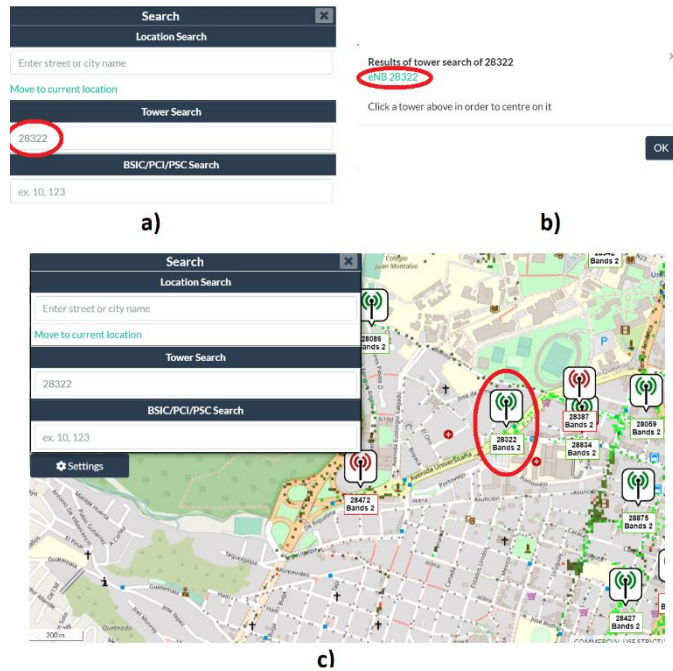


Figura 2.5: Mapa de la página de CellMapper con el proceso de determinar la ubicación visual de las estaciones base

Una vez obtenida la posición visual de la estación base, contrastamos la misma posición en la página web de Google Maps como se exhibe en la Figura: 2.6, posteriormente dando click izquierdo asignamos un punto coherente a la ubicación y automáticamente se nos desplegará en la parte inferior las coordenadas geográficas del punto seleccionado, que en este caso vendría a ser el punto donde se encuentra la estación base con el identificador de nodo tipeado.

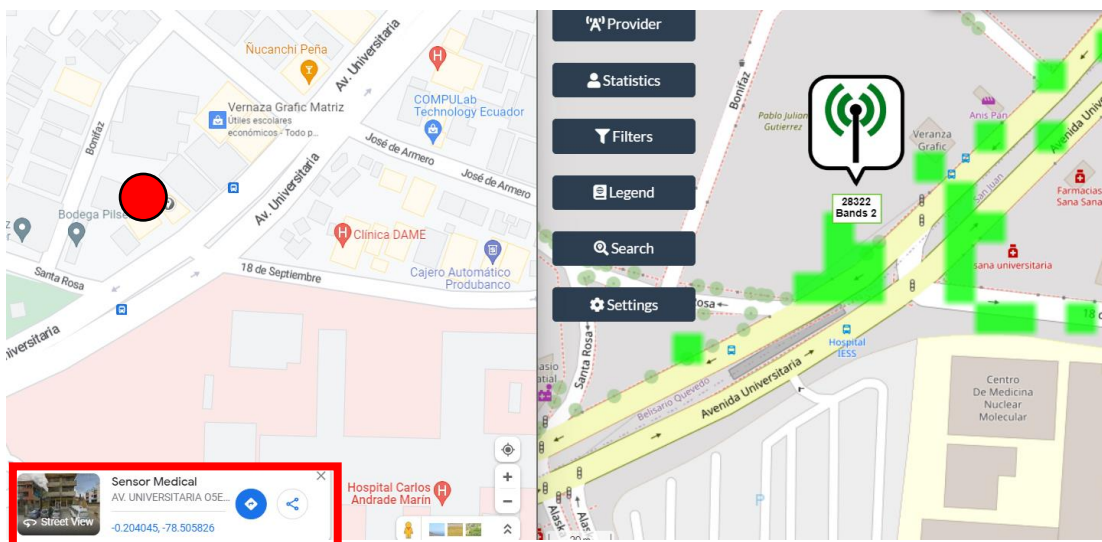


Figura 2.6: Obtención de las coordenadas geográficas para la estación base eNB 28322

Realizamos el mismo procedimiento para todas las estaciones base seleccionadas y como resultado obtendríamos la siguiente tabla:

Tabla 2.1: Coordenadas geográficas de la posición de las estaciones base [26] [27]

Identificador de nodo eNB	Ubicación	
	Latitud	Longitud
28834	-0.204217	-78.503457
28235	-0.214681	-78.504884
28542	-0.197953	-78.502984
28875	-0.206613	-78.501537
28059	-0.203657	-78.501355
28387	-0.203188	-78.503494
28322	-0.204045	-78.505826
28542	-0.197953	-78.502984
28334	-0.209241	-78.500387
28010	-0.212471	-78.503310
28200	-0.213228	-78.507345
28030	-0.210785	-78.494286

Las rutas resaltadas con color rojo también son estaciones base a las que el terminal móvil se conecta y por ende también intervienen en el proceso de handover.

2.3 RUTAS CONSIDERADAS

En el presente trabajo se ha considerado recopilar información en horario rotativo caminando a una velocidad de 4.125 km/h aproximadamente por dos rutas ubicadas en el sector San Juan, este sector es uno de los más arcaicos de Quito. Posee calles empedradas y algunas adoquinadas. En este sector se encuentra la Basílica del Voto Nacional, el antiguo Hospital Militar y el Estadio de San Juan.

2.3.1 PRIMERA RUTA – SECTOR SAN JUAN

La primera ruta planteada tiene una distancia de recorrido de 1.8 km y se representa en la Figura: 2.7, en esta ruta se consideró como punto de partida y llegada las calles Av. América y Rio de Janeiro.

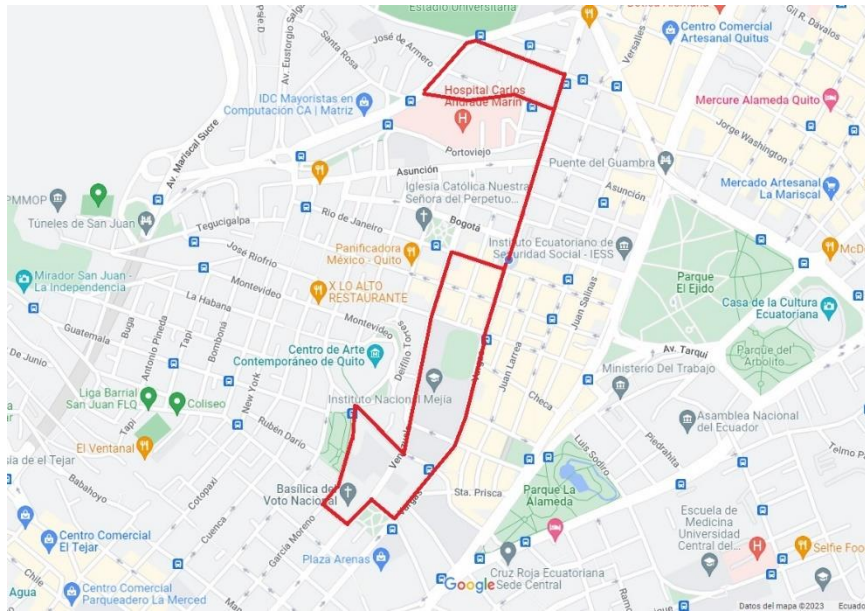


Figura 2.7: Ruta 1 para la recolección de datos

2.3.2 SEGUNDA RUTA – SECTOR SAN JUAN

La segunda ruta planteada tiene una distancia de recorrido de 2.2 km y se representa en la Figura: 2.8, en la cual se consideró de igual manera como punto de partida y llegada las calles Av. América y Río de Janeiro.

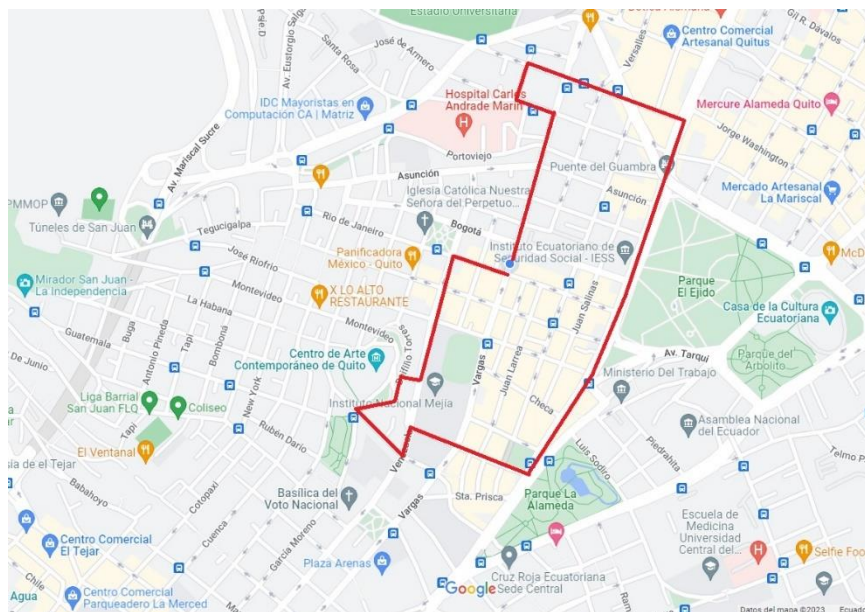


Figura 2.8: Ruta 2 para la recolección de datos

2.4 CRONOGRAMA DE LA RECOLECCIÓN DE DATOS

Una vez planteadas las rutas se ha programado un cronograma, el cual se tiene como objetivo llevar la bitácora de las rutas preestablecidas para la recolección de datos. En la

Tabla: 4, se muestra la fecha inicial de adquisición de información: Lunes 21 de Noviembre 2022, Ruta 1 y fecha de finalización: Jueves 22 de Diciembre del 2022, Ruta 2.

Tabla 2.2: Cronograma de la recolección de datos

Mes	Día	Fecha	Ruta
Noviembre	Lunes	21/11/2022	1
	Martes	22/11/2022	2
	Miércoles	23/11/2022	1
	Jueves	24/11/2022	2
	Viernes	25/11/2022	1
	Sábado	26/11/2022	2
	Domingo	27/11/2022	1
	Lunes	28/11/2022	2
	Martes	29/11/2022	1
	Miércoles	30/11/2022	2
Diciembre	Jueves	1/12/2022	1
	Viernes	2/12/2022	2
	Sábado	3/12/2022	1
	Domingo	4/12/2022	2
	Lunes	5/12/2022	1
	Martes	6/12/2022	2
	Miércoles	7/12/2022	1
	Jueves	8/12/2022	2
	Viernes	9/12/2022	1
	Sábado	10/12/2022	2
	Domingo	11/12/2022	1
	Lunes	12/12/2022	2
	Martes	13/12/2022	1
	Miércoles	14/12/2022	2
	Jueves	15/12/2022	1
	Viernes	16/12/2022	2
	Sábado	17/12/2022	1
	Domingo	18/12/2022	2
	Lunes	19/12/2022	1
	Martes	20/12/2022	2
	Miércoles	21/12/2022	1
	Jueves	22/12/2022	2

El cronograma de las actividades realizadas se muestra en el anexo 1.

2.5 PROCESAMIENTO DE DATOS

Una vez terminada la etapa de recolección de datos en las dos rutas establecidas y definidas con anterioridad, es de carácter importante analizar el conjunto de datos de radio frecuencia recopilados ya que son elementos clave en el proceso de handover. Los datos

obtenidos serán empleados en el algoritmo de Machine Learning con técnica de aprendizaje supervisado y modelo de regresión lineal.

Los datos fueron organizados en carpetas en orden jerárquico, de la siguiente manera: un directorio para ruta, dentro de esta un directorio para fecha y finalmente dentro de esta un directorio para cada aplicación como se exhibe en la Figura: 2.9.

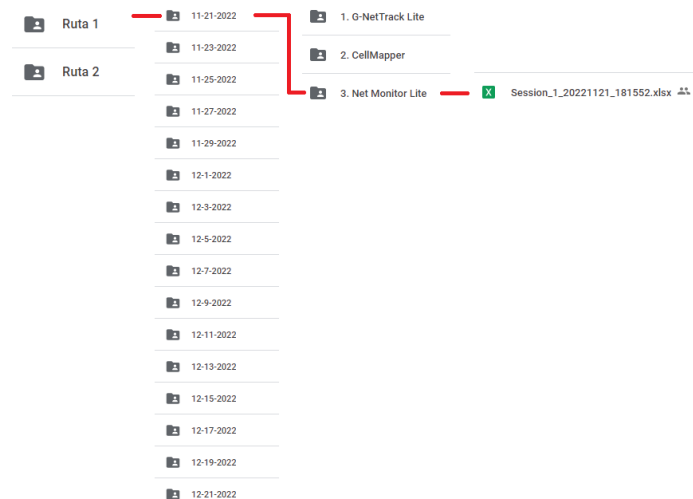


Figura 2.9: Organización de los datos por directorios

Se han empleado 32 sesiones en total para el almacenamiento de los datos en las fechas establecidas. El número total aproximado de muestras recolectadas en la ruta 1 por día es 3637, y en ruta 2 por día es 4000, estos valores son coherentes a la distancia y al tiempo de muestreo empleado. Cabe mencionar que, en los trayectos realizados, se presentaron errores en las mediciones tanto de potencia como en coordenadas de la ubicación causados por distintos factores como: la capacidad de multitarea del celular, interferencias por los cambios climáticos, etc.



Figura 2.10: Sesiones registradas por ruta

Para empezar con el procesamiento, ordenamos todos los datos por rutas en un mismo archivo como se indica en la Figura: 2.11. Estos archivos contienen todas las muestras organizadas por día, un día a continuación de otro en el que se haya realizado la recolección de datos en la misma ruta.

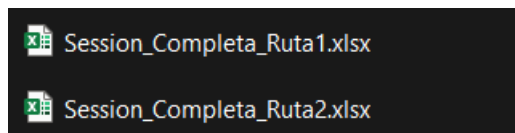


Figura 2.11: Sesiones completas para cada ruta

Una vez organizados los datos, procedemos a realizar el preprocesamiento, el cual consiste en cuatro puntos importantes:

1. Aumentar una columna que representará una nueva variable denominada `hand_over`, la cual indicará mediante valores lógicos o texto, si en cada punto registrado en la medición ocurrió o no el handover. Los valores lógicos empleados son: 0 para Fallido y 1 para Exitoso.
2. Aumentar una columna que representará una nueva variable denominada `Distancia_BS`, la cual indicará la distancia entre las estaciones base servidoras y la estación móvil.
3. Aumentar una columna que representará una nueva variable denominada `azimut`, la cual establecerá la dirección de movimiento del dispositivo móvil.

- De todas las estaciones base consideradas en el campo de investigación, indicar a cuál estación base se acerca el equipo de usuario y de cuáles se aleja

2.5.1 GENERACIÓN DEL ATRIBUTO HAND_OVER

Como sabemos, es un hecho que en las rutas establecidas se va a ejecutar el proceso de handover. Ya que al ir avanzando por la ruta nos vamos a alejar de la estación base a la que estamos conectados, esto quiere decir que el recorrido entre la estación base y la estación móvil va a aumentar y por ende la potencia va a disminuir. Para que el dispositivo mantenga la conexión a la red, debe conectarse a otra estación base con mejores características de mayor potencia y menor distancia.

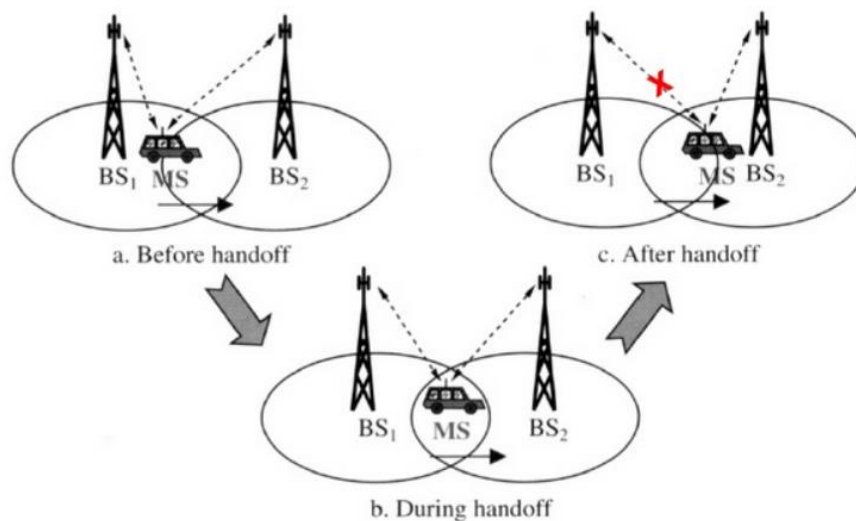


Figura 2.12: Proceso del handover [28]

Las condiciones adicionales para esta parte son:

Para el desarrollo de esta parte, debemos tener en cuenta la variable `node_id` de nuestra base de datos, ya que para adicionar el nuevo atributo (`hand_over`), inferiremos que un cambio de identificador de nodo en los datos se interpretará como un cambio de celda celular, es decir como handover.

Para tipificar esta variable en Excel, lo haríamos a través de la sentencia condicional "SI", siguiente manera:

Si el valor de la variable `node_id` es igual al de la siguiente celda, entonces se asignará un valor lógico "0", el cual indica que no se realizó un cambio de celda celular y por ende no ocurrió handover. Y si no se cumple la condición, se asignará el valor lógico "1", el cual

representará la existencia de un cambio de celda celular, y por ende la existencia de handover, como se exhibe en la Figura: 2.13.

=SI(J9=J10;0;1)

H	I	J	K	L	M	N	O	P	Q	R
trongest	mcc	node_id	cid	rssl	rsrq	rssnr	accuracy	lat	long	hand_over
-99	740	28235	191	-100	-15	-2	4	-0.208378333	-78.50326667	0
-99	740	28235	191	-104	-17	-2	4	-0.208366667	-78.50326333	0
-99	740	28235	191	-104	-17	0	4	-0.20835	-78.503265	0
-99	740	28235	191	-104	-17	0	4	-0.208333333	-78.50326167	0
-100	740	28235	191	-100	-20	0	4	-0.208318333	-78.50325833	0
-100	740	28235	191	-100	-20	0	3	-0.208301667	-78.50325667	0
-100	740	28235	191	-100	-20	0	3	-0.20829	-78.503255	0
-99	740	28235	191	-103	-20	-20	3	-0.20828	-78.50325167	0
-99	740	28235	191	-103	-20	-10	3	-0.208271667	-78.50324833	0
-99	740	28235	191	-103	-20	-10	3	-0.208258333	-78.50324833	1
-97	740	28334	190	-98	-14	-10	3	-0.208241667	-78.50325167	0
-97	740	28334	190	-98	-14	-10	3	-0.208216667	-78.50325833	0
-97	740	28334	190	-98	-14	-10	3	-0.208195	-78.50326	0
-96	740	28334	190	-91	-13	25	3	-0.208176667	-78.50326	0

Figura 2.13: Atributo hand_over creado en Excel

Una vez definida la variable hand_over con sus respectivos resultados, procedemos a establecer una zona general de handover, la cual representaremos a partir del cambio de celda celular, con cinco valores "1" hacia arriba y cinco valores "1" hacia abajo. Para su automatización en Excel empleamos recursividad en cada columna. Y finalmente para proporcionarle color a las celdas, creamos una regla que indique colorear todos los "1s", como se exhibe en la Figura 2.14.

R	S	T	U	V	W	X
hand_over	zone_ho					
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	1	1	1
0	0	0	1	1	1	1
0	0	1	1	1	1	1
1	1	1	1	1	1	1
0	0	1	1	1	1	1
0	0	0	1	1	1	1
0	0	0	0	1	1	1
0	0	0	0	0	1	1
0	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Figura 2.14: Zona de handover

Debido a que necesitamos automatizar las coordenadas de latitud y longitud de las estaciones base en toda nuestra base de datos procedemos a realizar lo siguiente:

Ubicamos las coordenadas de las estaciones base de la Tabla: 2.1, en la base de datos, como se exhibe en la Figura: 2.15.

	BG	BH	BI
1			
2			
3	28834	-0.20422	-78.5035
4	28235	-0.21468	-78.5049
5	28542	-0.19795	-78.503
6	28875	-0.20661	-78.5015
7	28059	-0.20366	-78.5014
8	28387	-0.20319	-78.5035
9	28322	-0.20405	-78.5058
10	28542	-0.19795	-78.503
11	28334	-0.20924	-78.5004
12	28010	-0.21247	-78.5033
13	28200	-0.21323	-78.5073
14	28030	-0.21079	-78.4943

Figura 2.15: Coordenadas de las estaciones base

Luego, creamos las respectivas condicionales que dependen del identificador de nodo, para asignar la latitud y longitud a las respectivas estaciones base, como se indica en la Figura: 2.16.

J	K	L	M	N	O	P	Q
node_id	cid	rss_i	rsrq	rssnr	accuracy	lat	long
28334	190	-105	-14	2147483647	5	-0.207256667	-78.50335
28334	190	-105	-14	2147483647	5	-0.207266667	-78.50333167
28542	191	-106	-19	2147483647	5	-0.207246667	-78.50331
28542	191	-106	-19	2147483647	5	-0.207245	-78.50329667
28542	191	-106	-19	2147483647	5	-0.20725	-78.50329167

Figura 2.16: Asignación de las coordenadas de latitud y longitud a las estaciones base

Posteriormente para determinar si el handover fue exitoso o existieron fallas en dicho proceso, consideramos las siguientes condicionales:

Si el valor de potencia de las variables rssi_strongest, rssi y rsrq son diferentes a "2147483647", las coordenadas geográficas son diferentes de "-1" y además esté dentro de la zona de handover, entonces se podrá concluir que el handover es Exitoso, caso contrario existen fallas en el handover.

=SI(Y(X86=1;H86<>2147483647;L86<>2147483647;M86<>2147483647;P86<>1);"Exitoso";"Fallido")																									
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y					
tiempo	Distancia	Velocidad	ssi_strongest	mcc	node_id	cid	ssi	rsrq	rssnr	accuracy	lat	long	hand_over	U	U	U	U	U	U	zone_ho	HO_EoF				
0.00028	0.66824	1.9E-07	-101	740	28235	191	-99	-15	-2	4	-0.20840167	-78.5032767	0	0	0	0	0	0	0	0	Fallido				
0.00028	0.76386	2.1E-07	-99	740	28235	191	-100	-15	-2	4	-0.20839833	-78.5032717	0	0	0	0	0	0	0	0	Fallido				
0.00028	1.65733	4.6E-07	-99	740	28235	191	-100	-15	-2	4	-0.20839167	-78.5032733	0	0	0	0	0	0	0	0	Fallido				
0.00028	1.34932	3.7E-07	-99	740	28235	191	-100	-15	-2	4	-0.20837833	-78.5032667	0	0	0	0	0	0	0	0	Fallido				
0.00028	1.86257	5.2E-07	-99	740	28235	191	-104	-17	-2	4	-0.20836667	-78.5032633	0	0	0	0	0	0	0	0	Fallido				
0.00028	1.88991	5.2E-07	-99	740	28235	191	-104	-17	0	4	-0.20835	-78.503265	0	0	0	0	0	0	0	0	Fallido				
0.00028	1.70877	4.7E-07	-99	740	28235	191	-104	-17	0	4	-0.20833333	-78.5032617	0	0	0	0	0	0	0	0	Fallido				
0.00028	1.86234	5.2E-07	-100	740	28235	191	-100	-20	0	4	-0.20831833	-78.5032583	0	0	0	0	0	0	0	0	Exitoso				
0.00028	1.31053	3.6E-07	-100	740	28235	191	-100	-20	0	3	-0.20830167	-78.5032567	0	0	0	0	0	0	1	1	Exitoso				
0.00028	1.17198	3.3E-07	-100	740	28235	191	-100	-20	0	3	-0.20829	-78.503255	0	0	0	0	1	1	1	1	Exitoso				
0.00028	0.99825	2.8E-07	-99	740	28235	191	-103	-20	-20	3	-0.20828	-78.5032517	0	0	0	1	1	1	1	1	Exitoso				
0.00028	1.48267	4.1E-07	-99	740	28235	191	-103	-20	-10	3	-0.20827167	-78.5032483	0	0	1	1	1	1	1	1	Exitoso				
0.00028	1.89002	5.3E-07	-99	740	28235	191	-103	-20	-10	3	-0.20825833	-78.5032483	1	1	1	1	1	1	1	1	Exitoso				
0.00028	2.87682	8E-07	-97	740	28334	190	-98	-14	-10	3	-0.20824167	-78.5032517	0	0	1	1	1	1	1	1	Exitoso				
0.00028	2.41641	6.7E-07	-97	740	28334	190	-98	-14	-10	3	-0.20821667	-78.5032583	0	0	0	1	1	1	1	1	Exitoso				
0.00028	2.03854	5.7E-07	-97	740	28334	190	-98	-14	-10	3	-0.208195	-78.50326	0	0	0	0	1	1	1	1	Exitoso				
0.00028	2.072	5.8E-07	-96	740	28334	190	-91	-13	25	3	-0.20817667	-78.50326	0	0	0	0	0	1	1	1	Exitoso				
0.00028	2.9886	8.3E-07	-96	740	28334	190	-91	-13	25	3	-0.20815833	-78.5032633	0	0	0	0	0	0	0	1	Exitoso				
0.00028	3.01661	8.4E-07	-96	740	28334	190	-91	-13	25	3	-0.208135	-78.5032767	0	0	0	0	0	0	0	0	Fallido				
0.00028	2.35879	6.6E-07	-96	740	28334	190	-87	-13	25	3	-0.208115	-78.503295	0	0	0	0	0	0	0	0	Fallido				
0.00028	0.41408	1.2E-07	-96	740	28334	190	-87	-13	70	3	-0.2081	-78.50331	0	0	0	0	0	0	0	0	Fallido				
0.00028	1.31081	3.6E-07	-96	740	28334	190	-81	-12	70	3	-0.20809833	-78.5033067	0	0	0	0	0	0	0	0	Fallido				
0.00028	0.58606	1.6E-07	-96	740	28334	190	-81	-12	70	3	-0.2081	-78.503295	0	0	0	0	0	0	0	0	Fallido				

Figura 2.17: Condiciones para la determinación de handover en Excel

2.5.2 GENERACIÓN DEL ATRIBUTO DISTANCIA_BS

Para el cálculo de la distancia emplearemos la ecuación Haversine, ya que nos permite calcular la distancia existente entre dos puntos geográficos, conociendo los parámetros de longitud y su latitud de dichos puntos.

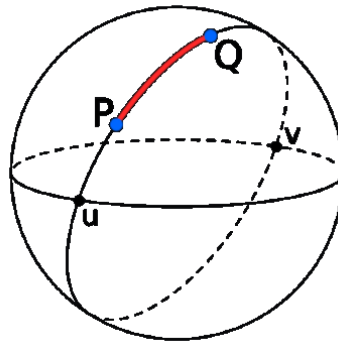


Figura 2.18: Puntos geográficos P (λ_1, ϕ_1) y Q (λ_2, ϕ_2)

Ecuación

$$distancia = 2 * r * \sin^{-1} \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) * \cos(\phi_2) * \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (2.1)$$

Donde:

r: radio de la tierra

ϕ_1 : Latitud inicial.

λ_1 : Longitud inicial.

ϕ_2 : Latitud final.

Longitud2 (λ_2): Longitud final.

Para su implementación, esta ecuación requiere las siguientes consideraciones [29].

- Es necesario expresar las Latitudes y Longitudes en grados decimales, y no en grados, minutos o segundos
- El radio terrestre en kilómetros varía dependiendo de la ubicación:
 - En los polos: 6,356.8 km
 - En la zona ecuatorial: 6,378.10 km
 - El radio promedio es de 6,371.0 km
- La distancia como resultado se expresa en metros [29].

Para su implementación en Excel, condicionamos ciertos parámetros como: si la latitud presenta un valor de "-1", asignamos un "0" caso contrario se calcula la distancia geográfica considerando como punto inicial la ubicación de la estación móvil y como punto final la ubicación de la estación base a la que se conecta. En este caso al hablar de distancia, no importa la precedencia de punto inicial y final.

=SI(O(P114=-1;Z114=FALSE);0;2*ASENO(RAIZ(SENO((P114-Z114)*PI()/180*0.5)^2+COS(P114*PI()/180)*COS(Z114*PI()/180)*SENO((Q114-AA114)*PI()/180*0.5)^2))*1000*6371)

accuracy	lat	long	and_over	R	S	T	U	V	W	X	Y	Z	AA	AB
										zone_ho	HO_EoF	Lat_Est_base	Long_Est_base	Distancia
	-0.208098553	-78.50305067	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	348.8298285
	-0.2081	-78.503295	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	347.3526054
	-0.20809833	-78.50329	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	346.902957
	-0.2080833	-78.503285	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	346.7951006
	-0.2080833	-78.503267	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	346.140981
	-0.208075	-78.503267	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	345.455729
	-0.2080633	-78.50326	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	345.258351
	-0.20805167	-78.503253	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	345.067289
	-0.2080433	-78.5032467	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	344.740325
	-0.20803167	-78.5032417	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	344.7315153
	-0.20802167	-78.503285	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	349.602678
	-0.20800833	-78.5033083	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	352.5690025
	-0.20798833	-78.5033067	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	353.2698784
	-0.20797167	-78.5033017	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	353.4948604
	-0.207955	-78.5032933	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	353.3906694
	-0.20793667	-78.5032833	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	353.2062845
	-0.20792	-78.5032833	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	353.9713231
	-0.20790667	-78.5032733	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	353.5794983
	-0.20788833	-78.5032933	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	356.4547609
	-0.20787833	-78.5032933	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	356.925583
	-0.20787167	-78.5032933	0	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	357.2406778

Radio promedio

Figura 2.19: Cálculo de la distancia entre dos puntos geográficos en Excel

2.5.3 GENERACIÓN DEL ATRIBUTO AZIMUT

El azimut se define como el ángulo horizontal que se mide en sentido horario desde el norte o desde otro meridiano de referencia. Aunque generalmente se mide desde el norte, en algunas situaciones se puede medir desde el sur. Para calcular el azimut, es necesario conocer las coordenadas geográficas tanto de las estaciones base como de las estaciones móviles [30].

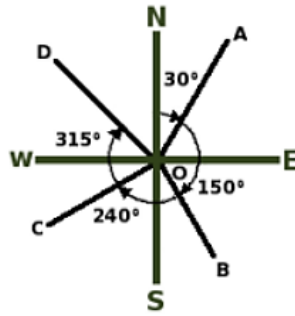


Figura 2.20: Acimuts y rumbos

$$azimuth = \tan^{-1} \left(\frac{\lambda_2 - \lambda_1}{\phi_2 - \phi_1} \right) \quad (2.2)$$

Donde:

ϕ_1 : Latitud inicial.

λ_1 : Longitud inicial.

ϕ_2 : Latitud final.

λ_2 : Longitud final.

Una vez calculado el azimut se procede a ubicarlo en el cuadrante que corresponde, para ello definimos ciertas condicionales que nos ayudaran en el cálculo Figura: 2.21.

Tabla 2.3: Condicionales para el ángulo azimut [30]

Si $(\lambda_2 - \lambda_1) > 0$ y $(\phi_2 - \phi_1) > 0$	Primer cuadrante – ángulo directo
Si $(\lambda_2 - \lambda_1) > 0$ y $(\phi_2 - \phi_1) < 0$	Segundo cuadrante – se suma 200° al ángulo adquirido
Si $(\lambda_2 - \lambda_1) < 0$ y $(\phi_2 - \phi_1) < 0$	Tercer cuadrante – se suma 200° al ángulo adquirido
Si $(\lambda_2 - \lambda_1) < 0$ y $(\phi_2 - \phi_1) > 0$	Cuarto cuadrante – se suma 400° al ángulo adquirido

En la implementación en Excel primero filtramos las mediciones de coordenadas obtenidas de valor “-1” en las columnas de latitud y longitud, posteriormente procedemos a tipificar las condicionales por cuadrantes mencionadas.

=SI(P11=-1;0;SI(Y(AD11>0;AC11>0);ATAN(AD11/AC11);SI(Y(AD11>0;AC11<0);ATAN(AD11/AC11)+200;SI(Y(AD11<0;AC11<0);ATAN(AD11/AC11)+200;SI(Y(AD11<0;AC11>0);ATAN(AD11/AC11)+400))))))																
P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
at	long	hand_over	U	U	U	U	U	U	zone_ho	HO EaF	Lat	Est_base	Distancia	diff lat	diff long	azimuth
-0.208418333	-78.503325	0	0	0	0	0	0	0	Fallido	-0.214681	-78.504884	718.5288738	-0.006266	-0.001579	200.246855	467.397432
-0.208413333	-78.50329	0	0	0	0	0	0	0	Fallido	-0.214681	-78.504884	719.1178965	-0.006267	-0.001594	200.249041	466.980292
-0.208401667	-78.50327667	0	0	0	0	0	0	0	Fallido	-0.214681	-78.504884	720.7412701	-0.0062793	-0.0016073	200.250591	465.74558
-0.208398333	-78.50327167	0	0	0	0	0	0	0	Fallido	-0.214681	-78.504884	721.23842	-0.0062827	-0.0016123	200.25121	465.399487
-0.208391667	-78.50327333	0	0	0	0	0	0	0	Fallido	-0.214681	-78.504884	721.9105891	-0.0062893	-0.0016107	200.250707	464.65084
-0.208378333	-78.50326667	0	0	0	0	0	0	0	Fallido	-0.214681	-78.504884	723.5307169	-0.0063027	-0.0016173	200.25119	463.202849
-0.208366667	-78.50326333	0	0	0	0	0	0	0	Fallido	-0.214681	-78.504884	724.8795194	-0.0063143	-0.0016207	200.251242	461.924173
-0.20835	-78.503265	0	0	0	0	0	0	0	Fallido	-0.214681	-78.504884	726.6287388	-0.006331	-0.001619	200.25036	460.064256
-0.208333333	-78.50326167	0	0	0	0	0	0	0	Fallido	-0.214681	-78.504884	728.5159894	-0.0063477	-0.0016223	200.250222	458.230376
-0.208318333	-78.50325833	0	0	0	0	0	0	0	Exitoso	-0.214681	-78.504884	730.2239348	-0.0063627	-0.0016257	200.25015	456.582152
-0.208301667	-78.50325667	0	0	0	0	0	1	1	Exitoso	-0.214681	-78.504884	732.0651773	-0.0063793	-0.0016273	200.249767	454.740161
-0.20829	-78.503255	0	0	0	0	1	1	1	Exitoso	-0.214681	-78.504884	733.3681463	-0.006391	-0.001629	200.249575	453.453572
-0.20828	-78.50325167	0	0	0	1	1	1	1	Exitoso	-0.214681	-78.504884	734.5371047	-0.006401	-0.0016323	200.24969	452.361528
-0.208271667	-78.50324833	0	0	1	1	1	1	1	Exitoso	-0.214681	-78.504884	735.5267405	-0.0064093	-0.0016357	200.249868	451.45506
-0.208258333	-78.50324833	1	1	1	1	1	1	1	Exitoso	-0.214681	-78.504884	736.9634612	-0.0064227	-0.0016357	200.24937	449.974352
-0.208241667	-78.50325167	0	0	1	1	1	1	1	Exitoso	-0.209241	-78.500387	337.3605647	-0.0009993	0.00286467	198.764851	448.104577
-0.208216667	-78.50325833	0	0	0	1	1	1	1	Exitoso	-0.209241	-78.500387	338.983796	-0.0010243	0.00287133	198.771875	445.290984
-0.208195	-78.50326	0	0	0	0	1	1	1	Exitoso	-0.209241	-78.500387	339.9753827	-0.001046	0.002873	198.778366	442.875483
-0.208176667	-78.50326	0	0	0	0	0	1	1	Exitoso	-0.209241	-78.500387	340.6781774	-0.0010643	0.0028733	198.783989	440.839453
-0.208158333	-78.50326333	0	0	0	0	0	0	1	Exitoso	-0.209241	-78.500387	341.7382525	-0.0010827	0.00287633	198.789206	438.785006
-0.208135	-78.50327667	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	344.045763	-0.001106	0.00288967	198.794745	436.122924
-0.208115	-78.503295	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	346.7468586	-0.001126	0.002908	198.798634	433.811978
-0.2081	-78.50331	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	348.9058077	-0.001141	0.002923	198.801366	432.079187
-0.208083333	-78.50330667	0	0	0	0	0	0	0	Fallido	-0.209241	-78.500387	348.6284183	-0.0011437	0.00291967	198.802327	431.90813

Figura 2.21: Calculo del Azimut en Excel

2.5.4 GENERACIÓN DEL ATRIBUTO SE ACERCA O SE ALEJA

Para esta parte procedemos a utilizar los datos de la Tabla: 2.1, y utilizar la fórmula de distancia entre dos puntos geográficos, para calcular todas las distancias desde el terminal móvil a todas las estaciones base, como se exhibe en la Figura: 2.22.

=SI(Y(J2<>1;P2<>-1);2*ASENO(RAI(Z(SENO((P2-SBHS3)*PI()/180*0.5)^2+COS(P2*PI()/180)*COS(SBHS3*PI()/180)*SENO((Q2-SBIS3)*PI()/180*0.5)^2))*1000*6371,0)												
AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	
28834	28235	28542	28875	28059	28387	28322	28542	28334	28010	28200	28030	
469.762526	711.912	1167.429	294.1344	580.7209	584.2032	555.7958	1167.429	351.8147	448.3053	686.2882	1051.591	
469.020846	712.592	1166.696	293.7568	580.1163	583.4603	555.0554	1166.696	352.1823	449.0526	686.7461	1051.955	
468.649963	712.912	1166.334	293.6359	579.8516	583.0882	554.6411	1166.334	352.4563	449.4295	686.9167	1052.227	
468.095515	713.62	1165.746	292.7153	579.0445	582.5387	554.506	1165.746	351.8819	449.9598	687.8158	1051.647	
468.09743	713.745	1165.722	292.3126	578.8216	582.5441	554.7716	1165.722	351.3447	449.9416	688.1665	1051.109	
468.105762	714.08	1165.658	291.2414	578.2295	582.5613	555.482	1165.658	349.9128	449.8964	689.1027	1049.673	
468.31456	714.45	1165.744	289.6369	577.444	582.7825	556.8055	1165.744	347.5376	449.6482	690.4855	1047.294	
467.973598	715.283	1165.294	287.9193	576.3015	582.4501	557.4699	1165.294	345.667	449.9754	692.0665	1045.414	
468.073955	716.375	1165.125	284.6278	574.501	582.5635	559.739	1165.125	341.1982	449.9115	695.0225	1040.929	
467.397432	717.629	1164.31	282.5342	572.963	581.89	560.1947	1164.31	339.2541	450.6391	697.0164	1038.965	
467.102183	718.529	1163.869	280.7091	571.7735	581.5957	560.981	1163.869	337.2138	451.007	698.7285	1036.908	
466.980292	719.118	1163.634	279.4107	570.9714	581.4731	561.656	1163.634	335.6601	451.1975	699.9438	1035.342	
465.74558	720.741	1162.294	277.4473	569.212	580.2368	561.2816	1162.294	334.5944	452.5044	701.8908	1034.24	
465.399487	721.238	1161.908	276.794	568.6598	579.8898	561.2428	1161.908	334.1643	452.88	702.5326	1033.798	
464.65084	721.911	1161.173	276.3917	568.0419	579.1414	560.5102	1161.173	334.55	453.6195	702.9805	1034.167	
463.202849	723.531	1159.67	274.8139	566.3897	577.692	559.6084	1159.67	334.2621	455.1093	704.5918	1033.834	
461.924173	724.88	1158.363	273.6282	565.0479	576.4124	558.6812	1158.363	334.2813	456.4105	705.8239	1033.812	
460.064256	726.629	1156.516	272.4415	563.4007	574.5527	556.9929	1156.516	335.0024	458.2616	707.1251	1034.475	
458.230376	728.516	1154.653	270.8685	561.5447	572.7179	555.5909	1154.653	335.202	460.119	708.7848	1034.604	

Figura 2.22: Distancia entre la estación móvil y todas las estaciones base

Una vez obtenidas las distancias, procedemos analizar si el dispositivo móvil se acerca o se aleja de las estaciones base. Para ello discriminamos que conforme avanza el tiempo, al disminuir la distancia existente entre la estación móvil y la estación base, el terminal móvil se "Se Acerca", y si aumenta la distancia entre estas estaciones, el terminal móvil se "Se Aleja", Figura: 2.23.

=SI(AF68<>0);SI(AF69<AF68;"Se Acerca";"Se Aleja");0)

AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD
28834	28235	28542	28875	28059	28387	28322	28542	28334	28010	28200	28030		28834	28235	28542	28875	28059	28387	28322	28542	28334	28010	28200	28030
469.02085	712.592	1166.7	293.757	580.116	583.46	555.055	1166.7	352.182	449.053	686.746	1051.95		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja	Se Aleja
468.64996	712.912	1166.33	293.636	579.852	583.088	554.641	1166.33	352.456	449.429	686.917	1052.23		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja	Se Aleja
468.09552	713.62	1165.75	292.715	579.044	582.539	554.506	1165.75	351.882	449.96	687.816	1051.65		Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
468.09743	713.745	1165.72	292.313	578.822	582.544	554.772	1165.72	351.345	449.942	688.167	1051.11		Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
468.10576	714.08	1165.66	291.241	578.229	582.561	555.482	1165.66	349.913	449.896	689.103	1049.67		Se Aleja	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
468.31456	714.45	1165.74	289.637	577.444	582.782	556.805	1165.74	347.538	449.648	690.486	1047.29		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Aleja	Se Aleja	Se Aleja
467.9736	715.283	1165.29	287.919	576.301	582.45	557.47	1165.29	345.667	449.975	692.066	1045.41		Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
468.07396	716.375	1165.12	284.628	574.501	582.363	559.739	1165.12	341.198	449.912	695.022	1040.93		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
467.39743	717.629	1164.31	283.534	572.963	581.89	560.195	1164.31	339.254	450.839	697.016	1038.96		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
467.10218	718.529	1163.87	280.709	571.774	581.596	560.981	1163.87	337.214	451.007	698.728	1036.91		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
466.98029	719.118	1163.63	279.411	570.971	581.473	561.656	1163.63	335.66	451.197	699.944	1035.34		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
465.74558	720.741	1162.29	277.447	569.212	580.237	561.282	1162.29	334.594	452.504	701.891	1034.24		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
465.39949	721.238	1161.91	276.794	568.66	579.89	561.243	1161.91	334.164	452.88	702.533	1033.8		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
464.65084	721.911	1161.17	276.392	568.042	579.141	560.51	1161.17	334.55	453.619	702.981	1034.17		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
463.20285	723.531	1159.67	274.814	566.39	577.692	559.608	1159.67	334.262	455.109	704.592	1033.83		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
461.92417	724.98	1158.36	273.628	565.048	576.412	558.681	1158.36	334.281	456.411	705.824	1033.81		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
460.06426	726.629	1156.52	272.441	563.401	574.555	556.993	1156.52	335.002	458.262	707.125	1034.47		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
458.23038	728.516	1154.65	270.868	561.545	572.718	555.591	1154.65	335.202	460.119	708.785	1034.6		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
456.58215	730.224	1152.98	269.429	559.86	571.069	554.351	1152.98	335.354	461.791	710.304	1034.69		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
454.74016	732.065	1151.12	267.996	558.075	569.226	552.859	1151.12	335.751	463.647	711.847	1035		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
453.45357	733.368	1149.82	266.956	556.804	567.939	551.845	1149.82	335.98	464.946	712.964	1035.17		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
452.36153	734.537	1148.7	265.915	555.635	566.845	551.086	1148.7	335.981	466.063	714.057	1035.11		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
451.45506	735.527	1147.76	265.003	554.837	565.938	550.488	1147.76	335.925	466.995	715.098	1035		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
449.97435	736.963	1146.28	263.973	553.265	564.457	549.223	1146.28	336.403	468.477	716.151	1035.41		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
448.10458	738.668	1144.43	262.962	551.694	562.588	547.448	1144.43	337.361	470.325	717.346	1036.27		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja
445.29098	741.182	1141.68	261.596	549.411	559.776	544.692	1141.68	338.984	473.095	719.029	1037.74		Se Acerca	Se Aleja	Se Acerca	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Acerca	Se Acerca	Se Aleja	Se Aleja	Se Aleja

Figura 2.23: Estaciones base que se acercan y alejan del equipo de usuario

2.6 MACHINE LEARNING

El aprendizaje automático se define como una rama de la inteligencia artificial que otorga a los dispositivos con capacidad computacional la habilidad de aprender a identificar patrones en el conjunto de datos de manera autónoma, es decir, sin la necesidad de la intervención humana para su programación. Esos patrones se pueden usar luego para tomar decisiones futuras y predecir comportamientos [31] [32].

Se pueden identificar tres categorías de Machine Learning:

- Aprendizaje supervisado
- Aprendizaje no supervisado
- Aprendizaje de refuerzo

2.6.1 APRENDIZAJE SUPERVISADO

Es una parte del Machine Learning en el que se enseñan o entrenan algoritmos a partir de datos que ya vienen asociados o etiquetados a una respuesta correcta. Cuando el conjunto de datos es considerablemente extenso el algoritmo adquiere de mejor manera las características para aprender. Luego de completar la fase de entrenamiento, se suministran al algoritmo nuevos datos con el fin de que pueda realizar predicciones basadas en la experiencia adquirida durante la fase previa de entrenamiento [31].

Existen dos tipos de aprendizaje supervisado:

- **Clasificación:** Se trata de un tipo de aprendizaje automático que se emplea para pronosticar una variable de clasificación o categórica a partir de un conjunto de

datos de entrada. El algoritmo es entrenado con un conjunto de datos categorizados, es decir, datos en los que se sabe cuál es la categoría correcta para cada entrada. Con esta información, el algoritmo puede aprender a clasificar nuevas entradas dentro de las categorías ya definidas previamente.

- **Regresión:** Este tipo de aprendizaje automático se utiliza para predecir una variable continua a partir de un conjunto de datos de entrada como referencia. El algoritmo es entrenado con un conjunto de datos etiquetados, es decir, datos en los que se sabe el valor correspondiente para cada entrada. De esta manera, el algoritmo puede aprender a predecir el valor de una nueva entrada [33].

2.6.1.1 Algoritmos de aprendizaje supervisado

- **Regresión logística:** Se trata de un modelo estadístico utilizado para evaluar la probabilidad de que suceda un evento determinado. Emplea cálculos matemáticos para identificar la relación entre distintas características y, posteriormente, evalúa la probabilidad de un resultado específico. [34] [35].
- **Support vector machine (SVM):** Se trata de un conjunto de algoritmos utilizados con el fin de encontrar un hiperplano en un espacio con n-dimensiones que clasifique claramente los puntos de datos. SVM es muy eficaz para trabajar con datos que tienen muchas características, incluso en espacios de alta dimensión [36].
- **Redes neuronales:** Es un modelo de aprendizaje automático similar a la estructura y funcionamiento del cerebro humano, con una serie de capas de neuronas artificiales interconectadas, que procesan información de entrada y producen una salida. Estas redes pueden aprender de los datos; es decir, se pueden entrenar a partir de los datos, de tal manera que reconozca patrones, clasifique datos y prediga eventos futuros [37] [38].
- **Clasificador Naive Bayes:** Se trata de un algoritmo de aprendizaje automático de clasificación supervisada, que es altamente rápido y eficiente. Este algoritmo hace uso del teorema de Bayes para realizar predicciones probabilísticas. Emplea una función de distribución para estimar la posibilidad de que un objeto de entrada pertenezca a una determinada clase y posteriormente clasifica el objeto de entrada en la clase con la probabilidad más alta. [39].
- **Árbol de decisión:** Se trata de una estructura con forma de árbol con gran similitud a un diagrama de flujo que sirve para clasificar o predecir una variable de salida a

partir de varias variables de entrada. Cada nodo del árbol representa un atributo o variable de entrada, la rama simboliza una salida o una regla de decisión y finalmente las hojas representan la clase a la que se debe asignar el objeto de entrada [40] [41].

- **Análisis de discriminante:** Es una técnica predictiva donde se conocen previamente las clases antes de realizar la clasificación. El propósito de esta técnica es identificar una combinación de coeficientes lineales de las respectivas variables predictoras y maximizar la separación entre las diferentes clases de objetos. [42] [43].
- **K vecinos más cercanos:** Se trata de un algoritmo aprendizaje supervisado empleado para la clasificación y regresión de objetos de entrada. En este algoritmo se realiza el cálculo de la distancia entre el objeto de entrada y todos los objetos del conjunto de entrenamiento, y luego seleccionando los k objetos más cercanos [44] [45].
- **Clasificación de ensembles:** Se trata de una técnica donde se usa la combinación de múltiples modelos de clasificación para mejorar la precisión y el rendimiento del modelo. Los ensembles se pueden construir a partir de diferentes modelos de clasificación, como árboles de decisión, SVM, redes neuronales y k vecinos más cercanos [46].

2.6.2 APRENDIZAJE NO SUPERVISADO

Se trata de una técnica de aprendizaje automático, en la que el algoritmo se entrena mediante un conjunto de datos que no contienen categorías o etiqueta previamente definidas, es decir el algoritmo no tiene información sobre que representan los datos. El algoritmo debe ser capaz de identificar patrones y relaciones en el conjunto de datos sin que se le proporcione información previa. [31] [47].

2.6.2.1 Algoritmos de aprendizaje no supervisado

- **Algoritmo de clustering:** Se trata de un método de aprendizaje no supervisado que se basa en la agrupación del conjunto de datos o "clusters" basados en la similitud entre ellos. El objetivo es encontrar patrones y estructuras en los datos sin tener una etiqueta de clase o categoría [48].
- **Análisis de componentes principales:** Se trata de una técnica estadística de aprendizaje no supervisado que transforma un conjunto de variables originales en un conjunto reducido de variables no correlacionadas, denominadas componentes

principales. El objetivo de este algoritmo es reducir la complejidad de un conjunto de datos mediante la transformación de las variables originales en componentes principales no correlacionados [49] [51].

- **Descomposición de valores singulares:** Esta técnica estadística multivariante se utiliza para reducir los conjuntos de datos. También plantea soluciones significativas para este menor número de valores. [50].

2.5.3 Aprendizaje Por Refuerzo

En esta técnica el algoritmo aprende observando todo a su alrededor e interactuando con su entorno. En este enfoque, la retroalimentación del mundo exterior es su entrada como respuesta a sus acciones. Es así como el sistema adquiere conocimiento a base de prueba y error [31].

2.7 TÉCNICA DE CLASIFICACIÓN DE MACHINE LEARNING SELECCIONADA

En este trabajo se utilizará la técnica de aprendizaje supervisado, ya que este tipo de técnica se aplicará para la obtención de un modelo predictivo en base a las medidas de los parámetros de radiofrecuencia en las redes celulares y el planteamiento posterior de un algoritmo básico para la ejecución del proceso de handover.

Se utiliza el modelo “Regresión Lineal” debido a que permite predecir una variable continua, definida como salida, partiendo de un conjunto de variables o atributos de entrada. El modelo escogido resulto ideal ya que la variable definida como salida del modelo es de tipo numérico.

2.8 DESCRIPCIÓN DE LA ETAPA PARA EL ANÁLISIS DE DATOS

Con el propósito del desarrollo de esta etapa, primero importamos los datos obtenidos de las herramientas de recolección, luego los depuramos eliminando las columnas de parámetros que no son de interés y los errores en las mediciones, posteriormente desarrollamos y aplicamos el modelo predictivo de aprendizaje supervisado “Regresión Lineal” a los datos seleccionando variables de entrada y salida, y finalmente visualizamos los resultados obtenidos para posteriormente establecer conclusiones.

Para el desarrollo de esta etapa, primero importamos los datos obtenidos de las herramientas de recolección, luego los depuramos eliminando las columnas de parámetros que no son de interés y los errores en las mediciones, posteriormente desarrollamos y aplicamos el modelo predictivo de aprendizaje supervisado “Regresión Lineal” a los datos seleccionando variables de entrada y salida, y finalmente visualizamos los resultados obtenidos para posteriormente establecer conclusiones.

2.8.1 PARÁMETROS ESCOGIDOS COMO VARIABLES DE ENTRADA

Una vez organizados los datos del procesamiento, primero definimos variables de entrada en el modelo. En este caso para el desarrollo de este trabajo las variables de entrada son las potencias medidas, tales como: rssi_strongest, rssi y rsrq.

rssi_strongest	mcc	node_id	cid	rssi	rsrq
-101	740	28235	191	-99	-17
-99	740	28235	191	-98	-17
-99	740	28235	191	-98	-17
-99	740	28235	191	-98	-17
-99	740	28235	191	-102	-18
-99	740	28235	191	-102	-18
-99	740	28235	191	-102	-18
-101	740	28235	191	-99	-15
-101	740	28235	191	-99	-15
-101	740	28235	191	-99	-15
-99	740	28235	191	-100	-15
-99	740	28235	191	-100	-15
-99	740	28235	191	-100	-15
-99	740	28235	191	-104	-17
-99	740	28235	191	-104	-17
-100	740	28235	191	-100	-20
-100	740	28235	191	-100	-20
-100	740	28235	191	-100	-20

Figura 2.24: Variables de entrada del modelo

2.8.2 PARÁMETRO ESCOGIDO COMO VARIABLE DE SALIDA

El parámetro definido como salida es “HO_vl”, que prácticamente es una nueva variable numérica con valores lógicos de “0” y “1”, esta variable es determinada a partir de la condicional para establecer la presencia de handover en los datos recolectados.

hand_over						zone_ho	HO_EoF	HO_vl
0	0	0	0	0	0	0	Fallido	0
0	0	0	0	0	0	0	Fallido	0
0	0	0	0	0	0	0	Fallido	0
0	0	0	0	0	0	0	Fallido	0
0	0	0	0	0	0	1	Exitoso	1
0	0	0	0	0	1	1	Exitoso	1
0	0	0	0	1	1	1	Exitoso	1
0	0	0	1	1	1	1	Exitoso	1
0	0	1	1	1	1	1	Exitoso	1
1	1	1	1	1	1	1	Exitoso	1
0	0	1	1	1	1	1	Exitoso	1
0	0	0	1	1	1	1	Exitoso	1
0	0	0	0	1	1	1	Exitoso	1
0	0	0	0	0	1	1	Exitoso	1
0	0	0	0	0	0	1	Exitoso	1
0	0	0	0	0	0	0	Fallido	0
0	0	0	0	0	0	0	Fallido	0
0	0	0	0	0	0	0	Fallido	0
0	0	0	0	0	0	0	Fallido	0

Figura 2.25: Variable de salida del modelo

2.8.3 DESCRIPCIÓN DE LA CODIFICACIÓN EN RSTUDIO

Para implementar el modelo “Regresión Lineal” en Rstudio, inicialmente debemos importar la base de datos, para ello es importante instalar previamente las librerías “*openxlsx*” y “*readxl*”, las cuales nos van a permitir abrir y manipular archivos Excel. Una vez instaladas, las cargamos a la sesión con su respectiva sintaxis.

```
##Codificación Modelo Regresión Lineal
#Jhon Cando

#Librerías para manipular archivos Excel
library("openxlsx")#Librería utilizada para leer y escribir archivos de Excel (.xlsx)
library("readxl")#Librería utilizada para utilizada para leer datos de archivos de Excel
```

Código 2.1: Importación de la base de datos en Rstudio

Para seleccionar el archivo, utilizamos la función “*read_xlsx*”, la cual tiene como único parámetro de entrada la ubicación del archivo .xlsx. Para encontrar la ubicación del archivo ejecutamos la función “*file.choose()*” en la consola. La cual nos permite seleccionar un archivo local usando una interfaz gráfica. Luego de abrir el archivo, se presentará por consola Figura: 36, la ruta de ubicación necesaria para llevarla al script [52].

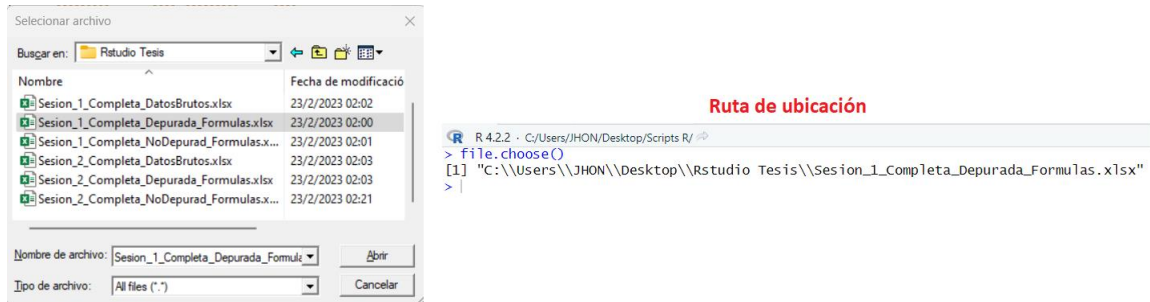


Figura 2.26: Ruta absoluta de ubicación de la base de datos

```
#Seleccionamos el registro
#file.choose()#Permite seleccionar un archivo usando una interfaz gráfica
#Para leer el conjunto de datos de una hoja específica.
ruta1_data1<-read_xlsx("C:\\Users\\JHON\\Desktop\\Rstudio
Tesis\\CodificacionRStudio\\Sesion_1_Completa_Depurada_Valores.xlsx")
```

Código 2.2: Selección del archivo mediante la ruta absoluta en Rstudio

Posteriormente, procedemos a codificar el proceso de Machine Learning con técnica de aprendizaje supervisado y modelo de regresión lineal. Para ello, llamamos a la librería *recipes*, la cual nos permite crear recetas de preprocesamiento a un conjunto de datos para transformarlo y prepararlo para su análisis. Luego mediante la función *selec* procedemos a escoger las columnas: *rssi_strongest*, *rssi*, *rsrq* y *hand_over* del conjunto de datos.

Con la función *sample_frac*, muestreamos una fracción aleatoria de filas del conjunto de datos. En esta ocasión la función seleccionó aleatoriamente 0.1% de las filas, lo que resultó en un conjunto de datos de 16 filas. Y se indica también, que el muestreo no se permite con reemplazo.

```
#seleccionamos columnas específicas de un conjunto de datos
data_train1<-select(ruta1_data1,c(rssi_strongest,rssi,rsrq,hand_over))
head(data_train1)#Visualizamos las primeras filas del conjunto de datos
#muestreamos una fracción aleatoria de filas de un conjunto de datos.
data_muestra1<-sample_frac(data_train1,size=.001,replace=F)
```

Código 2.3: Preprocesamiento del conjunto de datos en Rstudio

En la creación de la receta de procesamiento de datos utilizamos la función *recipe*, esta función nos permite transformar y preparar el conjunto de datos para el análisis del modelo. La función *recipe()* se utiliza para crear la receta inicial, y los operadores *%>%* se utilizan para aplicar las transformaciones y ajustes a la receta.

La receta comienza seleccionando todas las variables predictoras (*hand_over~.*) del conjunto de datos *data_muestra1*. A continuación, con la función *step_zv(all_numeric(),-*

`all_outcomes()`), se eliminan las variables numéricas con varianza cercana a cero o varianza cero que no son variables de resultado. Luego, se agrega una nueva columna `distcic` al conjunto de datos utilizando la función `step_mutate()` que genera valores aleatorios entre 0 y 1. A continuación, se agrega una nueva columna `cic` al conjunto de datos utilizando la función `step_mutate()`, que asigna valores a esta columna en función de una condición sobre los valores en la columna `distcic`. Luego, utilizando la función `step_select(-distcic)`, se elimina la columna `distcic` del conjunto de datos. Finalmente, se prepara la receta utilizando la función `prep()`.

```
# herramienta útil en Rstudio para preparar y transformar datos antes de aplicar
# modelos estadísticos.
# creamos la receta de preprocesamiento de datos
mk_rec1 <- recipe(hand_over ~ ., data = data_muestra1) %>%
  step_zv(all_numeric(), -all_outcomes()) %>%
  step_mutate(distcic = runif(nrow(data_muestra1), 0, 1)) %>%
  step_mutate(cic = as.factor(if_else(distcic > 0.8, "Norm", "Rec"))) %>%
  step_select(-distcic) %>%
  prep()
mk_rec1
```

Código 2.4: Creación de la receta en Rstudio

Para la última sección del código, se utiliza la función `juice()`, para extraer los resultados de las operaciones encadenadas de la receta `mk_rec1` y se almacena el resultado en el objeto `data_coef`. Luego se carga la librería `caret` que se utiliza para entrenar modelos de aprendizaje automático supervisado.

A continuación, se establece una semilla inicial para la generación de números aleatorios utilizando la función `set.seed(123)`. Se utiliza la función `createDataPartition()`, para dividir el conjunto de datos `data_coef` en un conjunto de prueba y un conjunto de entrenamiento de manera aleatoria, con una proporción del 80% para entrenamiento y el 20% restante para prueba. La división se almacena en el objeto `training_sample1`.

A continuación, se establece la definición del grupo de entrenamiento llamado `training_base1` y el grupo de prueba denominado `test_base1`. Después, se procede a ajustar el un modelo de regresión lineal con la variable `hand_over` como salida, utilizando la función `lm()`. El modelo ajustado se almacena en el objeto `modH_O`.

```
# proporciona un estilo de programación de "tuberías" para encadenar operaciones de
# una manera más clara y legible
library("magrittr")
data_coef <- juice(mk_rec1) # Extraemos los resultados de las operaciones encadenadas
data_coef
# La librería a continuación facilita el proceso de entrenamiento de modelos de
# machine learning supervisado
```

```

library(caret)
set.seed(123)#establecemos una semilla inicial para la generación de números
aleatorios
training_sample1<-data_coef$hand_over%>%
  #se divide un conjunto entrenamiento y un conjunto de prueba de manera aleatoria
  createDataPartition(p=0.8,list=F)
#dividimos el conjunto de datos data_coef
training_base1<-data_coef[training_sample1,]
test_base1<-data_coef[-training_sample1,]

#ajustamos un modelo de regresión lineal con variable de salida hand_over
modH_O<-lm(hand_over~.,data=training_base1)
modH_O

```

Código 2.5: Fase de entrenamiento del modelo en Rstudio

Hay diversas maneras de valorar un modelo de Machine Learning, y la selección de la técnica de evaluación más acorde al análisis dependerá de la naturaleza del problema a resolver y del tipo de datos empleados.

Algunas de las técnicas más comunes son:

- **Validación cruzada:** Es una técnica utilizada para evaluar la capacidad de generalización de un modelo de Machine Learning, para ello se divide el conjunto de datos en k subconjuntos o "pliegues" y se realiza k experimentos, en cada uno de los cuales se entrena el modelo con k-1 pliegues y se evalúa con el pliegue restante, luego se promedian las métricas de evaluación obtenidas en cada experimento [53].
- **División en entrenamiento y prueba:** En esta técnica se dividen los datos en dos subconjuntos de entrenamiento y otro de prueba. El modelo se entrena utilizando el subconjunto de entrenamiento y se evalúa utilizando el subconjunto de prueba. La división de los subconjuntos dependerá del tamaño de la base de datos y de la complejidad del modelo [54].
- **Matriz de confusión:** Consiste en una tabla que exhibe el número de predicciones correctas e incorrectas que hace el modelo. Es especialmente útil cuando se trabaja con problemas de clasificación [55].
- **Curva ROC y AUC:** La curva ROC (Característica operativa del receptor) es el gráfico de una curva que refleja la relación entre la tasa de verdaderos positivos y la tasa de falsos positivos a distintos niveles de umbral de decisión. La métrica del área bajo la curva (AUC) evalúa la capacidad del modelo para discriminar entre las clases positivas y negativas [56].

- **Métricas de evaluación específicas:** Dependiendo de la naturaleza del problema que se esté tratando, se pueden aplicar diversas métricas de evaluación útiles, como el error cuadrático medio (RMSE) para problemas de regresión, la precisión y el recall para problemas de clasificación, o la función de pérdida para problemas de aprendizaje profundo [57].

En nuestro modelo, utilizamos el error cuadrático medio como métrica de evaluación. Para la implementación en Rstudio, primero se calcula las predicciones del modelo ajustado con los datos de prueba, luego las almacena en un data frame y finalmente calcula el RMSE entre las predicciones y los valores observados de la variable "hand_over" en los datos de prueba.

```
##Cálculo de predicciones
library(Metrics)#libreria con funciones para evaluar y comparar modelos de aprendizaje
automático.

#se aplica la función predic al modelo utilizando los datos de prueba almacenados en
test_base1
predic_1<-modH_O%>%
  predict(test_base1)
#El vector de predicciones almacenado en "predic_1" se convierte en un data frame de
una sola columna
predic1_df<-data.frame(predic_1)
#Se imprimen las predicciones por consola
predic_1
predic1_df
#se establece la semilla inicial para la generación de números aleatorio
help(set.seed)
```

Código 2.5: Cálculo de las predicciones en Rstudio

La métrica utilizada para evaluar el modelo es el error cuadrático medio, el cual lo calculamos por medio de la siguiente sección de Código.

La función *rmse()* calcularía el RMSE y lo devolvería como resultado, pero para ello, se necesita un vector de valores de predicción en este caso *predic_1* y un vector de valores reales *test_base1\$HO_vl*.

```
#cálculo del RMSE entre los valores observados de la variable de respuesta y las
predicciones del modelo.
rmse(predic_1,test_base1$HO_vl)
```

Código 2.6: Métricas de evaluación del modelo en Rstudio

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 RESULTADOS

En este capítulo se exhiben los resultados obtenidos en este trabajo del análisis del conjunto de datos de las distintas rutas recorridas en el sector San Juan (ruta 1 y ruta 2). Como parte de este análisis se procesan los datos con el modelo de regresión para obtener el algoritmo básico de handover, para ello, se utilizan las métricas correspondientes para la evaluación del modelo.

3.1.1 SECTOR SAN JUAN – RUTA 1

El modelo de aprendizaje supervisado mediante la técnica de regresión se obtiene para predecir si el handover es exitoso o fallido; para esto, se utilizan valores numéricos a partir de los datos recopilados en la ruta 1.

En la Figura: 3.1 se puede apreciar el resumen de la receta implementada en la codificación, en donde se puede observar que se tienen tres variables de entrada (rssi, rssi_strongest y rsrq) y una de salida que corresponde a la predicción del handover exitoso o fallido.

```
R 4.2.2 · C:/Users/JHON/Desktop/Scripts R/ ↗
  role #variables
outcome          1
predictor        3

Training data contained 16 data points and no missing data.

Operations:

Zero variance filter removed <none> [trained]
Variable mutation for ~runif(nrow(data_muestra1), 0, 1) [trained]
Variable mutation for ~as.factor(if_else(distcic > 0.8, "Norm... [trained]
Variables selected rssi_strongest, rssi, rsrq, hand_over, cic [trained]
```

Figura 3.1: Visualización de la receta implementada

Seguidamente en la Figura: 3.2, se muestra el modelo de regresion lineal obtenido como resultado de la relación entre las variables de entrada para predecir el handover en base a los datos recolectados.

```

Console Terminal Background Jobs
R 4.2.2 · C:/Users/JHON/Desktop/Scripts R/
+ createDataPartition(p=0.8, list=F)
> training_base1<-data_coef[training_sample1,]
> test_base1<-data_coef[-training_sample1,]
> modH_O<-lm(hand_over~., data=training_base1)
> modH_O

Call:
lm(formula = hand_over ~ ., data = training_base1)

Coefficients:
(Intercept)  rssi_strongest          rssi          rsrq          cicRec
-1.40846     -0.03384         0.02390     -0.06027     -0.51650

```

Figura 3.2: Resultado del modelo regresión para la ruta 1

Luego, se evalúa el algoritmo obtenido en MatLab, ya que con esta herramienta se puede automatizar de mejor manera el proceso.

Primero se carga el data set en el espacio de trabajo.

```

%Codificación para automatizar el proceso de prueba para la ecuación obtenida
clear all, close all, clc
%Cargamos la base de datos
Base_datos=xlread( "Sesion_R1_Completa_Depurada_Valores.xlsx");

```

Código 3.1: Cargamos la base de datos en Matlab

Se extraen las variables de salida y de entrada que están involucradas en el modelo.

```

%Extraemos las variables de salida con las que esta implementado el modelo
rssi_stronges=Base_datos(:,9);%Extraemos la columna 9 perteneciente a rssi_stronges
rssi=Base_datos(:,13);%Extraemos la columna 13 perteneciente a rssi
rsrq=Base_datos(:,14);%Extraemos la columna 14 perteneciente a rsrq

%Extraemos la variable de salida
HO_vl=Base_datos(:,26);%Extraemos la columna 26 perteneciente a hand_over

```

Código 3.2: Extracción de variables de entrada y salida del modelo en Matlab

Se desarrolla un bucle para recorrer cada fila del conjunto de datos extraídos. Posteriormente, se tipifica la ecuación dentro del bucle acoplado a cada iteración a cada fila del conjunto de variables y se redondea el resultado de la ecuación al entero más próximo.

```

%A continuación se establece un lazo para recorrer los datos
for i=1: length(HO_vl)
%Ecuación de Regresión Lineal
Ec_LinealRegresion(i,:)= 0.15017 + 0.03438*rssi_stronges(i,:) - 0.03899*rssi(i,:) +
0.02894*rsrq(i,:);

```

```
end
%Redondeamos los valores obtenidos
salida=round(Ec_LinealRegresion);
```

Código 3.3: Implementación de la ecuación lineal obtenida en Matlab

Para comparar los dos vectores se establece un operador lógico entre la salida del modelo y el resultado obtenido al aplicar la ecuación del modelo al conjunto de datos. Luego, se determina la cantidad de predicciones correctas y predicciones incorrectas. Finalmente, se expresan los resultados en porcentaje.

```
%% Comparación de resultados
comparacion= HO_vl==salida;%Condicion que arroja un estado lógico de verdad o falso
valor_erroneos=sum(comparacion==0);%Operador que indica los valores errados
valor_buenos=sum(comparacion==1);%Operador que indica los valores buenos

tamano=length(Base_datos);% tamaño del conjunto de datos
Porcentajes_valor_prediccion_buena=valor_buenos/tamano*100%Calculo del
porcentaje de valores bien predichos
Porcentajes_valor_prediccion_mala=valor_erroneos/tamano*100%Calculo del
porcentaje de valores mal predichos
```

Código 3.4: Cálculo del porcentaje de los valores predictivos en Matlab

Se debe tener en cuenta que para este tipo de vectores no es posible emplear la función `corr2` de MatLab, ya que los vectores están conformados por 0s y al realizar el proceso de correlación se obtendrían valores indefinidos.

En la Figura: 3.3 se puede apreciar la comparación entre la variable de salida del modelo en el data set y la obtenida a través de la ecuación en Matlab.

	1	2		1	2		1	2
1		1		1	1		1	1
2		1		2	1		2	1
3		1		3	1		3	1
4		1		4	1		4	1
5		1		5	1		5	1
6		1		6	0		6	0
7		1		7	0		7	0
8		1		8	0		8	0
9		1		9	0		9	0
10		1		10	0		10	0
11		0		11	0		11	1
12		0		12	0		12	1
13		0		13	0		13	1
14		0		14	0		14	1
15		0		15	0		15	1

Figura 3.3: Representación visual de la salida del modelo para la ruta 1

En la Figura 3.4, se puede apreciar en términos de porcentaje la exactitud de la predicción planteada por la ecuación obtenida por el modelo regresión.

```

Command Window

Porcentajes_valor_prediccion_buena =

  77.3501

Porcentajes_valor_prediccion_mala =

  22.6499

```

Figura 3.4: Representación porcentual de predicciones correctas y erróneas para la ruta 1

3.1.1.1 CÁLCULO DEL ERROR RMSE PARA EL MODELO DE LA RUTA 1

Para calcular el RMSE o error cuadrático medio en Rstudio, se usa la función `rmse` como se indica en la Figura 3.5. Al presentar un valor bajo, se puede inferir que el modelo es más preciso, ya que un valor alto indica que el modelo tiene un mayor error en las predicciones.

```

> rmse(predic_1, test_base1$HO_v1)
[1] 0.4020335

```

Figura 3.5: Cálculo del RMSE en Rstudio para la ruta 1

3.1.2 SECTOR SAN JUAN – RUTA 2

De la misma manera, el modelo de aprendizaje supervisado de regresión obtenido para predecir valores numéricos a partir de los datos recopilados en la ruta 2 es el siguiente.

Debido a los resultados obtenidos, se encontró que la variable de entrada rsrq presentaba una considerable cantidad de errores en las mediciones, por lo que no se tomó en cuenta en el análisis. A continuación, se detallan los resultados del proceso.

```
R 4.2.2 · C:/Users/JHON/Desktop/Scripts R/
> #ajustamos un modelo de regresion lineal con variable de salida hand_over
> modH_0<-lm(HO_v1~.,data=training_base1)
> modH_0

Call:
lm(formula = HO_v1 ~ ., data = training_base1)

Coefficients:
(Intercept)  rssi_strongest          rssi          cicRec
-0.2560609   -0.0002748   -0.0024771    0.1008567
```

Figura 3.6: Resultado del modelo de regresión para la ruta 2

Como se puede ver el modelo ahora entrega una ecuación más simple y al estar conformada por menos variables de entrada, se obtienen mejores resultados.

HO_v1		salida		comparacion	
1	2	1	2	1	2
172	0	172	0	172	1
173	0	173	0	173	1
174	1	174	0	174	0
175	1	175	0	175	0
176	1	176	0	176	0
177	1	177	0	177	0
178	1	178	0	178	0
179	1	179	0	179	0
180	0	180	0	180	1
181	0	181	0	181	1
182	0	182	0	182	1
183	0	183	0	183	1
184	0	184	0	184	1
185	0	185	0	185	1
186	0	186	0	186	1

Figura 3.7: Representación visual de la salida del modelo para la ruta 2

En la Figura 3.8 se puede apreciar en términos de porcentaje la exactitud de la predicción planteada por la ecuación obtenida con el modelo de regresión.

```
Command Window
Porcentajes_valor_prediccion_buena =
90.2591
Porcentajes_valor_prediccion_mala =
9.7409
```

Figura 3.8: Representación porcentual de predicciones correctas y erróneas para la ruta 2

3.1.2.1 CÁLCULO DEL ERROR RMSE PARA EL MODELO DE LA RUTA 2

Como se puede estimar en la Figura 3.9, el error cuadrático medio obtenido es insignificante, lo que indica que el modelo de predicción es muy eficiente y no presenta errores en las predicciones.

```
> rmse(predic_1, test_base1$HO_v1)
[1] 0.07709241
```

Figura 3.9: Cálculo del RMSE en Rstudio para la ruta 1

3.1.3 ALGORITMO BÁSICO DEL HANDOVER

El algoritmo consiste en: recolectar los datos, limpiar los datos, preprocesarlos con el fin de incluir el atributo de interés que no se encuentra en el data set inicial y que corresponde al “hand_over”, luego se implementa el modelo de regresión. El proceso de obtención del modelo consiste en: leer los datos, obtener y definir las variables de salida y entrada, obtener los objetos mk_rec1 y data_coef para crear el conjunto de prueba y entrenamiento y así obtener el modelo. Finalmente, para la evaluación del modelo se procede a: leer el data set, extraer las variables que involucra el modelo, tipear el algoritmo y ejecutarlo para obtener la salida. Por último, en base a los resultados obtenidos en la evaluación del modelo, el algoritmo básico del handover recomienda o no realizar el handover a la celda objetivo (candidata), ya que existe la probabilidad de que el traspaso de una celda a otra sea exitoso o fallido. En caso de que exista una probabilidad alta de que el handover sea exitoso, el algoritmo sugerirá conectarse a la celda más cercana.

Se muestra a continuación el diagrama de flujo correspondiente al algoritmo que se propone.

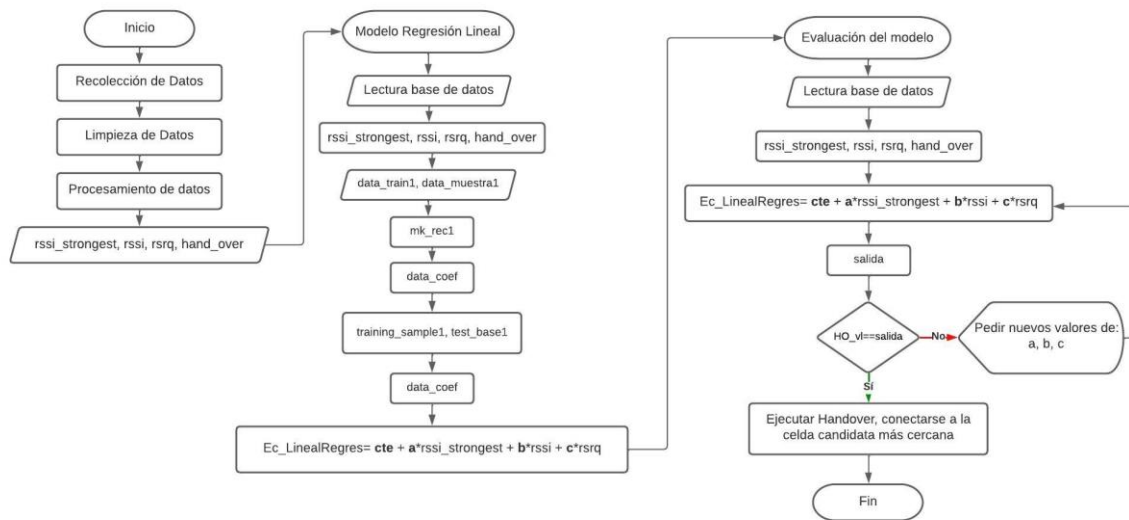


Figura 3.10: Diagrama de flujo correspondiente al algoritmo propuesto

- Determinación de la celda candidata más cercana

A continuación, se incluye el código de determinación de la celda candidata más cercana para realizar el traspaso.

Para cargar la base de datos, procedemos a utilizar la función *xlsread* de Matlab con el nombre del archivo y su respectiva extensión.

```
%Análisis de la celda objetivo
clear all, close all, clc
%Cargamos la base de datos
Base_datos=xlsread("Sesion_R1_Completa_Depurada_Valores.xlsx");
```

Código 3.5: Importación de la base de datos en Matlab

	A	B	C	D	E	F	G	H	I	J	K	L
	SesionR1CompletaDepuradaValores											
	report	Fecha	Hora	day	tiempo_ho...	tiempo	Distancia	Velocidadk...	rss_i_strong...	mcc	node_id	cid
	Number	Datetime	Text	Categorical	Number	Number	Number	Number	Number	Number	Number	Number
1	report	Fecha	Hora	day	tiempo_hor...	tiempo	Distancia	Velocidad [...]	rss_i_strong...	mcc	node_id	cid
2	0	21-nov-2022	14:03:11	lunes	14.0531	2.7778e-04	869.1784	2.4144e-04	-114	740	28334	190
3	1	21-nov-2022	14:03:12	lunes	14.0533	2.7778e-04	0	0	-114	740	28334	190
4	2	21-nov-2022	14:03:13	lunes	14.0536	2.7778e-04	0	0	-114	740	28334	190
5	3	21-nov-2022	14:03:14	lunes	14.0539	2.7778e-04	305.5794	8.4883e-05	-114	740	28334	190
6	4	21-nov-2022	14:03:15	lunes	14.0542	0.0156	203.4721	0.0032	-114	740	28334	190
7	5	21-nov-2022	14:04:11	lunes	14.0697	2.7778e-04	4.5433	1.2620e-06	-104	740	28235	191
8	6	21-nov-2022	14:04:12	lunes	14.0700	2.7778e-04	1.1721	3.2558e-07	-104	740	28235	191
9	7	21-nov-2022	14:04:13	lunes	14.0703	2.7778e-04	0.7642	2.1229e-07	-104	740	28235	191
10	8	21-nov-2022	14:04:14	lunes	14.0706	2.7778e-04	0.4145	1.1515e-07	-101	740	28235	191
11	9	21-nov-2022	14:04:15	lunes	14.0708	2.7778e-04	0.9269	2.5748e-07	-101	740	28235	191

Figura 3.11: Carga del dataset en el entorno de Matlab

Posteriormente, creamos un vector de string, con el nombre de las respectivas estaciones bases, respetando el orden que se estableció en la base de datos y exportamos todas las

columnas pertinentes a los atributos creados acorde a la distancia. En este caso, se extraemos todas las columnas de 33 a 44. Luego definimos el tamaño del conjunto de datos para análisis posteriores.

```
%Definimos los identificadores de nodo en un vector de strings
Identificadores_celda=["eNB28834" "eNB28235" "eNB28542" "eNB28875" "eNB28059"
"eNB28387" "eNB28322" "eNB28542" "eNB28334" "eNB28010" "eNB28200"
"eNB28030"];

%Extraemos las distancias de la estacion movil a las estaciones base
distancias=Base_datos(:,33:44);
tamano_bd=length(distancias);
```

Código 3.6: Determinación de los identificadores de nodo en Matlab

	1	2	3	4	5	6	7	8	9	10	11	12
1	1.7492e+03	1.1127e+03	2.3538e+03	1.4116e+03	1.6980e+03	1.8502e+03	1.9137e+03	2.3538e+03	1.0946e+03	1.0594e+03	1.4241e+03	777.0886
2	935.5405	514.3172	1.6056e+03	630.0107	956.3735	1.0464e+03	1.0611e+03	1.6056e+03	340.2095	263.5837	719.3823	758.6529
3	935.5405	514.3172	1.6056e+03	630.0107	956.3735	1.0464e+03	1.0611e+03	1.6056e+03	340.2095	263.5837	719.3823	758.6529
4	935.5405	514.3172	1.6056e+03	630.0107	956.3735	1.0464e+03	1.0611e+03	1.6056e+03	340.2095	263.5837	719.3823	758.6529
5	662.5413	561.7963	1.3540e+03	411.4313	734.7002	776.6218	760.5217	1.3540e+03	275.0566	269.4514	621.3493	938.0781
6	469.2509	713.7197	1.1666e+03	289.7533	578.0036	583.7221	557.9753	1.1666e+03	346.5812	448.7049	690.2408	1.0463e+03
7	470.1367	711.8007	1.1678e+03	293.5890	580.6159	584.5845	556.6513	1.1678e+03	350.6452	447.8981	686.7036	1.0504e+03
8	469.7625	711.9115	1.1674e+03	294.1344	580.7209	584.2032	555.7958	1.1674e+03	351.8147	448.3053	686.2882	1.0516e+03
9	469.0208	712.5924	1.1667e+03	293.7568	580.1163	583.4603	555.0554	1.1667e+03	352.1823	449.0526	686.7461	1.0520e+03
10	468.6500	712.9122	1.1663e+03	293.6359	579.8516	583.0882	554.6411	1.1663e+03	352.4563	449.4295	686.9167	1.0522e+03

Figura 3.12: Distancias extraídas del dataset

A continuación, se determina la distancia más pequeña por cada fila del conjunto de datos y almacenamos dicha distancia en la variable *dismin*.

Luego creamos un bucle iterativo, donde se obtendrá una matriz conformada por valores lógicos, cuya condición indica, si el valor es verdadero marcará la posición del nodo con "1", caso contrario marcará con "0" como se muestra en la Figura: 3.9.

```
%Selecciona la distancia menor por fila
for i=1:tamano_bd
    dismin(i,:)=min(distancias(i,:));
end
%Asignamos un valor lógico de 1 en las columnas que se tengan la menor distancia a la estación base
for i=1:tamano_bd
    for j=1:12
        matriz_eNb_logicos(i,j)= dismin(i)==distancias(i,j);
    end
end
```

Código 3.7: Determinación de la menor distancia y creación de la matriz lógica en Matlab

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	0	1	0	0
6	0	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0	0	0	0
9	0	0	0	1	0	0	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	0	0	0

Figura 3.13: Matriz lógica con valores de 1 en la distancia mínima

Finalmente, para obtener la posición donde se cumple la condición lógica mencionada antes, utilizamos el comando *find* por cada fila y la almacenamos en la variable *posicion*.

Para la selección de la celda candidata más cercana, accedemos al nodo resultante a través de la indexación de variable *posicion* en la variable *Identificadores_celda*. Los resultados se almacenarán en la variable *celda_seleccioanda*. El resultado se muestra en la Figura:

```

%Encuentra la posición en la matriz de logica matriz_eNb_logicos
for i=1:tamano_bd
    posicion(i)= find(matriz_eNb_logicos(i,:)==1);
end

%Se asigna la celda candidata más cercana
for i=1:tamano_bd
    celda_seleccioanda(i)= Identificadores_celda(posicion(i));
end
celda_seleccioanda=celda_seleccioanda';

```

Código 3.8: Determinación de la celda candidata más cercana en Matlab

	1	2	3	4	5	6	7	8	9	10	11	12
1	1.7492e+03	1.1127e+03	2.3538e+03	1.4116e+03	1.6980e+03	1.8502e+03	1.9137e+03	2.3538e+03	1.0946e+03	1.0594e+03	1.4241e+03	777.0886
2	935.5405	514.3172	1.6056e+03	630.0107	956.3735	1.0464e+03	1.0611e+03	1.6056e+03	340.2095	263.5837	719.3823	758.6529
3	935.5405	514.3172	1.6056e+03	630.0107	956.3735	1.0464e+03	1.0611e+03	1.6056e+03	340.2095	263.5837	719.3823	758.6529
4	935.5405	514.3172	1.6056e+03	630.0107	956.3735	1.0464e+03	1.0611e+03	1.6056e+03	340.2095	263.5837	719.3823	758.6529
5	662.5413	561.7963	1.3540e+03	411.4313	734.7002	776.6218	760.5217	1.3540e+03	275.0566	269.4514	621.3493	938.0781
6	469.2509	713.7197	1.1666e+03	289.7533	578.0036	583.7221	557.9753	1.1666e+03	346.5812	448.7049	690.2408	1.0463e+03
7	470.1367	711.8007	1.1678e+03	293.5890	580.6159	584.5845	556.6513	1.1678e+03	350.6452	447.8981	686.7036	1.0504e+03
8	469.7625	711.9115	1.1674e+03	294.1344	580.7209	584.2032	555.7958	1.1674e+03	351.8147	448.3053	686.2882	1.0516e+03

	1
1	eNB28030
2	eNB28010
3	eNB28010
4	eNB28010
5	eNB28010
6	eNB28875
7	eNB28875
8	eNB28875
9	eNB28875
10	eNB28875
11	eNB28875
12	eNB28875
13	eNB28875
14	eNB28875
15	eNB28875
16	eNB28875
17	eNB28875
18	eNB28875
19	eNB28875
20	eNB28875

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	0	1	0	0
6	0	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0	0	0	0

Figura 3.14: Representación visual de determinación de la celda candidata más cercana

3.2 CONCLUSIONES

El modelo de predicción de handover exitoso o fallido obtenido mediante la técnica de regresión para la ruta 1, se apega bastante a los datos recolectados, ya que al evaluarlo en Matlab se observa que casi el 80% de las predicciones fueron correctas con un valor de RMSE de 0.402.

El modelo de predicción de handover para la ruta 2, es sumamente similar a los datos recolectados, ya que al evaluarlo en Matlab se observa que el 90% de las predicciones fueron correctas con un valor de RMSE de 0.077.

Se pudo comprobar que el modelo de regresión lineal implementado mediante Rstudio con variables de entrada, rssi_strongest, rssi, rsrq y la variable de salida hand_over, estima los coeficientes de la ecuación lineal que mejor se ajustan a los datos recolectados.

3.3 RECOMENDACIONES

Se recomienda en la etapa de recolección de las mediciones de radio frecuencia se utilice un dispositivo con capacidad de procesamiento mayor a 6GB, ya que de esta manera se puede asegurar que las aplicaciones que estén en segundo plano también estén recopilando información coherente y no errónea.

Se recomienda, en un trabajo futuro, comprobar la eficiencia del algoritmo básico propuesto en este estudio con el uso de datos recolectados en otras zonas.

Sería conveniente aplicar otra técnica de Machine Learning para la obtención de un modelo de predicción y evaluar mediante las métricas respectivas su efectividad.

4 REFERENCIAS BIBLIOGRÁFICAS

[1] L. Delgado, «LECTURAS La evolución de la telefonía móvil La guerra de los celulares», Accedido: 1 de marzo de 2023. [En línea]. Disponible en: https://www.academia.edu/6042142/LECTURAS_La_evoluci%C3%B3n_de_la_telefon%C3%ADa_m%C3%B3vil_La_guerra_de_los_celulares

[2] «Sistemas de telefonía y comunicaciones móviles». <https://mastermoviles.gitbook.io/tecnologias2/sistemas-de-telefon%C3%ADa-y-comunicaciones-moviles> (accedido 16 de febrero de 2023).

[3] A. F. Durán y M. M. García, «EFECTO DEL TAMAÑO DE LA VENTANA DE PROMEDIADO Y LA HISTERESIS VARIABLE EN ALGORITMOS DE HANDOVER HORIZONTAL EN REDES CONVERGENTES».

- [4] R. J. Acosta Plazas, «Análisis del algoritmo de handover reactivo y proactivo en sistemas femtocelda para la tecnología 4G (Long Term Evolution-Advanced)», bachelorThesis, Quito, 2018., 2018. Accedido: 16 de febrero de 2023. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/19211>
- [5] D. Ordóñez, H. Ávalos, y E. Gomez, «High degree complete orthogonal polynomials to improve distance prediction based on RSSI», RISTI - Rev. Iber. Sist. E Tecnol. Inf., pp. 584-594, ene. 2019.
- [6] admin, «▷ A cuanto camina una persona por hora | Actualizado febrero 2023», El deporte en una web, 21 de diciembre de 2022. <https://imdet.mx/caminar/a-cuanto-camina-una-persona-por-hora/> (accedido 16 de febrero de 2023).
- [7] M. Salinas y C. Silva, «Modelos de regresión y correlación II: regresión lineal múltiple», Cienc Trab, pp. 39-41, 2007.
- [8] CellMapper, «First time startup», CellMapper. https://www.cellmapper.net/First_Time_Startup, <https://www.cellmapper.net/it> (accedido 8 de febrero de 2023).
- [9] CellMapper, «Estimated Band», CellMapper. https://www.cellmapper.net/Estimated_Band, <https://www.cellmapper.net/it> (accedido 8 de febrero de 2023).
- [10] «Manual G-NetTrack – Gyokov Solutions». <https://gyokovsolutions.com/manual-g-nettrack/> (accedido 8 de febrero de 2023).
- [11] «Descarga de APK de NetMonitor Cell Signal Logging para Android», APKPure.com. https://apkpure.com/es/netmonitor-cell-signal-logging/ru.v_a_v.netmonitor (accedido 8 de febrero de 2023).
- [12] «Xiaomi Redmi Note 9S - Specs», 23 de marzo de 2020. <https://www.phonemore.com/specs/xiaomi/redmi-note-9s/> (accedido 9 de febrero de 2023).
- [13] «Xiaomi Redmi Note 9S - Características y especificaciones», DeviceSpecifications. <https://www.devicespecifications.com/es/model/74615336> (accedido 9 de febrero de 2023).
- [14] «R-intro-1.1.0-espanol.1.pdf». Accedido: 13 de febrero de 2023. [En línea]. Disponible en: <https://cran.r-project.org/doc/contrib/R-intro-1.1.0-espanol.1.pdf>
- [15] N. Matloff, The Art of R Programming: A Tour of Statistical Software Design. No Starch Press, 2011.
- [16] F. Hernández y O. Usuga, Manual de R. Accedido: 13 de febrero de 2023. [En línea]. Disponible en: <https://fhernanb.github.io/Manual-de-R/>
- [17] Rafa, «Qué es R y RStudio? ⚡ [curso de R Studio] 🇧🇷 Rafa González Gouveia», Rafa González Gouveia, 22 de marzo de 2019. <https://gonzalezgouveia.com/que-es-r-y-rstudio/> (accedido 13 de febrero de 2023).
- [18] G. B. B. y F. R. Bruzzone, Capítulo 4 Uso básico de RStudio | RStudio para Estadística Descriptiva en Ciencias Sociales. Accedido: 13 de febrero de 2023. [En línea]. Disponible en: <https://bookdown.org/gboccardo/manual-ED-UCH/uso-basico-de-rstudio.html>

- [19] «Instalación de R y Rstudio.» <https://estadistica-dma.ulpgc.es/cursoR4ULPGC/2-instalacion.html> (accedido 13 de febrero de 2023).
- [20] «What can you do with R? 5 Top R Ideas». <https://www.datacamp.com/blog/what-can-you-do-with-r> (accedido 20 de febrero de 2023).
- [21] S. B. Andrés, «Glossary:Mobile Country Code (MCC)», CROS - European Commission, 11 de junio de 2020. https://ec.europa.eu/eurostat/cros/content/Glossary%3AMobile_Country_Code_%28MCC%29_en (accedido 17 de febrero de 2023).
- [22] M. DEBUSSCHERE, «Glossary:Location Area Code/Tracking Area Code (LAC/TAC)», CROS - European Commission, 5 de junio de 2020. https://ec.europa.eu/eurostat/cros/content/Glossary%3ALocation_Area_Code/Tracking_Area_Code_%28LAC/TAC%29_en (accedido 17 de febrero de 2023).
- [23] «LTE Glossary». <https://www.winncom.com/en/lte/glossary/55/cid.html> (accedido 17 de febrero de 2023).
- [24] P. Kline, «Understanding RSSI, RSRP, and RSRQ - Welcome To The 5Gstore Blog Welcome To The 5Gstore Blog - Lastest News, Product Info & More», Welcome To The 5Gstore Blog, 8 de abril de 2021. <https://5gstore.com/blog/2021/04/08/understanding-rssi-rsrp-and-rsrq/> (accedido 17 de febrero de 2023).
- [25] O. Simpson y Y. Sun, «LTE RSRP, RSRQ, RSSNR and local topography profile data for RF propagation planning and network optimization in an urban propagation environment», Data Brief, vol. 21, ago. 2018, doi: 10.1016/j.dib.2018.08.137.
- [26] «Google Maps», Google Maps. <https://www.google.com/maps/search/google+maps/@-0.1858842,-78.540073,13z/data=!3m1!4b1> (accedido 17 de febrero de 2023).
- [27] CellMapper, «Movistar (Ecuador) - Cellular Coverage and Tower Map», CellMapper. <https://www.cellmapper.net/map?MCC=740&MNC=0>, <https://www.cellmapper.net/map> (accedido 17 de febrero de 2023).
- [28] «Explain various types of handoffs in GSM network». <https://www.ques10.com/p/24351/explain-various-types-of-handoffs-in-gsm-network-2/?> (accedido 21 de febrero de 2023).
- [29] J. Macalupu, «DistanciaGeográfica». 11 de octubre de 2022. Accedido: 21 de febrero de 2023. [En línea]. Disponible en: <https://github.com/jmacalupur/DistanciaGeografica>
- [30] «Azimut: cómo calcularlo». <https://www.certicalia.com/blog/como-calcular-azimut> (accedido 22 de febrero de 2023).
- [31] «Machine Learning - Libro online de IAAR». <https://iaarbook.github.io/machine-learning/> (accedido 22 de febrero de 2023).
- [32] BBVA, «Te contamos qué es el “machine learning” y cómo funciona», BBVA NOTICIAS, 8 de noviembre de 2019. <https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/> (accedido 22 de febrero de 2023).

- [33] «¿Qué es el aprendizaje supervisado?», TIBCO Software. <https://www.tibco.com/es/reference-center/what-is-supervised-learning> (accedido 22 de febrero de 2023).
- [34] «¿Qué es la regresión logística? - Explicación del modelo de regresión logística - AWS», Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/logistic-regression/> (accedido 22 de febrero de 2023).
- [35] «¿Qué es la regresión logística?», TIBCO Software. <https://www.tibco.com/es/reference-center/what-is-logistic-regression> (accedido 22 de febrero de 2023).
- [36] R. Gandhi, «Support Vector Machine — Introduction to Machine Learning Algorithms», Medium, 5 de julio de 2018. <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47> (accedido 22 de febrero de 2023).
- [37] «¿Qué es una red neuronal?» <https://la.mathworks.com/discovery/neural-network.html> (accedido 22 de febrero de 2023).
- [38] Enzyme, «Redes Neuronales Artificiales y Deep Learning, explicado para dummies». <https://enzyme.biz/blog/redes-neuronales-artificiales-y-deep-learning> (accedido 22 de febrero de 2023).
- [39] V. Roman, «Algoritmos Naive Bayes: Fundamentos e Implementación», Ciencia y Datos, 29 de abril de 2019. <https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fundamentos-e-implementaci%C3%B3n-4bcb24b307f> (accedido 22 de febrero de 2023).
- [40] «¿Qué es un árbol de decisión? | IBM». <https://www.ibm.com/es-es/topics/decision-trees> (accedido 22 de febrero de 2023).
- [41] sitio big data, «Árbol de decisión en Machine Learning (Parte 1)», sitiobigdata.com, 14 de diciembre de 2019. <https://sitiobigdata.com/2019/12/14/arbore-de-decision-en-machine-learning-parte-1/> (accedido 22 de febrero de 2023).
- [42] F. Parra, 6 Métodos de clasificación | Estadística y Machine Learning con R. Accedido: 22 de febrero de 2023. [En línea]. Disponible en: <https://bookdown.org/content/2274/metodos-de-clasificacion.html>
- [43] por J. A. Camacho, «Linear Discriminant Analysis | JacobSoft», 26 de octubre de 2018. https://www.jacobsoft.com.mx/es_mx/linear-discriminant-analysis/ (accedido 22 de febrero de 2023).
- [44] «¿Qué es el algoritmo de k vecinos más cercanos? | IBM». <https://www.ibm.com/es-es/topics/knn> (accedido 22 de febrero de 2023).
- [45] «Árboles de clasificación y K vecinos más cercanos». https://rstudio-pubs-static.s3.amazonaws.com/807626_862f88f88eef4e12b2792da722b674e7.html (accedido 22 de febrero de 2023).
- [46] «Ensemble Methods: Elegant Techniques to Produce Improved Machine Learning Results | Toptal®», Toptal Engineering Blog. <https://www.toptal.com/machine-learning/ensemble-methods-machine-learning> (accedido 22 de febrero de 2023).

- [47] «Tipos de aprendizaje en Machine Learning: supervisado y no supervisado», Think Big, 2 de diciembre de 2021. <https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/> (accedido 22 de febrero de 2023).
- [48] «Algoritmos de Clustering - Fernando Sancho Caparrini». <http://www.cs.us.es/~fsancho/?e=230> (accedido 22 de febrero de 2023).
- [49] J. M. Fanjul, «¿Qué es el Análisis de Componentes Principales y cómo reducir el tamaño de una base de datos?», Blog de Hiberus Tecnología, 19 de agosto de 2022. <https://www.hiberus.com/crecemos-contigo/analisis-de-componentes-principales/> (accedido 22 de febrero de 2023).
- [50] D. S. Team, «Descomposición del valor singular — Aprendizaje automático», DATA SCIENCE, 28 de marzo de 2021. <https://datascience.eu/es/aprendizaje-automatico/descomposicion-del-valor-singular/> (accedido 22 de febrero de 2023).
- [51] desarrollosidn, «Algoritmo PCA: de lo complejo a lo sencillo», LIS Data Solutions, 17 de septiembre de 2018. <https://www.lisdatasolutions.com/es/blog/algoritmo-pca-de-lo-complejo-a-lo-sencillo/> (accedido 22 de febrero de 2023).
- [52] E. Kosourova, «The Easy Way to Install a Package in R (with 8 Code Examples)», Dataquest, 13 de abril de 2022. <https://www.dataquest.io/blog/install-package-r/> (accedido 23 de febrero de 2023).
- [53] T. rédac, «Cross-Validation: definición e importancia en Machine Learning», Formation Data Science | DataScientest.com, 13 de mayo de 2022. <https://datascientest.com/es/cross-validation-definicion-e-importancia> (accedido 24 de febrero de 2023).
- [54] Na8, «Sets de Entrenamiento, Test y Validación», Aprende Machine Learning, 3 de marzo de 2020. <https://www.aprendemachinelearning.com/sets-de-entrenamiento-test-validacion-cruzada/> (accedido 24 de febrero de 2023).
- [55] J. I. B. Arce, «La matriz de confusión y sus métricas – Inteligencia Artificial –», Juan Barrios, 26 de julio de 2019. <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/> (accedido 24 de febrero de 2023).
- [56] L. Gonzalez, «Curvas ROC y Área bajo la curva (AUC)», Aprende IA, 31 de mayo de 2019. <https://aprendeia.com/curvas-roc-y-area-bajo-la-curva-auc-machine-learning/> (accedido 24 de febrero de 2023).
- [57] J. Saravanan, «How to Evaluate your Machine Learning Model.», Analytics Vidhya, 14 de junio de 2021. <https://medium.com/analytics-vidhya/how-to-evaluate-your-machine-learning-model-76a7671e9f2e> (accedido 24 de febrero de 2023).

5 ANEXOS

ANEXO I. Cronograma de actividades.

ANEXO II. Código Modelo regresión Lineal desarrollado en Rstudio para la ruta 1.

ANEXO III. Código Modelo regresión Lineal desarrollado en Rstudio para la ruta 2.

ANEXO IV. Código Comprobación modelo desarrollado en MatLab para la ruta 1.

ANEXO V. Código Comprobación modelo desarrollado en MatLab para la ruta 2.

ANEXO VI. Código Determinación de la celda candidata más cercana.

ANEXO I

Semana referencial / Etapas	Tareas específicas	Resultado esperado (si aplica)
1	Introducción y descripción de los parámetros de RF.	
2	Investigación de las diferentes herramientas para manejo de datos	
3	Inducción a las herramientas de análisis de datos	
4	Descargar y familiarizarse con el entorno de las aplicaciones.	Herramientas de análisis
5	Identificar los parámetros más importantes que recopilamos de las aplicaciones	
6	Proponer un algoritmo básico para la predicción de la celda destino	Algoritmo
7	Selección del algoritmo	
8	Análisis de los resultados obtenidos por el primer algoritmo	
9	Posibles ideas para mejorar el algoritmo.	Algoritmo
10	Evaluación del algoritmo propuesto	
11	Optimización del algoritmo	
12	Análisis de los resultados	
13	Revisión del Trabajo de Integración Curricular por parte de los profesores asignados (revisores).	
14	Revisión del Trabajo de Integración Curricular por parte de los profesores asignados (revisores).	
15	Revisión del Trabajo de Integración Curricular por parte de los profesores asignados (revisores).	
16	Ajustes finales y presentación del Trabajo de Integración Curricular.	Trabajo de Integración Curricular

Anexo II

```
##Codificación Modelo Regresión Lineal
#Jhon Cando

#Librerías para manipular archivos Excel
library("openxlsx")#Librería utilizada para leer y escribir archivos de Excel (.xlsx)
library("readxl")#Librería utilizada para utilizada para leer datos de archivos de Excel

#Seleccionamos el archivo
#file.choose()#Permite seleccionar un archivo usando una interfaz de usuario gráfica
#Para leer los datos de una hoja específica de un archivo de Excel.
ruta1_data1<-read_xlsx("C:\\Users\\JHON\\Desktop\\Rstudio
Tesis\\CodificacionRStudio\\Sesion_R1_Completa_Depurada_Valores.xlsx")

#Algoritmo de Machine Learning con técnica de aprendizaje supervisado y modelo de
regresión lineal.
library(recipes)#se utiliza para la preparación de datos

#seleccionamos columnas específicas de un conjunto de datos
data_train1<-select(ruta1_data1,c(rssi_strongest,rssi,rsrq,HO_vl))
head(data_train1)#Visualizamos las primeras filas del conjunto de datos
#muestreamos una fracción aleatoria de filas de un conjunto de datos.
data_muestra1<-sample_frac(data_train1,size=.001,replace=F)

# herramienta útil en RStudio para preparar y transformar datos antes de aplicar
modelos estadísticos.
#creamos la receta de preprocesamiento de datos
mk_rec1<-recipe(HO_vl~.,data= data_muestra1)%>%
  step_zv(all_numeric(),-all_outcomes())%>%
  step_mutate(distcic=runif(nrow(data_muestra1),0,1))%>%
  step_mutate(cic=as.factor(if_else(distcic>0.8,"Norm","Rec")))%>%
  step_select(-distcic)%>%
  prep()
mk_rec1

#proporciona un estilo de programación de "tuberías" para encadenar operaciones de
una manera más clara y legible
library("magrittr")
data_coef<-juice(mk_rec1)#Extraemos los resultados de las operaciones encadenadas
data_coef
#La librería a continuación facilita el proceso de entrenamiento de modelos de
aprendizaje automático supervisado
library(caret)
set.seed(123)#establecemos una semilla inicial para la generación de números
aleatorios
training_sample1<-data_coef$HO_vl%>%
  #se utiliza para dividir un conjunto de datos en un conjunto de entrenamiento y un
conjunto de prueba de manera estratificada y aleatoria
  createDataPartition(p=0.8,list=F)
#dividimos el conjunto de datos data_coef
training_base1<-data_coef[training_sample1,]
test_base1<-data_coef[-training_sample1,]
```

```

#ajustamos un modelo de regresión lineal con variable de salida hand_over
modH_O<-lm(HO_vl~.,data=training_base1)
modH_O

summary(modH_O)

##Metricas de evaluacion del modelo
library(Metrics)#libreria con funciones para evaluar y comparar modelos de aprendizaje
automático.

#se aplica la funcion predic al modelo utilizando los datos de prueba almacenados en
test_base1
predic_1<-modH_O%>%
  predict(test_base1)
#El vector de predicciones almacenado en "predic_1" se convierte en un data frame de
una sola columna
predic1_df<-data.frame(predic_1)
#Se imprimen las predicciones por consola
predic_1
predic1_df
#se establece la semilla inicial para la generación de números aleatorio
help(set.seed)

#cálculo del error cuadrático medio (RMSE) entre las predicciones del modelo y los
valores observados de la variable de respuesta
rmse(predic_1,test_base1$HO_vl)

#calculo el coeficiente de determinación (R2) entre las predicciones del modelo y los
valores observados de la variable de respuesta
R2(predic_1,test_base1$HO_vl)

```

Anexo III

```
##Codificación Modelo Regresión Lineal
#Jhon Cando

#Librerías para manipular archivos Excel
library("openxlsx")#Librería utilizada para leer y escribir archivos de Excel (.xlsx)
library("readxl")#Librería utilizada para utilizada para leer datos de archivos de Excel

#Seleccionamos el archivo
#file.choose()#Permite seleccionar un archivo usando una interfaz de usuario gráfica
#Para leer los datos de una hoja específica de un archivo de Excel.
ruta1_data1<-read_xlsx("C:\\Users\\JHON\\Desktop\\Rstudio
Tesis\\CodificacionRStudio\\Sesion_R2_Completa_Depurada_Valores.xlsx")

#Algoritmo de Machine Learning con técnica de aprendizaje supervisado y modelo de
regresión lineal.
library(recipes)#se utiliza para la preparación de datos

#seleccionamos columnas específicas de un conjunto de datos
data_train1<-select(ruta1_data1,c(rssi_strongest,rssi,HO_vl))
head(data_train1)#Visualizamos las primeras filas del conjunto de datos
#muestreamos una fracción aleatoria de filas de un conjunto de datos.
data_muestra1<-sample_frac(data_train1,size=.001,replace=F)

# herramienta útil en RStudio para preparar y transformar datos antes de aplicar
modelos estadísticos.
#creamos la receta de preprocesamiento de datos
mk_rec1<-recipe(HO_vl~.,data= data_muestra1)%>%
  step_zv(all_numeric(),-all_outcomes())%>%
  step_mutate(distcic=runif(nrow(data_muestra1),0,1))%>%
  step_mutate(cic=as.factor(if_else(distcic>0.8,"Norm","Rec")))%>%
  step_select(-distcic)%>%
  prep()
mk_rec1

#proporciona un estilo de programación de "tuberías" para encadenar operaciones de
una manera más clara y legible
library("magrittr")
data_coef<-juice(mk_rec1)#Extraemos los resultados de las operaciones encadenadas
data_coef
#La librería a continuación facilita el proceso de entrenamiento de modelos de
aprendizaje automático supervisado
library(caret)
set.seed(123)#establecemos una semilla inicial para la generación de números
aleatorios
training_sample1<-data_coef$HO_vl%>%
  #se utiliza para dividir un conjunto de datos en un conjunto de entrenamiento y un
conjunto de prueba de manera estratificada y aleatoria
  createDataPartition(p=0.8,list=F)
#dividimos el conjunto de datos data_coef
training_base1<-data_coef[training_sample1,]
test_base1<-data_coef[-training_sample1,]
```

```

#ajustamos un modelo de regresión lineal con variable de salida hand_over
modH_O<-lm(HO_vl~.,data=training_base1)
modH_O

summary(modH_O)

##Metricas de evaluacion del modelo
library(Metrics)#libreria con funciones para evaluar y comparar modelos de aprendizaje
automático.

#se aplica la funcion predic al modelo utilizando los datos de prueba almacenados en
test_base1
predic_1<-modH_O%>%
  predict(test_base1)
#El vector de predicciones almacenado en "predic_1" se convierte en un data frame de
una sola columna
predic1_df<-data.frame(predic_1)
#Se imprimen las predicciones por consola
predic_1
predic1_df
#se establece la semilla inicial para la generación de números aleatorio
help(set.seed)

#cálculo del error cuadrático medio (RMSE) entre las predicciones del modelo y los
valores observados de la variable de respuesta
rmse(predic_1,test_base1$HO_vl)

#calculo el coeficiente de determinación (R2) entre las predicciones del modelo y los
valores observados de la variable de respuesta
R2(predic_1,test_base1$HO_vl)

```

Anexo IV

```
%Codificación para automatizar el proceso de prueba la ecuación obtenida
clear all, close all, clc
%Cargamos la base de datos
Base_datos=xlsread("Sesion_R1_Completa_Depurada_Valores.xlsx");
%%
%Extraemos las variables de salida con las que esta implementado el modelo
rssi_stronges=Base_datos(:,9);%Extraemos la columna 9 perteneciente a rssi_stronges
de la base de datos
rssi=Base_datos(:,13);%Extraemos la columna 13 perteneciente a rssi de la base de
datos
rsrq=Base_datos(:,14);%Extraemos la columna 14 perteneciente a rsrq de la base de
datos

%Extraemos la variable de salida
HO_vl=Base_datos(:,26);%Extraemos la columna 26 perteneciente a hand_over de la
base de datos

%A continuacion se establece un lazo para recorrer los datos
for i=1: length(HO_vl)
%Ecuacion de Regresion Lineal
Ec_LinealRegresion(i,:)= 0.15017 + 0.03438*rssi_stronges(i,:) - 0.03899*rssi(i,:) +
0.02894*rsrq(i,:);
end
%Redondeamos los valores obtenidos
salida=round(Ec_LinealRegresion);

%% Comparacion de resultados
comparacion= HO_vl==salida;%Condicion que arroja un estado logico de verdad o falso
valor_erroneos=sum(comparacion==0);%Operador que indica los valores errados
valor_buenos=sum(comparacion==1);%Operador que indica los valores buenos

tamano=length(Base_datos);%Determinacion del tamaño de la base de datos
Porcentajes_valor_prediccion_buena=valor_buenos/tamano*100 %Calculo del
porcentaje de valores bien predichos
```

```
Porcentajes_valor_prediccion_mala=valor_erroneos/tamano*100    %Calculo    del
porcentaje de valores mal predichos
```

Anexo V

```
%Codificacion para automatizar el proceso de prueba la ecuacion obtenida
clear all, close all, clc
%Cargamos la base de datos
Base_datos=xlswread( "Sesion_R2_Completa_Depurada_Valores.xlsx");
%%
%Extraemos las variables de salida con las que esta implementado el modelo
rss_i_stronges=Base_datos(:,8);%Extraemos la columna 9 perteneciente a rss_i_stronges
de la base de datos
rss_i=Base_datos(:,12);%Extraemos la columna 13 perteneciente a rss_i de la base de
datos
% rsrq=Base_datos(:,14);%Extraemos la columna 14 perteneciente a rsrq de la base
de datos

%Extraemos la variable de salida
HO_vl=Base_datos(:,25);%Extraemos la columna 26 perteneciente a hand_over de la
base de datos

%A continuacion se establece un lazo para recorrer los datos
for i=1: length(HO_vl)
%Ecuacion de Regresion Lineal
Ec_LinealRegresion(i,:)= -0.2560609 + 0.0002748*rss_i_stronges(i,:) +
0.0024771*rss_i(i,:);
end
%Redondeamos los valores obtenidos
salida=ceil(Ec_LinealRegresion);

%% Comparacion de resultados
comparacion= HO_vl==salida;%Condicion que arroja un estado logico de verdad o
flaso
valor_erroneos=sum(comparacion==0);%Operador que indica los valores errados
valor_buenos=sum(comparacion==1);%Operador que indica los valores buenos

tamano=length(Base_datos);%Determinacion del tamaño de la base de datos
Porcentajes_valor_prediccion_buena=valor_buenos/tamano*100 %Calculo del
porcentaje de valores bien predichos
Porcentajes_valor_prediccion_mala=valor_erroneos/tamano*100 %Calculo del
porcentaje de valores mal predichos
```

ANEXO VI

```
%Análisis de la celda objetivo
clear all, close all, clc
%Cargamos la base de datos
Base_datos=xlsread( "Sesion_R1_Completa_Depurada_Valores.xlsx");

%Definimos los identificadores de nodo en un vector de Strings
Identificadores_celda=["eNB28834" "eNB28235" "eNB28542" "eNB28875" "eNB28059"
"eNB28387" "eNB28322" "eNB28542" "eNB28334" "eNB28010" "eNB28200"
"eNB28030"];

%Extraemos las distancias de la estación móvil a las estaciones base
distancias=Base_datos(:,33:44);
tamano_bd=length(distancias);

%Selecciona la distancia menor por fila
for i=1:tamano_bd
    dismin(i,:)=min(distancias(i,:));
end
%Aignamos un valor lógico de 1 en las columnas que se tengan la menor distancia a la
estación base
for i=1:tamano_bd
    for j=1:12
        matriz_eNb_logicos(i,j)= dismin(i)==distancias(i,j);
    end
end

%Encuentra la posición en la matriz de lógica matriz_eNb_logicos
for i=1:tamano_bd
    posicion(i)= find(matriz_eNb_logicos(i,:)==1);
end

%Se asigna la celda candidata más cercana
for i=1:tamano_bd
    celda_seleccioanda(i)= Identificadores_celda(posicion(i));
end
celda_seleccioanda=celda_seleccioanda!;
```