

**Análisis y simulación de un sistema para transmitir la calidad del aire utilizando internet de las cosas de banda estrecha**

Samuel Alejandro Clavijo Hernández

Universidad Nacional Abierta y a Distancia - UNAD  
Escuela de Ciencias Básicas Tecnología e Ingeniería – ECBTI  
Ingeniería de telecomunicaciones  
2022

**Análisis y simulación de un sistema para transmitir la calidad del aire utilizando internet de las cosas de banda estrecha**

Samuel Alejandro Clavijo Hernández

Trabajo de grado presentado como requisito para optar al título de  
Ingeniero en Telecomunicaciones

Directora  
Dra. Mónica Andrea Rico Martínez

Universidad Nacional Abierta y a Distancia - UNAD  
Escuela de Ciencias Básicas Tecnología e Ingeniería - ECBTI  
Ingeniería de Telecomunicaciones  
2022

### **Dedicatoria**

A Dios por regalarme la vida, la fuerza, salud, sabiduría y permitir mi superación profesional al elaborar este proyecto de grado, que tomo tiempo y sacrificio.

A mis padres quienes me han apoyado en cada paso que doy.

A mis hermanas quienes en silencio creen mucho en mí.

## **Agradecimientos**

Agradezco a la Dra. Mónica Andrea Rico Martínez quien fue mi tutora en una materia de la universidad y posteriormente se convirtió en mi directora de grado, por su compromiso, tiempo, paciencia y profesionalismo, aprendí mucho de ella, por otro lado, mi agradecimiento al Ingeniero Luis Felipe Ariza Vesga de la comunidad Sucre investiga, quien desinteresadamente me apoyó con su tiempo y conocimiento, sin ellos no hubiese sido posible este proyecto

## Resumen

El despliegue de las redes 4G y 5G, acompañado de la disrupción tecnología de la industria 4.0 e internet de las cosas, han permitido la implementación de diversos casos de uso y aplicaciones para evaluar su funcionamiento o valorar su eficacia. De la mano de estudios de investigación referentes a redes móviles de última generación, estudios de análisis de calidad del aire en Bogotá, contaminación atmosférica, estudio de emisión de material particulado PM10 y PM2.5, prototipos para la medición de partículas, entre otros, se disponen las bases teóricas y conceptuales para dar solución al problema de investigación planteado en este trabajo.

Este proyecto simula un sistema para transmitir datos de la calidad del aire en Bogotá, a través de herramientas de tecnológicas que permiten analizar la transmisión. La simulación del sistema se realiza teniendo en cuenta la arquitectura de red móvil instalada en la capital de Colombia. El alcance de este proyecto es simular y emular una red de internet de las cosas de banda estrecha, en la cual intervenga la implementación de un core, una red de acceso, un usuario final y por supuesto un conjunto de datos que se puedan transmitir para evaluar su desempeño con indicadores de calidad.

Palabras claves. Banda estrecha, 5G, industria 4.0, internet de las cosas, simulación, emulación, material particulado, calidad del aire, red móvil,

## **Abstrac**

The deployment of 4G and 5G networks, accompanied by the technological disruption of industry 4.0 and the Internet of Things, have allowed the implementation of various use cases and applications to evaluate their operation or assess their effectiveness. Hand in hand with research studies related to state-of-the-art mobile networks, air quality analysis studies in Bogotá, air pollution, PM10 and PM2.5 particulate matter emission studies, prototypes for particle measurement, among others, The theoretical and conceptual bases are arranged to solve the research problem raised in this work.

This project simulates a system to transmit air quality data in Bogotá, through technological tools that allow the transmission to be analyzed. The simulation of the system is carried out taking into account the mobile network architecture installed in the capital of Colombia. The scope of this project is to simulate and emulate a narrowband internet of things network, in which the implementation of a core, an access network, an end user and of course a set of data that can be transmitted to evaluate their performance with quality indicators.

Keywords. Narrow band, 5G, industry 4.0, internet of things, simulation, emulation, particulate matter, air quality, mobile network,

## Tabla de Contenido

Introducción .....	14
Generalidades .....	16
Problema de Investigación .....	16
Objetivos de la Investigación.....	19
Objetivo General.....	19
Objetivos Específicos .....	19
Justificación.....	19
Marco Referencial .....	21
<i>Marco Conceptual</i> .....	21
Internet of Things (IoT).....	21
5G o New Radio.....	22
Radio Definida por Software. ....	23
GNU Radio.....	24
OpenAirInterface Software Alliance. E .....	24
Estado Del Arte .....	25
Análisis de Inmisión de Material Particulado PM10 y PM2.5 en Dos Puntos Estratégicos de Bogotá.....	25
Niveles de Partículas Suspendidas Totales (PST), PM10 y PM2.5 y su Relación en Lugares Públicos de la Ciudad Riohacha, Caribe Colombiano. ....	25
Desarrollo de un Prototipo Para la Medición de Partículas PM2.5 en Bogotá.....	26

Relación entre PM2,5 y PM10 en Bogotá.....	27
Prototipo IoT para la Medición de la Polución Atmosférica en la Universidad Católica de Colombia. ....	27
Prototipo Basado en Internet de las Cosas Para el Monitoreo Ambiental Como Contribución a la Orientación de Bucaramanga (Colombia) Hacia Smart City. ....	28
Transferencia de Datos a Través del Transceptor OFDM con la Ayuda de GNU Radio. ....	29
Estudio de Plataformas SDR para LTE-5G. ....	29
Metodología .....	30
Etapa 1: Diseño del Sistema.....	30
Etapa 2: Implementación .....	31
Etapa 3. Pruebas al Sistema .....	31
Etapa 4. Redacción Documento .....	31
Elementos del Sistema de Comunicaciones .....	32
Redes 5G.....	32
IoT de Banda Estrecha.....	36
Banda Estrecha .....	36
Contexto para la Implementación de Una Red para Transportar los Datos de la Calidad del Aire .....	37
Proyecto de Red de Acceso de Radio OpenAirInterface 5G .....	37
Proyecto RED CORE 5G OpenAirInterface .....	38
Grupo de Proyecto OpenAirInterface MOSAIC5G .....	39

Software a Implementar en el Proyecto.....	39
GNURadio .....	39
OpenAirInterface Software Alliance .....	40
Sensores de Calidad del Aire .....	41
PMS5003 Digital Universal Particle Concentration Sensor .....	41
HK-A5 Laser PM2.5 - 10 Sensor.....	42
Dataset.....	43
Dataset del Sensor HK-A5 Laser y Sensor PMS5003.....	44
Sensores de la Red de Monitoreo de Calidad del Aire RMCAB .....	46
Diseño del Sistema para Transmitir Información Referente a la Calidad del Aire en Bogotá Empleando IoT de Banda Estrecha y Redes Móviles .....	50
Diseño de un Sistema de Transmisión OFDM.....	50
Opciones y Variables Simuladas .....	51
Codificación de los Datos y Transmisión de Paquetes.....	52
Etapa de Multiplexación.....	53
Etapa de División en Frecuencia Ortogonal y Prefijo Cíclico.....	54
Etapa de Visualización de la Señal .....	55
Etapa de Recepción .....	56
Diseño del Sistema de Transmisión NB-IoT Orientado a Redes 5G.....	57
Diseño de un Sistema de Transmisión entre la Tarjeta Madre GSM EVB KIT y el Módulo de Internet de las Cosas de Banda Estrecha BC95-G de QUECTEL.....	58
Instalación de Drivers en el Ordenador.....	60

Resultados de las Simulaciones y Configuración de la Red Móvil.....	63
Configuración General de la Red Móvil .....	63
Configuración en OpenAirInterface del CORE y RED DE ACCESO para Transmisión de Datos en la Red Móvil .....	64
Paso a Paso Para Configurar y Lanzar el Core de la Red .....	65
Crear una Red Virtual. ....	66
Crear los Contenedores. ....	67
Verificar la Conexión de los Contenedores o Entidades. ....	67
Utilizar la Base de Datos Cassandra. ....	68
Configuración del HSS. ....	69
Verificar Datos de Cassandra.....	69
Configuración del MME, SPGWC y SPGW-U. ....	70
Lanzamiento de las Entidades del EPC. ....	72
Lanzamiento del eNodeB Utilizando OpenAirInterface y la USRP B200 Mini. ....	74
Lanzar el eNodeB en OpenAirInterface.....	75
Ejecución de eNodeB.....	76
Simulación del Usuario Final en GNURADIO .....	79
Configuración de bloques para indicadores de calidad .....	81
Simulaciones y recopilación de datos .....	82
Conclusiones .....	90
Referencias.....	92

### Lista de tablas

<b>Tabla 1.</b> Datos que se obtienen mediante sensor HK-A5 Laser y sensor PMS5003 ...	44
<b>Tabla 2.</b> Resultados de las simulaciones en GNU radio del Throughput.....	84
<b>Tabla 3.</b> Resultados de las simulaciones en GNU radio del BER (Bit Error Rate).....	87

## Lista de Figuras

<b>Figura 1.</b>	Evolución de las generaciones de las redes móviles .....	33
<b>Figura 2.</b>	Esquema 5G NSA y 5G SA .....	34
<b>Figura 3.</b>	Comparación arquitectura 4G y 5G. ....	38
<b>Figura 4.</b>	Diagrama de bloques partes del sensor PMS5003 .....	42
<b>Figura 5.</b>	Imagen sensor HK-A5 Laser PM2.5 PM10 .....	43
<b>Figura 6.</b>	Mapa de las 20 estaciones de monitoreo RMCAB .....	46
<b>Figura 7.</b>	Información estaciones de monitoreo de Puente Aranda, Suba, El Tunal y Usaquén del 24 de septiembre de 2021 .....	47
<b>Figura 8.</b>	Definición bloque de opciones y variables .....	52
<b>Figura 9.</b>	Etapas de mapeo de los datos en paquetes .....	53
<b>Figura 10.</b>	Etapas de procesamiento y etiquetado de los datos.....	54
<b>Figura 11.</b>	Etapas de parametrización de OFDM y FFT .....	55
<b>Figura 12.</b>	Etapas de diagramación de la señal .....	56
<b>Figura 13.</b>	Etapas de demodulación de la señal OFDM.....	57
<b>Figura 14.</b>	Esquema red de acceso y del core con sus respectivas entidades.....	58
<b>Figura 15.</b>	Orden de conexión de las tarjetas QUECTEL.....	59
<b>Figura 16.</b>	Comunicación serial por comandos AT utilizando QCOM_V1.0 .....	61
<b>Figura 17.</b>	GSM EVB KIT y módulo BC95-G NB IoT de QUECTEL .....	62
<b>Figura 18.</b>	Conexión física de los equipos del core y eNodeB .....	64
<b>Figura 19.</b>	Clonar repositorio con los elementos que componen el núcleo de red...	65
<b>Figura 20.</b>	Acceder a la carpeta clonada y verificar versión .....	66
<b>Figura 21.</b>	crear la red de trabajo.....	67

<b>Figura 22.</b>	Ejecutar los contenedores en Docker .....	67
<b>Figura 23.</b>	verificar la conexión del MME .....	68
<b>Figura 24.</b>	Acceder a la base de datos Cassandra .....	68
<b>Figura 25.</b>	Ejecutar el HSS (Home Subscriber Server) .....	69
<b>Figura 26.</b>	Verificar base de datos Cassandra .....	70
<b>Figura 27.</b>	Configuración del MME .....	71
<b>Figura 28.</b>	Configuración del SPGWC .....	72
<b>Figura 29.</b>	Configuración del SPGW-U .....	72
<b>Figura 30.</b>	MME en ejecución .....	73
<b>Figura 31.</b>	Todo el CORE ejecución .....	74
<b>Figura 32.</b>	USRP B200mini.....	75
<b>Figura 33.</b>	Acceder a repositorio openairinterface5g.git .....	75
<b>Figura 34.</b>	Editar archivo de configuración.....	76
<b>Figura 35.</b>	Ejecución de eNodeB .....	77
<b>Figura 36.</b>	Estadística de conexión en el MME .....	77
<b>Figura 37.</b>	Paquete de 9216 bytes listo para transmitir .....	80
<b>Figura 38.</b>	Parámetros de transmisión en GNU radio para OFDM .....	81
<b>Figura 39.</b>	Bloques para indicadores de calidad throughput y BER .....	82
<b>Figura 40.</b>	Simulación del paquete enviado con GNU radio.....	82
<b>Figura 41.</b>	Archivo recibido por OFMD en GNU radio .....	83
<b>Figura 42.</b>	Grafica indicador de Throughput.....	86
<b>Figura 43.</b>	Grafica Bit Error Rate Fuente: autor .....	89

## Introducción

El creciente deterioro de la calidad del aire en Bogotá ha sido identificado por las autoridades como un problema de salud pública, a pesar de las estrategias implementadas para mitigarlo, la ciudad está en un acelerado crecimiento poblacional y económico, lo que involucra mayor demanda de energía e incremento en el consumo de combustibles fósiles. En Bogotá se han realizado múltiples estudios, análisis y observaciones para monitorear la calidad del aire, actualmente la secretaria distrital de ambiente cuenta con la Red de Monitoreo de Calidad del Aire de Bogotá – RMCAB la red consta de 13 estaciones fijas de monitoreo y una móvil. Por consiguiente, la finalidad de este proyecto es simular un sistema para transportar todos los datos en tiempo real de la calidad del aire medida en Bogotá utilizando Internet de las cosas de banda estrecha orientado a redes 5G.

Para tal fin, el presente documento se encuentra elaborado en 6 capítulos estructurados de la siguiente forma: Capítulo 1. Generalidades, donde se plasma el planteamiento del problema, justificación, objetivos, metodología, marco conceptual y teórico, impacto del proyecto, seguido a este se encuentra el capítulo 2, donde se describen los elementos del sistema de comunicaciones, en éste capítulo se profundiza en la descripción del software que se utilizó en el proyecto, el dataset a transmitir, la explicación concreta de la configuración 5G y una amplia descripción de IoT de banda estrecha, posteriormente en el capítulo 3. Diseño del sistema para transmitir información referente a la calidad del aire en Bogotá empleando IoT de banda estrecha orientado a redes 5G, en este capítulo se grafica el diagrama de bloques, se realiza el diseño de las pruebas de desempeño de la red y se presentan los elementos

seleccionados para la simulación de prototipo. Luego en el capítulo 4, se encontrará la simulación del prototipo, se ejecutan las simulaciones en GNU radio y en OpenAirInterface. Después en el capítulo 5 se encuentra un análisis de la simulación, se hicieron las pruebas pertinentes de calidad de la señal (SNR), pruebas de throughput y pruebas de probabilidad de error, y por último en el capítulo 6. Conclusiones y trabajos futuros, se consignan las deducciones del trabajo realizado, los logros más sobresalientes de la ejecución y los aspectos por mejorar que se presentaron a lo largo del proyecto, así mismo se deja como referencia los trabajos futuros que se pueden implementar con el proyecto.

## **Generalidades**

En este capítulo se presenta el planteamiento del problema, la justificación, los objetivos, el marco teórico, el marco conceptual y el enfoque metodológico de la investigación, así mismo el impacto de los resultados del proyecto aplicado.

### **Problema de Investigación**

En la actualidad las redes móviles se han convertido en una herramienta prácticamente indispensable para el desarrollo de la sociedad, la vigente quinta generación de esta tecnología puede ofrecer mayor cobertura en áreas metropolitanas, velocidades de datos de usuario del orden de Gbps en todas partes, latencia del orden de 1 milisegundo, 10 veces más la duración de la batería para los dispositivos, mejoras a la realidad virtual, posibilidad de experimentar llamadas holográficas, realizar transmisión de video en alta resolución, etc. (Agiwal, 2016). 5G es la transformación digital que hace realidad la industria 4.0 que se refiere a que todo está conectado a la red, ahorro y eficiencia energética, ciudades inteligentes, aplicaciones innovadoras y nuevos casos de uso que involucran IoT.

De la misma forma, el auge desmesurado de los servicios a través de los datos móviles, especialmente impulsado por el internet inalámbrico y la inmensa variedad de dispositivos inteligentes conectados a internet en donde resalta IoT, han llevado a iniciar investigaciones y desarrollos hacia las futuras tecnologías (Guevara Córdoba, 2019). Las características técnicas de 5G permiten pensar en un sin fin de nuevos escenarios de casos de uso, proporcionando oportunidades comerciales. Estos nuevos servicios tendrán un impacto positivo en el medio ambiente, la sociedad y las personas al reducir el tráfico, la congestión, optimizar el consumo de energía, lo que supone un

mejor uso de los recursos de red. Adicionalmente, en coherencia con este proyecto es posible informar a las personas sobre los niveles de contaminación en diversas zonas de una ciudad o lugar específico. Esto permite brindar información oportuna y personalizada a los ciudadanos ( Xiang , Zheng, & Shen, 2017).

Dicho lo anterior, este proyecto toma como problema la contaminación del aire, teniendo en cuenta que la contaminación atmosférica es el más relevante riesgo ambiental para la salud humana, pues el aire que se inhala a diario está colmado de agentes contaminantes como el material particulado (PM) con diámetro de hasta 10 micras (PM10), y el más letal para el ser humano es el que tiene un diámetro de 2.5 micras (PM2.5). Por su diminuto tamaño, estas partículas van directamente a los pulmones afectando a niños y adultos, provocando enfermedades respiratorias y cerebrovasculares, como asma, neumonía, cáncer de pulmón, entre otras (OPS, 2021).

Después de todo, la contaminación del aire es un tema que preocupa desde los defensores del medio ambiente hasta los políticos encargados de edificar las leyes que la regulan, es un tema que perturba a todo el planeta de forma sistemática, en el caso de los países latinoamericanos, a causa del fracaso de estrategias ambientales en empresas públicas y privada, el resultado se refleja en la afectación a la salud de la población en general. La Organización Mundial de la Salud establece el límite “sano” de material particulado en el aire en 20 microgramos por metro cuadrado o (mcg/m<sup>3</sup>), en el caso de las grandes ciudades colombianas se tiene en cuenta la presencia partículas de ozono y óxidos de carbono, azufre y nitrógeno. Particularmente Bogotá se ubica entre los 130 y 160 mcg/m<sup>3</sup> (Restrepo Mantilla, 2019).

De acuerdo con los datos de la Secretaría de Ambiente, el 21% de la contaminación de Bogotá, la aporta la industria y el 79% las fuentes móviles (vehículos particulares, buses, taxis, camionetas, el sistema integrado de transporte público). Los vehículos de transporte son los principales contaminantes del aire, y hoy en día, en Colombia se encuentran aproximadamente 13 millones de estos, de los cuales solo 1000 de ellos son vehículos eléctricos, situación que contribuye dramáticamente a altos los niveles de contaminación por partículas PM2.5 en el aire (Castiblanco, 2021). Según la Organización Mundial de la Salud (OMS), Colombia es el segundo país de Latinoamérica con más contaminación del aire, causando alrededor de 19.000 muertes anuales en todo el territorio de las cuales 3.8 millones de personas mueren prematuramente por enfermedades atribuibles a la contaminación del aire (Digital, 2018).

Teniendo en cuenta la importancia de conocer con exactitud las zonas con mayor contaminación en la ciudad con el fin de informar a la sociedad y tomar las medidas pertinentes para mitigar las consecuencias en la salud, en un tiempo eficaz, y considerando que IoT a través de 5G permite el envío de información en tiempo real a una escala que no es posible actualmente con otras tecnologías, se busca dar respuesta a la siguiente pregunta de investigación: ¿Cómo simular un sistema que transporte los datos medidos de la calidad del aire en Bogotá utilizando Internet de las cosas de banda estrecha orientado a redes 5G?

## **Objetivos de la Investigación**

### ***Objetivo General***

Simular un sistema para transmitir datos de la calidad del aire en Bogotá utilizando Internet de las cosas de banda estrecha orientado a redes 5G

### ***Objetivos Específicos***

Caracterizar los elementos del sistema que permitan transportar las mediciones de la calidad del aire en Bogotá

Diseñar el sistema para transmitir información referente a la calidad del aire en Bogotá empleando IoT de banda estrecha orientado a redes 5G.

Realizar las simulaciones del prototipo.

Ejecutar las pruebas de simulación, teniendo en cuenta los indicadores de desempeño de la red (Key Performance Indicators KPI -calidad de la señal (SNR), throughput, probabilidad de error)

### **Justificación**

En la actualidad un 54% o 4.000 millones de personas, más de la mitad de la población vive en zonas urbanas, en las últimas décadas se han identificado nuevas tendencias, la población humana ha migrado de lo rural a lo urbano, por lo que ha experimentado un crecimiento desmedido. Los cálculos estadísticos arrojan que para el año 2030 la cifra de personas que vivan en ciudades alcance los 5.000 millones y para 2050 se especula que lo hagan dos tercios de la especie humana (Gavidia, 2019), lo que resultará en un mayor consumo de energía, más residuos y más tráfico, por lo tanto, afectará negativamente la calidad del aire y aumentará la contaminación del aire en las ciudades. A medida que la calidad del aire urbano disminuye, aumenta el riesgo

de accidente cerebrovascular, enfermedades cardíacas, cáncer de pulmón y enfermedades respiratorias crónicas y agudas, incluido el asma, para las personas que viven en estas áreas hiperdensas.

En este contexto, y dado que el incremento del parque automotor en Bogotá, la circulación de vehículos diésel con más de 20 años de antigüedad, el humo producido por la industria y todos aquellos factores causantes de material particulado PM es importante empezar a crear un sistema que permita tomar medidas correctivas.

De acuerdo a un estudio llamado Normas de calidad del aire para la concentración de partículas PM<sub>2.5</sub>: análisis descriptivo global, el auto (Nazarenko, 2021) tomó datos de 58 países incluyó a Colombia y como resultado final expuso que a pesar del cumplimiento de normas relativamente estrictas, se deben continuar actualizando las reglas de medición ambiental de las partículas PM<sub>2.5</sub> y motiva a los gobiernos a no bajar la guardia con las tecnologías empleadas para el control de la contaminación del aire.

Conviene subrayar que respirar es un acto involuntario y necesario para la vida de los seres humanos y utilizar cada vez las mejores herramientas tecnológicas es un paso para garantizar una excelente calidad de vida, aun mejor en el contexto de pandemia que se encuentra el planeta. De esta manera, el hecho de realizar la simulación de un sistema para medir la calidad del aire en Bogotá utilizando Internet de las cosas de banda estrecha orientado a redes 5G, es un proyecto que sirve primero como punto de partida para construir un prototipo basado en la investigación y pruebas aquí realizadas, por otro lado, para contextualizar la problemática y recalcar que es prudente utilizar IoT y 5G como tecnología para este tipo de mediciones.

## **Marco Referencial**

Comprende el marco conceptual y teórico, se explica detalladamente la tecnología y tipos de redes que se utilizaron para la simulación del prototipo, de igual forma se referencian otros proyectos e investigaciones que involucran medición de material particulado en lugares públicos.

## ***Marco Conceptual***

**Internet of Things (IoT).** El Internet de las cosas (IoT) se refiere al uso de dispositivos y sistemas conectados de manera inteligente para aprovechar los datos recopilados por sensores y actuadores integrados en máquinas y otros objetos. Otra definición puede darse al descomponer la frase "Internet de las cosas", es decir, la primera palabra "Internet" y la segunda palabra "Cosas". Internet comprende un sistema global de redes informáticas interconectadas que utilizan el conjunto de protocolos de Internet estándar (TCP / IP) para servir a miles de millones de usuarios en todo el mundo. Es una red de redes que consta de millones de redes privadas, públicas, académicas, empresariales y gubernamentales, de alcance local a global, que están conectadas por una amplia gama de tecnologías de redes electrónicas, inalámbricas y ópticas (Mircea, Stoica, & Bogdan, 2021). Por otro lado, la palabra cosas hace referencia a todas las herramientas, dispositivos eléctricos y electrónicos que se pueden conectar a una red de área local.

IoT hace parte de la nueva convergencia y las redes de nueva generación, la implementación de esta tecnología dará rienda suelta a una nueva dimensión de servicios que mejoran la calidad de vida de los consumidores y productividad de las empresas.

**5G o New Radio.** El término 5G, es utilizado para referirse a la quinta generación de telefonía móvil. 5G necesita una evolución a largo plazo de (LTE) que incorpora 4G, el cual hasta ahora se ha implementado y está alcanzando la madurez, donde solo las mejoras incrementales y pequeñas cantidades de nuevo espectro. De manera general, se pueden mencionar tres características relevantes de la tecnología 5G; la ultra densificación, mmWave y el uso del uso masivo de múltiples entradas, múltiples salidas (MIMO) (Andrews, y otros, 2014).

Es importante tener en claro que 5G no sólo será una mejora incremental sobre sus antecesores; sino que pretende ser un salto revolucionario en términos de velocidad de transmisión de datos, latencia, conectividad masiva, confiabilidad de la red y eficiencia energética (Shafi, y otros, 2017). Estas capacidades están destinadas a realizar conectividad de alta velocidad, internet de las cosas, realidad virtual y aumentada, Internet táctil, etc. Los requisitos de 5G se espera se cumplan, utilizando grandes anchos de banda disponibles en bandas de ondas milimétricas, aumentando los grados de libertad espacial a través de grandes matrices de antenas y MIMO 3D, densificación de red y nuevas formas de onda que proporcionan escalabilidad y flexibilidad para cumplir con diferentes demandas de los servicios 5G.

Se espera que 5G sea un gran habilitador de la cuarta revolución industrial, dados los requisitos de su implementación, además será la base para desarrollar un escenario de conexión IoT masiva, que proporcione aplicaciones con experiencias en tiempo real, bajo demanda, todas en línea, reconfigurables y sociales, lo que requiere la arquitectura.

La arquitectura 5G-IoT debe poder coordinarse de un extremo a extremo, mediante un funcionamiento que sea ágil, flexible, automático e inteligente. 5G deberá propiciar una arquitectura de interconexión con sistemas IoT que proporcione redes lógicamente independientes de acuerdo con los requisitos de aplicaciones. Se espera que las redes 5G proporcionen: (1) banda ancha móvil mejorada (eMBB), (2) ultra fiable y de baja latencia comunicaciones (uRLLC) y (3) comunicaciones masivas de tipo máquina (mMTC). Es importante tener claro que una red de telefonía celular consta principalmente de dos subsistemas: la Red de Acceso Radio (RAN) y el Núcleo Móvil (Peterson & Sunay, 2020) .

**Radio Definida por Software.** La radio definida por software (SDR) fue diseñada por Joseph Mitola, a principios de los 90 como un identificador de una clase de radios que podían reprogramarse y reconfigurarse mediante software. Mitola diseñó una radio definida por software ideal, cuyos componentes físicos eran solo una antena y un convertidor analógico digital (ADC) en el lado del receptor. Asimismo, el transmisor tendría un convertidor analógico digital (DAC) y una antena transmisora. El resto de las funciones serían manejadas por procesadores reprogramables. Esta idea aún está desarrollándose en diferentes laboratorios de investigación en el mundo.

Actualmente, el término SDR se utiliza para describir un dispositivo empleado en comunicaciones inalámbricas, que se define principalmente por software, pero que incluye importantes componentes de hardware. Tanto los transmisores como los receptores SDR son bastante diferente de los tradicionales. En esta tecnología, una vez que el dispositivo SDR está en comunicación con el PC, se puede comenzar a buscar usos de la tecnología, ofreciendo soluciones específicas. El concepto de

plataforma unificada y la capacidad de corregir errores en tiempo real son las aplicaciones clásicas de SDR. Sin embargo, los estudios han identificado otras aplicaciones importantes, tales como: posicionamiento dinámico del espectro, acceso múltiple basado en oportunidades (ODMA), regulación del espectro y reducción de costos (algunas implementaciones de SDR son más baratas que su contraparte analógica) (Machado Fernández, 2014).

**GNU Radio.** GNU Radio es un kit de herramientas de desarrollo de software de código abierto y gratuito que proporciona bloques de procesamiento de señales para implementar radios de software. Se puede utilizar con hardware de RF externo de bajo costo y fácilmente disponible para crear radios definidas por software, o sin hardware en un entorno similar a una simulación. Se utiliza ampliamente en entornos de investigación, industria, academia, gobierno y aficionados para respaldar tanto la investigación de comunicaciones inalámbricas como los sistemas de radio del mundo real (GNURadio, 2021).

**OpenAirInterface Software Alliance.** Es una organización francesa sin ánimo de lucro creada en 2014, esta alianza es la responsable del desarrollo, control de calidad y promoción del software a nivel institucional en todo el planeta. La OSA es el hogar de OpenAirInterface, un software abierto que reúne a una comunidad de desarrolladores de todo el mundo, que trabajan juntos para construir tecnologías de Red de Acceso de Radio (RAN) celular inalámbrica y Red Central (CN) (OAI21). Para implementar el uso de OSA no se requiere ningún hardware de radio ya que puede simular las interfaces necesarias para la tecnología 5G.

### ***Estado Del Arte***

Se relacionan investigaciones realizadas por otros autores y que tiene relación con la temática del proyecto, aportando a la estructura teórica los resultados pertinentes para avanzar con el objeto de la investigación.

**Análisis de Inmisión de Material Particulado PM10 y PM2.5 en Dos Puntos Estratégicos de Bogotá.** En el 2018 se realizó un análisis de material particulado PM10 y PM2.5, se usaron equipos de campo HI-VOL PM10 VFC, TE-WILBUR), PQ-200 y URG-3000ABC, los primeros tres utilizados en la localidad de Puente Aranda y el URG-3000ABC emplazado en la Sevillana por el gran flujo vehicular. Para la fase de muestreo en campo usaron neveras para el transporte de filtros y geles refrigerantes para mantener la cadena de frío y toma de datos de inicio y finalización del muestreo (Pachon Quinche, 2018). En el estudio se utilizaron equipos manuales que requerían un gran andamiaje administrativo para el manejo, monitoreo y toma de muestras, lo que requirió capacitación y mayor presupuesto para su ejecución, por otro lado, el área de cobertura de estudio en tan solo dos puntos no garantiza una fiabilidad para tomar decisiones ya que el área metropolitana tiene más puntos críticos que requieren ser tenidos en cuenta.

**Niveles de Partículas Suspendidas Totales (PST), PM10 y PM2.5 y su Relación en Lugares Públicos de la Ciudad Riohacha, Caribe Colombiano.** Un trabajo mancomunado de la facultad de Ingeniería de la Universidad de La Guajira y la facultad de Ingeniería de la Universidad de Antioquia, realizó un estudio para determinar la concentración de partículas menores de 10 y 2.5 micrómetros con

instrumentos de muestreo de alto volumen, mediante seis estaciones localizadas en universidades y corporaciones gubernamentales.

De acuerdo a los estudios realizados, en los 6 sitios de muestreo, los resultados de las concentraciones de PST, PM10 y PM2.5, no violaban los estándares para un periodo de muestreo de 24 horas de la norma colombiana y la National Ambient Air Quality Standards (NAAQS) de los Estados Unidos (Rojano, Angulo, & Restrepo, 2012). Teniendo en cuenta que Riohacha no es una ciudad con sobrepoblación y en expansión como es el caso de Bogotá o Medellín, según los resultados de los niveles de concentración de PST, PM10 y PM2.5 fueron bajas por lo que no se requirió tomar medidas como pico y placa vehicular.

#### **Desarrollo de un Prototipo Para la Medición de Partículas PM2.5 en Bogotá.**

La autora Wendy Dayane Martínez Villalobos de la facultad de ingeniería de la universidad católica de Colombia mediante un trabajo de grado, el año pasado desarrolló un prototipo de bajo costo para medir partículas PM2.5 teniendo en cuenta que los sensores que actualmente se encuentran en el mercado tienen un costo elevado y poco asequible. El prototipo se fabricó basado en el sensor GP2Y1010AU0F, que permite medir la contaminación de partículas PM2.5. El proyecto fue concebido como una herramienta que permite informar a las personas de los niveles de contaminación a los que está expuesto, así mismo conozca las consecuencias de las partículas PM2.5 en su salud, todo a través del diseño y visualización de los datos censados por el prototipo desarrollado (Martínez Villalobos, 2020).

**Relación entre PM<sub>2,5</sub> y PM<sub>10</sub> en Bogotá.** Los autores Boris Galvis y Néstor Y. Rojas de la Universidad de los Andes y la Universidad Nacional de Colombia realizaron en Bogotá mediciones paralelas de material particulado menor a 2,5 micra (PM<sub>2,5</sub>) y menor a 10 micras (PM<sub>10</sub>), durante un periodo aproximado de dos meses. La relación entre (PM<sub>2,5</sub>) y (PM<sub>10</sub>), ha sido demostrada por diversos estudios realizados en ciudades latinoamericanas, en este caso se usaron los equipos BAMs Dasibi 7001 para monitorear PM<sub>10</sub> y los Metone-BAM1020 para las partículas PM<sub>2,5</sub>. Una de las conclusiones de los autores fueron que “Los datos de PM<sub>2,5</sub> no son predecibles a partir de datos de PM<sub>10</sub> en las estaciones Sony y Bosque. Sería necesario monitorear PM<sub>2,5</sub> de manera directa y estudiar con detalle las interacciones entre las fuentes y las variables meteorológicas alrededor de estas estaciones. En estas estaciones la relación PM<sub>2,5</sub>/PM<sub>10</sub> demuestra una mayor influencia de fuentes de material particulado grueso. En los sitios de monitoreo localizados en la localidad de Puente Aranda, al rededor del 15% de la masa de material particulado puede estar compuesta por carbón, nitratos y sulfatos provenientes de fuentes de combustión, formados en la atmósfera muy probablemente por procesos de nucleación y condensación” (Boris Galvis & Rojas, 2006).

**Prototipo IoT para la Medición de la Polución Atmosférica en la Universidad Católica de Colombia.** Este trabajo de grado realizado por los autores Gonzales Manrique y Rodríguez Chisco de la Universidad Católica de Colombia en el año 2020, propone la creación de un prototipo para medir gases contaminantes para la atmosfera (Monóxido de carbono (CO), dióxido de nitrógeno (NO<sub>2</sub>) y ozono (O<sub>3</sub>)) por medio de sensores electroquímicos capaces de transmitir información en tiempo real

por intermedio de un aplicativo. Para el envío de datos se testearon múltiples protocolos de comunicación entre los dispositivos y la red de datos con el fin de elegir el más apropiado y que se ajustara a las necesidades de los mismos. Para este proyecto se usaron dos dispositivos electrónicos cada uno con tres sensores y se desarrolló un aplicativo web para monitoreo de los datos. Los autores concluyen en el trabajo que: “Uno de los inconvenientes de la tecnología LoRa, es el sacrificio de velocidad de transmisión de datos, a cambio de bajo consumo y mayor alcance. Para este proyecto, ese percance fue sopesado usando una sinergia de tecnologías, LoRa para la transmisión de datos a largas distancias, y WiFi para establecer la conexión con internet. Por lo evidenciado en este trabajo, el uso de redes LoRa para transmisión de datos a internet, está todavía en desarrollo ya que sus protocolos de internet LoRa WAN no se encuentran acoplados con el modelo OSI, por lo cual se recomienda utilizar este tipo de tecnologías en conjunto con otras inalámbricas que sean robustas en el uso de sus protocolos de Internet.” (González Manrique & Rodríguez Chisco, 2020).

**Prototipo Basado en Internet de las Cosas Para el Monitoreo Ambiental Como Contribución a la Orientación de Bucaramanga (Colombia) Hacia Smart City.** Este trabajo de grado de la facultad de ingeniería de la universidad autónoma de Bucaramanga UNAB, ejecutado por los autores Velazco Ortiz y Pinzón Torres, tuvo como objetivo Implementar un prototipo basado en Internet de las cosas para enfrentar problemas de contaminación atmosférica en Bucaramanga y contribuir al desarrollo de la ciudad. El objetivo fue diseñar un prototipo tecnología de IoT de bajo costo, al final del proyecto se logró fabricar el prototipo con un NodeMCU, un sensor de gas MQ135 y

Cables Jumper (Hembra – Hembra), con alimentación de 5V de corriente directa (Velazco Ortiz & Pinzon Torres , 2018).

**Transferencia de Datos a Través del Transceptor OFDM con la Ayuda de GNU Radio.** Este estudio publicado en el 2021 International Conference on Computer Communication and Informática (ICCCI), tiene como objetivo construir un sistema de comunicaciones sin ruido utilizando la multiplexación por división de frecuencia ortogonal (OFDM) de un punto a otro, simulado mediante el software GNU radio. De este trabajo los autores concluyeron al final que al tratar de al tratar de transmitir archivos de texto y de audio, el audio para efectos de calidad fue tratado en la frecuencia 44,1 kHz, por otro lado, para transmitir datos se utilizó la técnica de multiplexación OFDM la cual al final del estudio se dedujo es la mejor solución para una comunicación de alta velocidad. En el cuerpo del trabajo se pueden encontrar graficas interesantes como los diagramas de bloques generados en el software GNU radio con todo el esquema de comunicación desde el transmisor hasta el receptor, tanto para los archivos de audio como para los archivos de texto (L, K, Savarinathan, & Jayapalan, 2021).

**Estudio de Plataformas SDR para LTE-5G.** En el presente trabajo de grado el autor Xueyan Xiang de la Universidad de Cantabria, pretende simular mediante el software OpenAirInterface (OAI) una red LTE utilizando máquinas virtuales para el eNB y por otro lado utilizar un User Equipment UE, para conectarse a la estación base creada. Así mismo en una segunda fase de este trabajo se realiza el análisis de los resultados obtenidos mediante la simulación. Para la implementación del escenario virtualizado se utiliza la tarjeta de desarrollo bladeRF, la cual funcionó como estación

base de una red LTE con resultados favorables, según el autor pudo conectar un UE a la red virtualizada y reproducir videos a una velocidad optima sin problemas.

Adicionalmente el autor manifiesta en las conclusiones que la plataforma OAI es una excelente opción para trabajos de investigación de redes móviles, sin embargo, también presenta debilidades, como la compatibilidad con algunos UE, inestabilidad en algunas secciones y complejidad en los procesos de ejecución del software ( Xiang , Zheng, & Shen, 2017).

### **Metodología**

El desarrollo del proyecto se llevó a cabo mediante cinco fases, enmarcadas en la metodología PHVA (Planear, Hacer, Verificar y Actuar), seguidas de un proceso de monitoreo y control que tenía como meta la mejora continua de los procesos del proyecto. La metodología PHVA es una técnica interactiva de resolución de problemas de cuatro pasos que se utiliza para mejorar los procesos.

En la fase de Planear, se establecieron los objetivos y procesos necesarios para lograr los resultados del proyecto.

En esta fase se desarrollaron las siguientes actividades:

- ✓ Planeación de los objetivos del proyecto
- ✓ Formulación del plan de trabajo y cronograma.

En la fase de Hacer o Implementar los procesos, se idearon las acciones pertinentes para dar cumplimiento a los objetivos planteados en la fase anterior. En esta fase se desarrollaron las siguientes etapas:

#### ***Etapas 1: Diseño del Sistema***

- ✓ Caracterizar los elementos del sistema

### ***Etapa 2: Implementación***

- ✓ Diseñar el sistema para transmitir información referente a la calidad del aire en Bogotá empleando IoT de banda estrecha orientado a redes 5G.
- ✓ Construir el sistema diseñado mediante las herramientas de simulación seleccionadas
- ✓ Realizar las simulaciones del prototipo.

En la fase de verificación, se monitorearon y se midieron los procesos y productos (resultados) respecto de los objetivos y requisitos. En esta fase se plantaron las siguientes etapas.

### ***Etapa 3. Pruebas al Sistema***

- ✓ Ejecutar las pruebas de simulación, teniendo en cuenta los KPI (calidad de la señal (SNR), throughput, probabilidad de error).

### ***Etapa 4. Redacción Documento***

- ✓ Redacción de cada etapa del desarrollo del proyecto (estado del arte, definición de requerimientos del diseño del sistema de comunicaciones, así como su diseño, implementación y validación).

En la fase de Actuar, se evaluó el proyecto y se establecieron mejoras con el fin de renovar continuamente el desempeño de los procesos. De igual forma, se analizaron los futuros trabajos que demandan esta investigación.

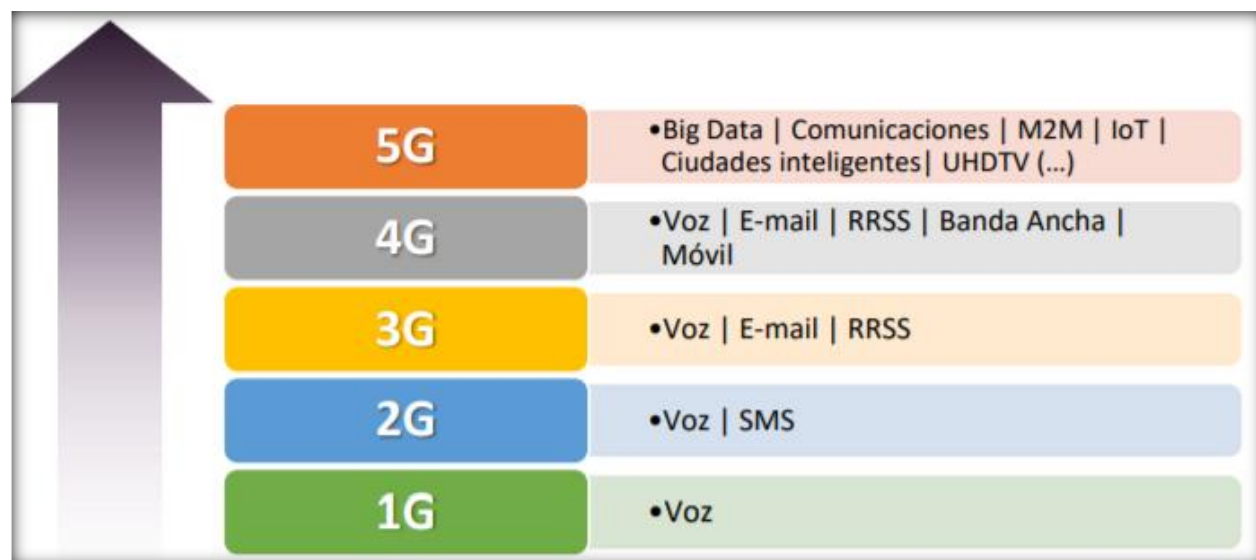
## Elementos del Sistema de Comunicaciones

Este capítulo caracteriza los elementos de comunicaciones, inicialmente se realiza un contexto mediante la descripción de las diversas configuraciones e implementación de las redes 4G y 5G-NR (New Radio) e IoT (Internet of Things) de banda estrecha. Posteriormente, se describe el software GNU radio y OpenAirInterface Software Alliance, así como el dataset a transmitir.

### Redes 5G

5G se refiere a la quinta generación de redes móviles donde se debe tomar como referencia la antigua red de 1G, en la cual los primeros teléfonos móviles ofrecieron un único servicio de llamadas de voz. Posteriormente, la tecnología 2G además de llamadas de voz permitió los SMS (Short Message Service), y los teléfonos celulares fueron evolucionando pasando a ser una herramienta de comunicación con múltiples recursos, la tercera generación 3G permitió visualizar páginas web, transferencia de fotos y datos de todo tipo, posteriormente evolucionó a redes 4G. Cuando la industria de las telecomunicaciones planea las redes 5G, con expectativas de navegación de hasta de 10 Gbps, representa cambios supremamente favorables para la transmisión de datos, donde otro factor importante que trae esta nueva tendencia es la latencia, toda vez que esta podría reducirse a 5 milisegundos, que garantizará conexiones extremadamente rápidas, casi en tiempo real, lo que será importante para campos como la telemedicina y todo lo que compone internet de las cosas IoT ( Kevin , 2021). En figura 1 es posible identificar gráficamente lo que se explicó anteriormente.

### *Evolución de las generaciones de las redes móviles*



Nota: Adaptado de evolución tecnológica servicios móviles, por Mintic, 2019, [https://mintic.gov.co/portal/715/articles-118058\\_plan\\_5g\\_2019120.pdf](https://mintic.gov.co/portal/715/articles-118058_plan_5g_2019120.pdf).

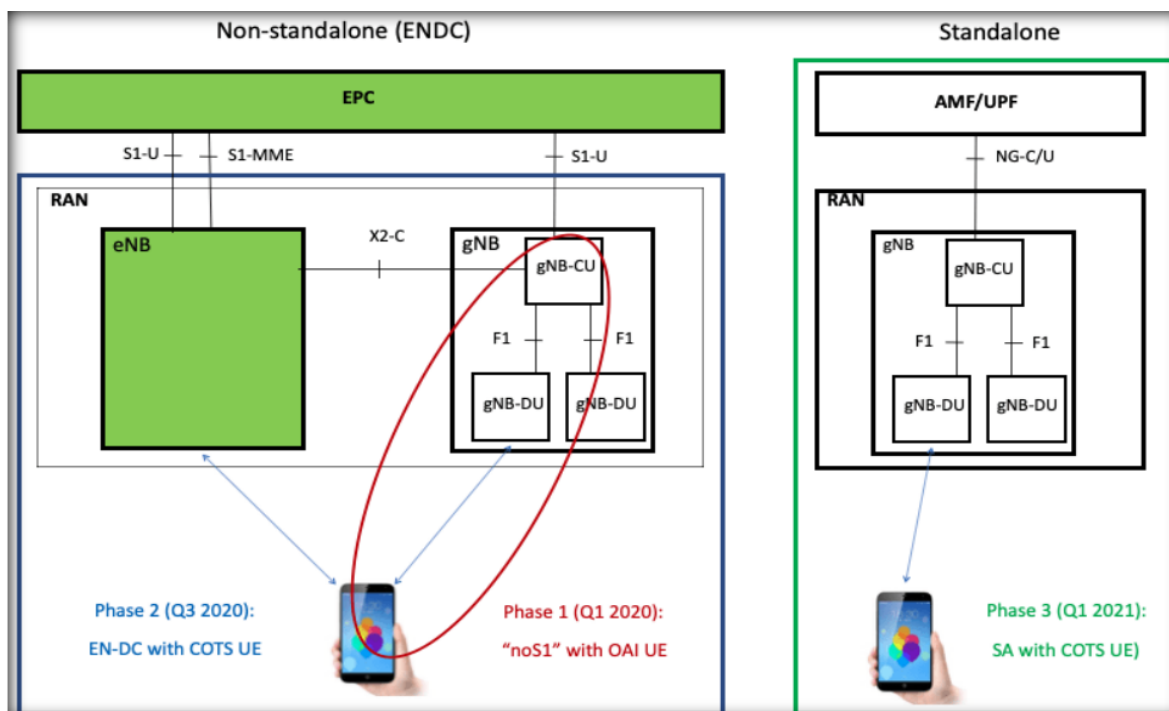
Es importante resaltar que la migración desde 4G LTE hacia 5G NR (New Radio) actualmente se realiza en dos fases, la primera es conocida como 5G NSA, una clase de quinta generación no autónoma, lo que se puede considerar un híbrido entre 4G LTE y 5G, simplemente se aprovecha toda la infraestructura que pertenece a las redes heredadas para iniciar a ofrecer a los dispositivos finales algunas de las bondades de 5G como es el caso de la velocidad en las búsquedas en internet, pero no aun las bajas latencias o la adopción de IoT de forma masiva, en otras palabras la transmisión entre el móvil y la antena se realiza a través de protocolos 5G, pero los datos se gestionan en tecnología LTE.

La segunda fase del despliegue de 5G se denomina 5G SA, es una quinta generación totalmente autónoma con todos los equipos y hardware definidos en el

release 16, esta fase es la encargada de abrir nuevas oportunidades al mercado en cuanto innovación de aplicaciones que aún son cosas de ciencia ficción, con velocidades de datos de hasta 10 Gbps, lo que es 10 a 100 veces más que 4G, por otro lado latencia de 1ms, ancho de banda de hasta 1000 veces más rápida por unidad de área, 100% de cobertura, reducción del 90% en el consumo de la red y hasta 10 años de duración de la batería en dispositivos IoT (Plokiko, 2019). Como se puede ver en la figura 2, se presenta el esquema de la arquitectura 5G NSA en la parte izquierda y en la parte derecha la arquitectura de la red de acceso 5G SA.



### Esquema 5G NSA y 5G SA



Nota: Adaptado de esquema 5G NSA y 5G SA, por OpenAirInterface, 2021, <https://openairinterface.org/oai-5g-ran-project>.

Es de resaltar que, 5G permitirá a los operadores ampliar las ofertas de servicios imponiéndose más allá de la banda ancha móvil mejorado (eMBB) de 4ms de latencia, nuevos servicios ligados a las comunicaciones ultra fiables de baja latencia (URLLC) de 1ms y las comunicaciones masivas de tipo maquina (mMTC) permitiendo cubrir más casos de uso e incursionar en nuevos mercados, 5G contará con una eficiencia espectral máxima de 30bps/Hz en DL y 15bps/Hz en UL.

Por otra parte, cada banda de frecuencia tiene sus ventajas y desventajas, las frecuencias bajas son aptas para cubrir áreas más grandes y llegar a ciertos espacios que no es posible con las altas frecuencias y haciendo uso de menos infraestructura gracias a que logra sobre pasar gran parte de los obstáculos que se encuentra en su camino pero la tasa de transmisión de datos es menor, a diferencia de las altas frecuencias que su tasa de transmisión de datos es mucho mayor pero su alcance se ve limitado en razón a que estas deben respetar la línea de vista para conservar el enlace, lo que hace que sea necesario una mayor infraestructura para cubrir las áreas requeridas.

También, es cierto que los avances tecnológicos de las redes móviles y la llegada de cada nueva generación traen consigo enfoques hacia la prestación de más y mejores servicios, uso más eficiente de la energía, mejor aprovechamiento del espectro en otras palabras eficiencia espectral, incremento exponencial del tráfico de datos, mayor cobertura, conexión de mayor número de dispositivos, optimización y conservación de una buena calidad del servicio.

## **IoT de Banda Estrecha**

Para hablar de IoT de banda estrecha es necesario definir nuevamente IoT y el concepto de banda estrecha, Internet de las cosas o Internet of the Things, se conoce como la interconexión de objetos físicos y sistemas inteligentes a la red, mediante la implementación de sensores, protocolos, software y otras tecnologías, lo que convierte a IoT en una de las tecnologías más disruptivas e innovadoras del planeta [26], gracias a IoT se disfrutan casas y ciudades inteligentes, que entregan servicios que hace 20 años eran tan solo ciencia ficción, como cafeteras que preparan el mocaccino favorito o collares para mascota que indican si el perro ha pasado una buena noche. Sin embargo, este fenómeno tiene impacto también en la industria y en la agricultura, influyendo en los procesos de automatización, seguridad, vigilancia, entre otros. Según el gigante tecnológico Cisco para el año 2023 habrá 29.300 millones de dispositivos conectados a Internet (europapress, 2020).

Se puede afirmar que gracias a IoT el mundo físico y el mundo digital interactúan y cooperan entre si apoyados en otros avances tecnológicos como el Wifi, el bluetooth o el Near-field communication NFC, la tecnología de computación por nube, los sensores de bajo costo o los revolucionarios asistentes de voz.

### ***Banda Estrecha***

En el mundo de las telecomunicaciones este término hace referencia a las conexiones a internet que utilizan un reducido ancho de banda, la banda estrecha se utiliza esencialmente para la transmisión de voz en medios no guiados, el desplazamiento de frecuencia es generalmente 3KHz a 5KHz. En las señales de audio

se utilizan frecuencias entre los 800Hz a 3KHz. Se puede afirmar que las conexiones de banda estrecha son todas aquellas que no superan los 128 kbps (ETMY, 2019).

### **Contexto para la Implementación de Una Red para Transportar los Datos de la Calidad del Aire**

OpenAirInterface maneja diversos proyectos en los que se encuentran; 5G RAN, Red Core 5G, MOSAIC5G y CI / CD, entre otros. A continuación, se describen cada uno de los proyectos que tienen relación con este trabajo de grado:

#### ***Proyecto de Red de Acceso de Radio OpenAirInterface 5G***

El propósito principal del proyecto OAI 5G RAN es construir la fuente de protocolos 5G tanto para gNB como para UE, lo que permite la implementación de un extremo a otro de una red 5G, El objetivo es desarrollar el software 5G Non-Stand Alone (NSA) RAN y admitir la conexión y tráfico con un User Equipment UE comercial 5G compatible con la arquitectura NSA. Como es bien sabido en la configuración NSA, el gNB se complementa con el LTE eNB que lleva el plano de control de la señalización 5G. La red central de paquetes evolucionada (EPC) 3GPP Rel-15 4G compatible con NSA está conectada a través de la interfaz S1 a eNB y la interfaz X2-C permite la conexión entre eNB y gNB para enrutar y administrar el flujo de tráfico IP (OpenAirInterface.org, Proyecto de red de acceso de radio OpenAirInterface 5G, 2021).

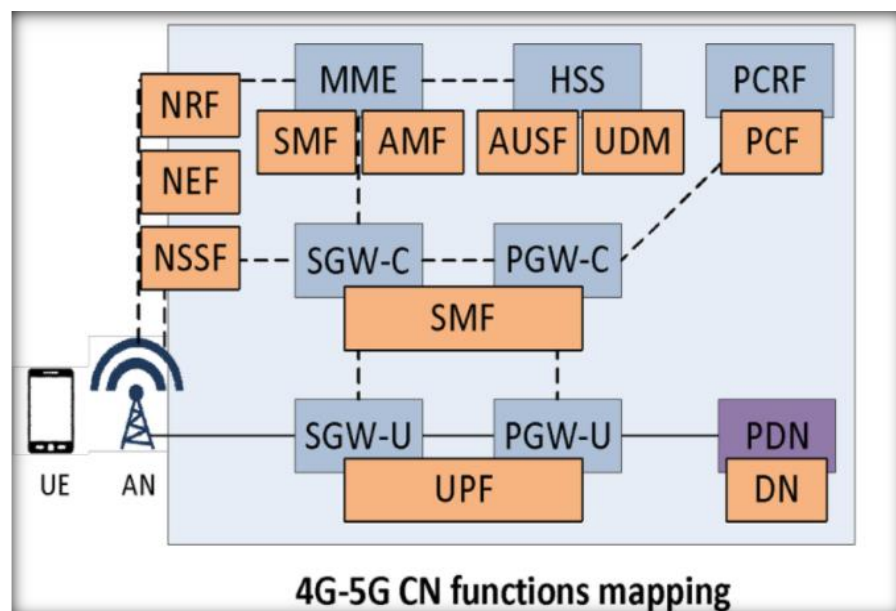
Todo este proceso cuenta con su hoja de ruta establecida que se puede encontrar en la página de referencia que inicia en marzo de 2021 y termina en marzo de 2022, además la fundación dice en su sitio web que está trabajando de la mano de colaboradores con otras iniciativas de la industria.

### Proyecto RED CORE 5G OpenAirInterface

El alcance de este proyecto 5G CN es ofrecer una red central 5G compatible con la arquitectura de 3GPP 5G, esto lo pretende hacer basado en la correspondencia entre el núcleo 5G y 4G LTE-EPC. El proyecto contará con tres fases, la primera Despliegue básico de AMF, SMF y UPF en contenedores docker basados en Ubuntu bionic con pruebas básicas de tráfico y flujo de llamadas, la segunda implementación continua de funciones como actualizaciones agregadas para los componentes de red existentes (AMF, SMF y UPF) y la tercera es la implementación del núcleo 5G independiente y marco para una arquitectura basada en micro servicios. En la figura 3 se puede apreciar una comparación entre los nodos que funcionan en 4G y que serán reemplazados por nuevos nodos virtualizados en 5G (OpenAirInterface, 2021).



Comparación arquitectura 4G y 5G.



Nota: Adaptado de esquema 5G core network, por OpenAirInterface, 2022,

<https://openairinterface.org/oai-5g-core-network-project>.

### **Grupo de Proyecto OpenAirInterface MOSAIC5G**

Este proyecto creado en 2021 pretende transformar el acceso a la RAN, así como las redes centrales CN para entregar servicios más rápidos en redes abiertas, de esta manera se pueden explorar diferentes casos de uso en las líneas industriales de producción. Mosaic5G es el primer entorno del mundo de plataformas de código abierto de I + D 5G, desde el control hasta el acceso a la red móvil. El rumbo principal del M5G PG será facilitar ejecuciones de software de los siguientes protocolos; el protocolo O-RAN E2 llamado E2 Agent, un controlador versátil de RAN llamado FlexRIC, un controlador Core flexible llamado FlexCN y un operador RAN y CN inteligente llamado Trirematics (OpenAirInterface, 2021)

### **Software a Implementar en el Proyecto**

Se utilizó GNU radio como herramienta para el procesamiento de señales de radio y OpenAirInterface como plataforma de tecnología de Red de Acceso de Radio (RAN) celular inalámbrica y Red Central (CN).

### **GNURadio**

Se conoce como una herramienta de software de código abierto bajo la Licencia Pública General de GNU (GPL) versión 3 o superior, mediante la cual se pueden procesar señales de radio mediante bloques de programación, este potente programa tiene la capacidad de funcionar con hardware de RF para crear radios de acuerdo con el software o simplemente como simulador. Así que, este software puede ser utilizado para una diversidad de aplicaciones, ejecuta todo el procesamiento de la señal. De ser necesario es capaz de escribir aplicaciones para recibir o enviar datos de flujos digitales. Este programa de radiofrecuencias, simula filtros, demoduladores, vocoders,

decodificadores, códigos de canal, elementos de sincronización, ecualizadores, y muchos otros elementos, todos estos elementos dentro del programa se conocen como bloques y existe la manera de interconectar los bloques para transmitir datos de un punto a otro, entendiendo esto, GNU radio solo maneja datos digitales ya que normalmente recibe muestras de señales en banda base y envía datos hacia los transmisores, sin embargo cualquier tipo de dato (bits, bytes, vectores, ráfagas o tipos de datos más complejos) puede transmitirse de un lugar a otro (GNURadio, 2021).

La mayoría de las aplicaciones que corren sobre GNU radio están escritas en Python, sin embargo, el procesamiento de señales está escrito en C++, esto no quiere decir que para usar GNU radio se necesite alta experiencia programando, ya que este software cuenta con una interfaz gráfica que permite al usuario crear aplicaciones de procesamiento de señales con solo arrastrar y soltar bloques, las cuales tienen ciertas funciones esenciales como grabación de señales RF, análisis de espectro, entre otras. Se utiliza largamente en entornos de academia, gobierno, investigación, industria y aficionados para respaldar tanto la investigación de comunicaciones inalámbricas.

### ***OpenAirInterface Software Alliance***

OpenAirInterface Software Alliance se reconoce con las siglas OSA es una fundación francesa establecida en 2014 sin fines de lucro, financiada por patrocinadores corporativos que desarrolla ambientes para la creación de software y hardware de código abierto, todo enfocado en redes móviles, especialmente para la red central (EPC) y la red de acceso (EUTRAN) de las redes celulares 3GPP. Esta alianza se encarga del patrocinio del trabajo inicial de EURECOM para crear OpenAirInterface TM hacia el desarrollo de 5G Cellular Stack en hardware comercial listo para usar

(COTS) (OpenAirInterface, 2021). En este momento cuenta con cinco proyectos que tienen que ver directamente con el desarrollo de software para soportar tráfico de 5G y su integración con otras tecnologías y protocolos, así; 5G RAN, Red Core 5G, Mosaic 5G y por último CI/CD que traduce integración continua / implementación continua.

### **Sensores de Calidad del Aire**

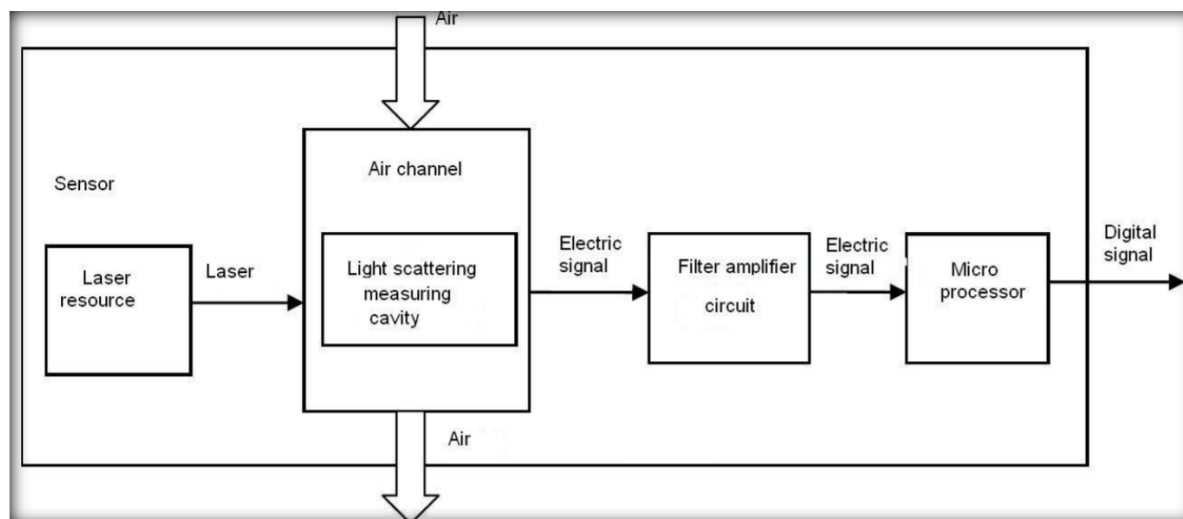
Un sensor es un dispositivo electrónico que toma un fenómeno físico y lo convierte en una señal analógica o digital, a la salida de un sensor se puede obtener una señal eléctrica ya sea de voltaje, corriente, resistencia u otro atributo que varía con el tiempo.

#### ***PMS5003 Digital Universal Particle Concentration Sensor***

El PMS5003 es un sensor universal de concentración de partículas digital, que tiene la capacidad de determinar la cantidad de partículas suspendidas en el aire y mostrarlas de manera digital en una interfaz de salida, este tipo de sensor es utilizado para reforzar las mediciones de equipos convencionales que también miden la cantidad de micro partículas que están suspendidas en determinado sitio.

Su principio de funcionamiento se basa en un láser de dispersión, en otras palabras, usa el láser para irradiar las partículas y calcular la cantidad de micro partículas en ese instante, el diámetro de las partículas las calcula mediante un procesador basado en la teoría MIE “La teoría MIE se utiliza para evaluar medidas de partículas cuyos diámetros no son significativamente mayores que la longitud de onda de la luz irradiada” (Paluga, 2021). En la figura 4 se puede observar un diagrama de bloques del principio de funcionamiento de este sensor:

Diagrama de bloques partes del sensor PMS5003



Nota: Adaptado de functional block diagram of sensor, por Manual PMS5003, 2016, [https://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual\\_v2-3.pdf](https://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual_v2-3.pdf)

### **HK-A5 Laser PM2.5 - 10 Sensor**

HK-A5 es un sensor digital universal de concentración de partículas, se utiliza para obtener el número de partículas suspendidas de 0,3 a 10 micras por unidad de volumen en el aire y la salida en forma de una interfaz digital, Así como los datos de masa de cada partícula. El sensor se puede incrustar en varios instrumentos y equipos de mejora ambiental para proporcionar datos puntuales y precisos (Sensor, 2021). En la figura 5 se puede observar físicamente el sensor con sus accesorios:

Imagen sensor HK-A5 Laser PM2.5 PM10



Nota: Adaptado de HK-A5 Laser PM2.5/10 Sensor, por datasheet, 2016,

<https://www.zeanoit.jp/pdf/3f-SEN0177.pdf>

### **Dataset**

Se refiere a un conjunto de datos tabulados en un sistema de almacenamiento, este término es muy conocido en las tecnologías de Big Data y es fundamental a la hora de entender la forma como se guarda información actualmente. Existen cuatro tipos de Datasets catalogados según su origen y formato, los cuales son usados según las necesidades de los modelos de datos a trabajar (KeepCoding, 2021).

El primero es conocido como Archivo, que es un fichero independiente donde se guarda toda la información, sus ventajas son la seguridad y rapidez para el trabajo con los datos, pero, se dificulta la escalabilidad y conexión con otros Datasets, el segundo es el Folder, que es la suma de varios Datasets guardados en una carpeta, pueden ser en formato .csv, .mif o dxf. Por otro lado, están las Bases de Datos, Por ejemplo, bases

de datos de Oracle, que solo trabajan para desarrollos y por último el Dataset Web, que es un grupo de datos compilados dentro de un sitio web.

### ***Dataset del Sensor HK-A5 Laser y Sensor PMS5003***

Teniendo en cuenta la definición de dataset, de acuerdo al datasheet de los sensores en referencia se tiene que dentro de su protocolo de comunicación se maneja una tasa de 9600bps y una longitud fija de paquete de 32 bytes (PMS5003, 2021), en la siguiente tabla se es posible verificar el valor de cada byte de información que el sensor arroja y su significado:

**Tabla 1.**

*Datos que se obtienen mediante sensor HK-A5 Laser y sensor PMS5003*

<b>No</b>	<b>No. Dato</b>	<b>Descripción</b>
1	Start character 1	0x42 (Fixed)
2	Start character 2	0x4d (Fixed)
3	Frame length high 8 bits	..... Frame length=2x13+2 (data+check bytes)
4	Frame length low 8 bits	
5	Data 1 high 8 bits	..... Data1 refers to PM1.0 concentration unit $\mu$
6	Data 1 low 8 bits	..... g/m <sup>3</sup> (CF=1, standard particle)
7	Data 2 high 8 bits	..... Data2 refers to PM2.5 concentration unit $\mu$
8	Data 2 low 8 bits	..... g/m <sup>3</sup> (CF=1, standard particle)
9	Data 3 high 8 bits	..... Data3 refers to PM10 concentration unit $\mu$
10	Data 3 low 8 bits	..... g/m <sup>3</sup> (CF=1, standard particle)
11	Data 4 high 8 bits	..... Data4 refers to PM1.0 concentration unit *
12	Data 4 low 8 bits	..... $\mu$ g/m <sup>3</sup> (under atmospheric environment)

No	No. Dato		Descripción
13	Data 5 high 8 bits	.....	Data 5 refers to PM2.5 concentration unit $\mu$
14	Data 5 low 8 bits	.....	g/m <sup>3</sup> (under atmospheric environment)
15	Data 6 high 8 bits	.....	Data 6 refers to concentration unit (under
16	Data 6 low 8 bits	.....	atmospheric environment) $\mu$ g/m <sup>3</sup>
17	Data 7 high 8 bits	.....	Data7 indicates the number of particles
18	Data 7 low 8 bits	.....	with diameter beyond 0.3 $\mu$ m in 0.1 L of air.
19	Data 8 high 8 bits	.....	Data 8 indicates the number of particles
20	Data 8 low 8 bits	.....	with diameter beyond 0.5 $\mu$ m in 0.1 L of air.
21	Data 9 high 8 bits	.....	Data 9 indicates the number of particles
22	Data 9 low 8 bits	.....	with diameter beyond 1.0 $\mu$ m in 0.1 L of air
23	Data 10 high 8 bits	.....	Data10 indicates the number of particles
24	Data 10 low 8 bits	.....	with diameter beyond 2.5 $\mu$ m in 0.1 L of air.
25	Data 11 high 8 bits	.....	Data11 indicates the number of particles
26	Data 11 low 8 bits	.....	with diameter beyond 5.0 $\mu$ m in 0.1 L of air
27	Data 12 high 8 bits	.....	Data12 indicates the number of particles
28	Data 12 low 8 bits	.....	with diameter beyond 10 $\mu$ m in 0.1 L of air.
29	Data 13 high 8 bits	.....	Data13 Reserved
30	Data 13 low 8 bits	.....	
31	Data high 8 bits	.....	Check code=Start character1+ Start
32	Data low 8 bits	.....	character2+.....+data13 Low 8 bits

Nota: Adaptado de HK-A5 Laser PM2.5/10 Sensor, por datasheet, 2016,

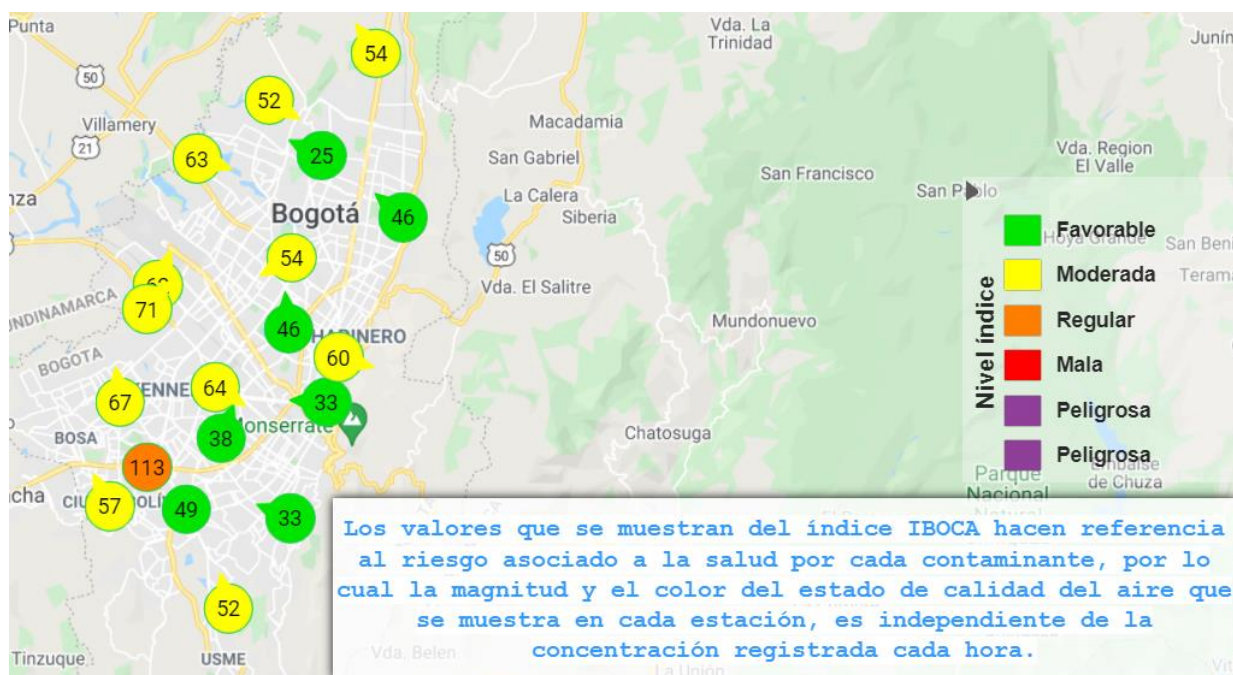
<https://www.zeanoit.jp/pdf/3f-SEN0177.pdf>.

### **Sensores de la Red de Monitoreo de Calidad del Aire en Bogotá RMCAB**

La RMCAB cuenta con 20 estaciones equipada de sensores y analizadores automáticos, los cuales reportan datos actualizados con frecuencia horaria, informando la calidad del aire en la ciudad. Cada estación se encuentra ubicada estratégicamente atendiendo a los requerimientos definidos en la normatividad vigente, lo que permite un barrido por la ciudad para informar la cantidad de material particulado PM10 y PM2.5 en el ambiente y otras variables como O3, SO2, NO2 y CO. (Secretaria de Ambiente de Bogotá, 2021). La figura 6, tomada de la página de la Secretaria de Ambiente se muestra la ubicación geográfica de las 20 estaciones de monitoreo:



*Mapa de las 20 estaciones de monitoreo RMCAB*



Nota: Adaptado de mapas interactivos, por red de monitorio de calidad del aire en Bogotá, 2022, <http://rmcab.ambientebogota.gov.co/home/map>.

Cada estación de monitoreo es independiente y gracias al aplicativo de la web de la secretaria se pueden exportar reportes filtrando la estación de interés, el día o la hora, asimismo reportes en tabla, grafica o en Excel. En la figura 7 se puede apreciar los reportes que según la Secretaría de Ambiente arrojan las estaciones de Puente Aranda, Suba, El Tunal y Usaquén, primero en tabla y luego a modo de gráfica:

*Información estaciones de monitoreo de Puente Aranda, Suba, El Tunal y Usaquén del 24 de septiembre de 2021*

DateTime	Puente Aranda		Suba		Tunal		Usaquen	
	PM10	PM2.5	PM10	PM2.5	PM10	PM2.5	PM10	PM2.5
	µg/m3	µg/m3	µg/m3	µg/m3	µg/m3	µg/m3	µg/m3	µg/m3
24-09-2021 01:00	21.2	12	27.5	16	25	6	9.6	8
24-09-2021 02:00	27.4	12	19.3	15	23	6	8	8
24-09-2021 03:00	22.6	8.8	28.9	8	22	12	16.5	9
24-09-2021 04:00	17.1	8.8	21.7	9	24	10	10.1	9
24-09-2021 05:00	14.4	14.2	20	13	22	12	11.6	9
24-09-2021 06:00	14.7	12	34	17	31	14	10.6	5
24-09-2021 07:00	20	18.6	45.1	37	35	10	7.2	1
24-09-2021 08:00	23.6	17.5	45.9	25	28	6	7.7	3
24-09-2021 09:00	30	18.6	47.9	16	21	8	15.8	7
24-09-2021 10:00	24.7	----	37.6	4	24	----	13.1	8
24-09-2021 11:00	23.2	13.1	27.6	8	----	----	13.3	11
24-09-2021 12:00	32	15.3	57.8	7	----	----	14.3	13
24-09-2021 13:00	22.7	----	54.4	5	54	13	18.4	10
24-09-2021 14:00	26.8	18.6	40.8	5	27	15	14.8	10
24-09-2021 15:00	19.4	12	31.1	4	30	12	13.3	12
24-09-2021 16:00	19.8	7.7	26.8	6	22	10	16	16
24-09-2021 17:00	19	----	30.3	17	19	8	20.2	11
24-09-2021 18:00	20.2	13.1	29.6	11	38	11	15.5	8
24-09-2021 19:00	14.1	6.6	34.8	10	21	8	13.6	4
24-09-2021 20:00	12.3	3.3	25.7	10	28	8	11.3	5

Nota: Adaptado de reporte de monitoreo, por red de monitorio de calidad del aire en Bogotá, 2022, <http://rmcab.ambientebogota.gov.co/report/MonitorReport>.

De las 20 estaciones de monitoreo con las que cuenta RMCAB se escogieron las ubicadas en Puente Aranda, Suba, Tuna y Usaquéen, se tuvieron en cuenta los monitores de PM 10 y PM 2.5, en la figura No. 8 se puede observar mediciones del 24 de octubre del 2021 desde las 1:00 hasta las 20:00 horas, estos reportes sacados la página de la secretaria de ambiente de Bogotá se pueden exportar mínimo con intervalo de tiempo de una hora. Tomando como referencia la estación de Puente Aranda, se puede observar que los datos de la calidad del aire son variables a través de todo el día, en cuanto a PM 10 se presentó un pico máximo de  $32 \mu g/m^3$  a las 12:00 horas y un mínimo de  $12.3 \mu g/m^3$  a las 20:00, por otro lado, las PM 2.5 presentó un pico máximo de  $18.6 \mu g/m^3$  a las 07:00, 09:00 y 14:00 horas y un mínimo de  $3.3 \mu g/m^3$  a las 20:00. Adicionalmente los reportes de PM 2.5 de las 10:00, 13:00 y 17:00 horas no se muestran en pantalla, quiere decir que ese horario los instrumentos de medición no trabajaron de manera correcta o se tiene algún error de configuración para con la variabilidad de esos horarios.

Desde un análisis cualitativo de los datos medidos por la RMCAB, los reportes cada hora, podrían no ser suficientes para informar a la ciudadanía de la calidad de aire que se está respirando en la ciudad. De acuerdo a un estudio realizado por (Baldauf, y otros, 2012), en conjunto con la U.S. Environmental Protection Agency, la Federal Highway Administration y la Alion Science & Technology de los Estados Unidos, llamado Air quality variability near a highway in a complex urban environment, el estudio combinó mediciones de cuatro estaciones de monitoreo del aire en sitios fijos y móviles con experimentos de túneles de viento de laboratorio para examinar cómo el transporte y la dispersión de los contaminantes emitidos por el tráfico varían con los

cambios en la configuración de las carreteras, para ello, los datos se recopilaron en un tiempo promedio de 1 un minuto, sin embargo los informes se recolectaron cada 5 minutos. Para llevar a cabo estas mediciones, una estación se ubicó a 100 metros en dirección predominante contra el viento, y por otro, lado tres estaciones adicionales ubicadas a 20 metros, 100 metros y 300 metros en la dirección a favor del viento. Este estudio de la variabilidad de la calidad del aire demostró que las características urbanas influyeron en el transporte le material contaminante, así mismo la ubicación de las estaciones de medida y la altura a la cual se emplazaron. Dicho lo anterior, es posible mejorar una la red de 20 estaciones de RMCAB, debido a que para cubrir  $307Km^2$  de superficie urbana que tiene Bogotá, se necesitan más puntos de medición, separados por distancias menores a las actuales, por lo que es urgente implementar nuevas tecnologías de transporte de datos de grandes volúmenes como lo es IoT (Internet de las cosas), y aumentar así tanto la cantidad de estaciones que miden las micro partículas contaminantes como el envió de datos en tiempo real tomando por lo menos un periodo de 5 minutos o menos.

## **Diseño del Sistema para Transmitir Información Referente a la Calidad del Aire en Bogotá Empleando IoT de Banda Estrecha y Redes Móviles**

Como primera medida, para realizar el diseño del sistema de transmisión de datos utilizando OFDM (Orthogonal Frequency Division Multiplexing) se ejecutaron las simulaciones correspondientes en GNU Radio, que se encarga de la red de acceso, donde se toma la señal o el conjunto de datos, se muestrea, cuantifica, codifica, modula y se multiplexa, dicha simulación se explicará en una serie de etapas en la sección 3.1 con su respectivo diagrama de bloques.

Posteriormente, para establecer una red de Internet de las cosas de banda estrecha, dentro de una red móvil es necesario valerse de dos módulos de la familia Quectel. En la sección 3.2 se explica detalladamente como establecer comunicación con el kit EVB (Evaluation Board) GSM (Global System for Mobile Communications) NB-IoT (Narrowband-Internet of Things) de Quectel Wireless Solutions y el módulo BC95-G NB-IoT mediante un ordenador con sistema operativo Windows. En este caso se señalan los parámetros de configuración, el software del fabricante, así como las líneas de código de Comandos AT que se deben ejecutar para comunicarse con la SIM (Subscriber Identity Module) card.

### **Diseño de un Sistema de Transmisión OFDM**

Tal como se mencionó en la sección que trata sobre la quinta generación de las comunicaciones móviles 5G o New Radio, esta tecnología proyecta ser un salto revolucionario en términos de velocidad de transmisión de datos, latencia, conectividad masiva, confiabilidad de la red y eficiencia energética (Shafi, y otros, 2017). Ahora bien, para aumentar la velocidad de transmisión, 4G y 5G se apoya en la OFDM (Orthogonal

Frequency Division Multiplexing), la cual básicamente es un tipo de multiplexación que consiste en enviar un conjunto de ondas portadoras en diferentes frecuencias ortogonales entre sí, cada una de estas portadoras transporta información, de esta manera, OFDM aprovecha un determinado ancho de banda para propagar mayor cantidad de información que otros tipos de multiplexación en el mismo periodo de tiempo.

Es necesario indicar que esta simulación de transmisión por OFDM está basada en un algoritmo base que se puede encontrar en la wiki oficial de GNU Radio (GNUradio, 2022), teniendo en cuenta lo anterior y tomando como referencia dicho algoritmo se realizan los ajustes necesarios para simular la transmisión de datos que se requiere para este proyecto. A continuación, se describen una serie de bloques por etapas de programación que se tuvieron en cuenta para simular la transmisión por OFDM, el conjunto de datos se asemeja a los que puede medir el sensor HK-A5 Laser o el sensor PMS5003.

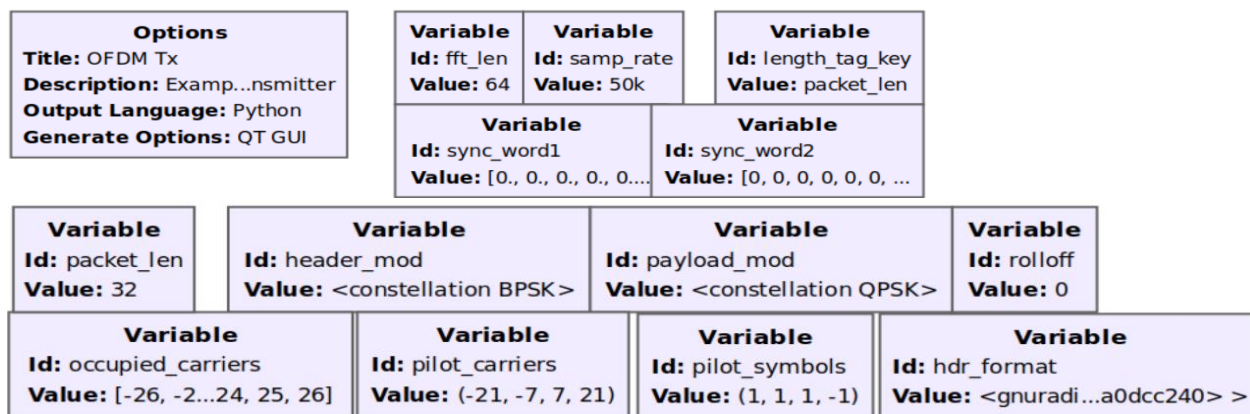
### ***Opciones y Variables Simuladas***

En esta etapa se define el bloque de opciones y las variables que OFDM utiliza para la transmisión de paquetes. El bloque de opciones como se visualiza en la figura 8, establece parámetros especiales para el diagrama de flujo y solo se permite uno para por diagrama de flujo, en este mismo se define el título del diagrama, el autor y la descripción del mismo, además, se define el ID, el lenguaje de programación que en este caso es Python. Por otro lado, están las variables utilizadas para configurar las otras etapas de transmisión de datos, por ejemplo, las variables para la FFT (Fast

Fourier Transform), el tamaño de los paquetes que se van a transmitir, el paquete de carga y los parámetros numéricos que definen las frecuencias portadoras de OFDM.



### Definición bloque de opciones y variables



Nota: Autoría propia

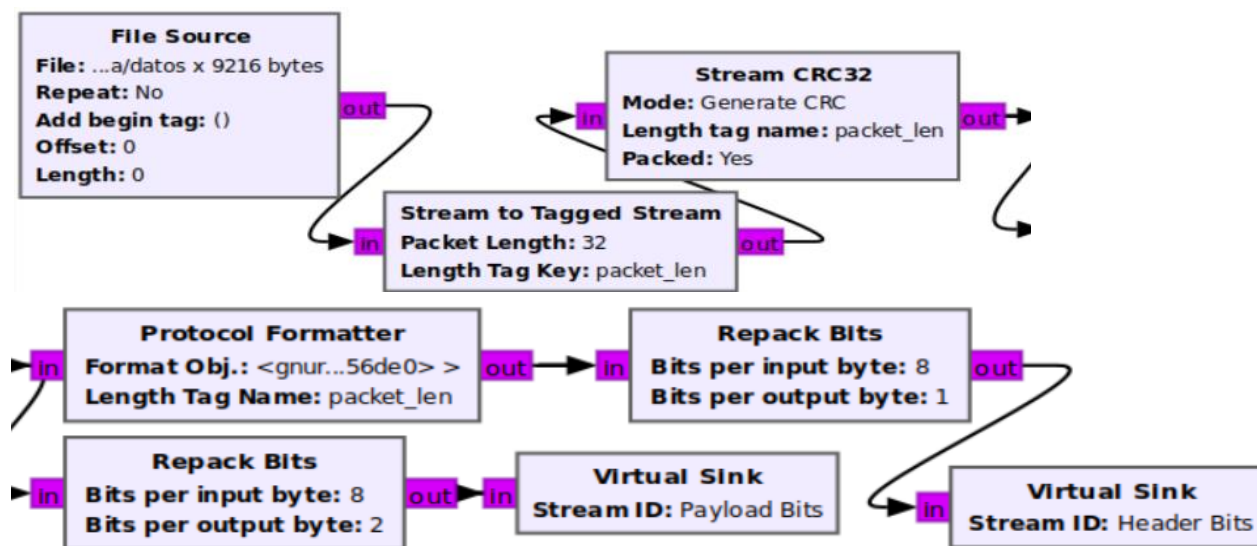
### Codificación de los Datos y Transmisión de Paquetes

En este paso se hace un mapeo de los datos en paquetes para su transmisión, para ello, con el primer bloque de la izquierda de la figura 10, que contiene implícito un archivo plano con la información se genera un numero de muestras aleatorias entre un mínimo y máximo que determina el programador o la necesidad, el segundo bloque que se tiene es el Stream to Tagged Stream, este bloque define el tipo de dato que se transporta, si es un tipo byte, complex, float, int o short, simplemente lo que hace es agregar etiquetas de longitud a intervalos regulares, en este paso es donde se llama la variable packet\_len definida en la primera etapa. El tercer bloque del diagrama de la figura 9, es el Stream CRC32 que es un algoritmo utilizado para realizar la codificación de los datos y mediante ésta encontrar errores en la transmisión de datos, de éste, se desprenden dos bloques el Packet Header Generator y Repack Bits, el primero se

encarga de generar el encabezado del paquete y el segundo vuelve a empaquetar los bits de flujo de entrada en flujo de salida, los cuales pertenecen al payload y los últimos bloques son los conectores virtuales para continuar con en el diagrama de flujo en la parte inferior.



### *Etapa de mapeo de los datos en paquetes*



Nota: Autoría propia

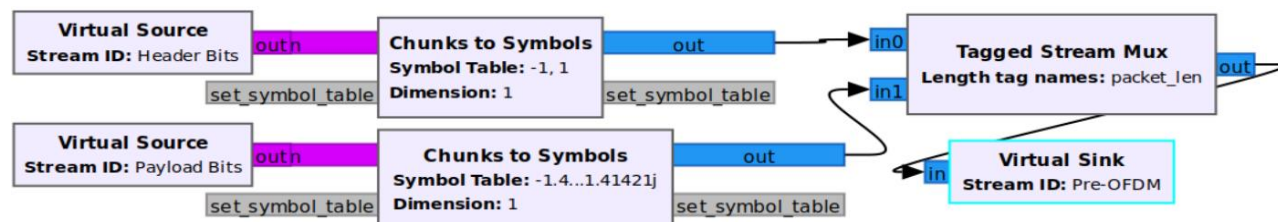
### *Etapa de Multiplexación*

Como se evidencia en la figura 10, durante esta etapa se llaman los Stream ID *Header Bits* y *Payload Bits* que fueron tratados en el paso anterior, los siguientes bloques llamados Chunks to symbols cumplen la función de procesar los datos que ingresan de tipo Byte y salen de tipo Complex, realizando un mapeo general. El siguiente bloque llamado Tagged Stream Mux recibe los datos de los anteriores bloques y realiza un proceso de multiplexación, este bloque cuenta con dos entradas y una salida, la función de este bloque es etiquetar nuevamente el flujo de datos que

ingresa y transmitirlo secuencialmente, esta señal de salida mixta tiene una nueva etiqueta de longitud. Finalmente, se une a un bloque Virtual Sink para prolongar la señal a la siguiente etapa.



### *Etapa de procesamiento y etiquetado de los datos*



Nota: Autoría propia

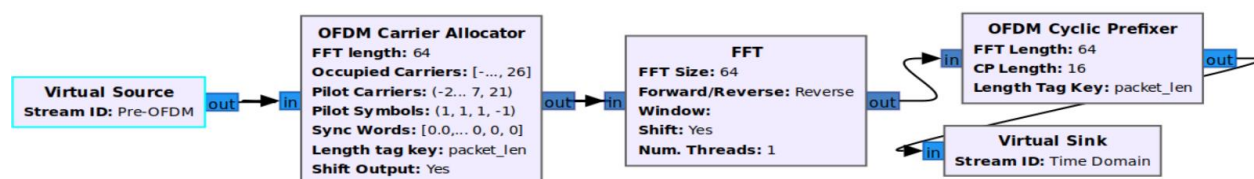
### ***Etapa de División en Frecuencia Ortogonal y Prefijo Cíclico***

En la figura 11 se pueden observar los bloques de implementación de OFDM para la transmisión de paquetes, como primera medida se reciben los datos del *Pre-OFDM*. El bloque *OFDM carrier Allocator* utiliza la mayoría de las variables definidas en el primer paso, para asignar cada portadora a una frecuencia ortogonal; FFT length define la cantidad de bits de la FFT, así mismo las variables Occupied Carriers, Pilot Carriers, Pailot Symbol, Sync words y Length tag key, convierten los flujos de símbolos de modulación escalar complejos los cuales sirven para la entrada de una IFFT (Inverse Fast Fourier Transform). Posterior a ello se encuentra el bloque FFT que básicamente realiza la Transformada Rápida de Fourier para los datos tipo Complex obteniendo la señal en el dominio de la frecuencia, con la transformada queda realizada la división de frecuencia ortogonal. Luego en el bloque OFDM Cyclic Prefix añade un prefijo cíclico a la señal OFDM, en los parámetros se define la longitud de FFT, la longitud de las muestras, la longitud del flanco de rolloff y la clave de la etiqueta

de longitud. Finalmente, se une un bloque Virtual Sink para prolongar la señal a la siguiente etapa.



### *Etapa de parametrización de OFDM y FFT*

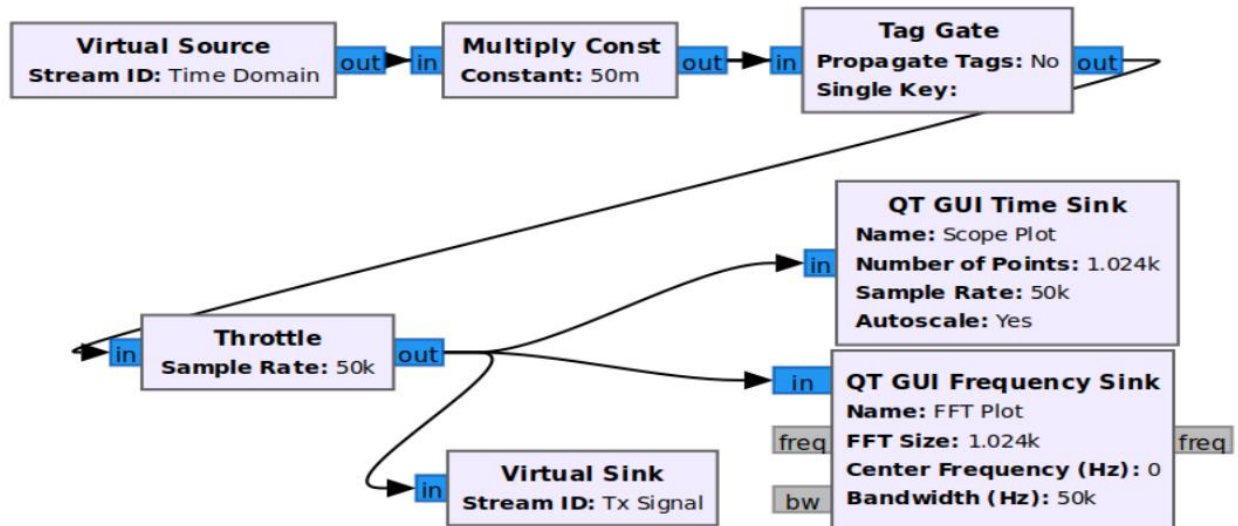


Nota: Autoría propia

### *Etapa de Visualización de la Señal*

Los bloques del diagrama de la figura 12, se encargan de mostrar la señal en una interfaz gráfica, primero se llama el Stream ID, para ello *Time Domain* que viene de la anterior etapa, y se pasa la señal por un multiplicador, en este caso se parametriza con una constante de 0.05, equivalente a 50ms, que será el tiempo de ejecución, dicho tiempo es ajustable. El siguiente bloque, Tag Gate se utiliza para evitar que las etiquetas de los datos se propaguen, se puede decir que es un filtro de etiquetas. Luego, en el bloque Throttle llama la varia *samp\_rate* con un valor de 50.000 y su función principal es igualar el procesamiento del computador a la tasa de muestreo, de este último bloque se desprenden tres hilos, uno de ellos para graficar la señal en el dominio del tiempo, otro para la señal en el dominio de la frecuencia y otro para un bloque Virtual Sink para continuar con el diagrama.

### Etapa de diagramación de la señal

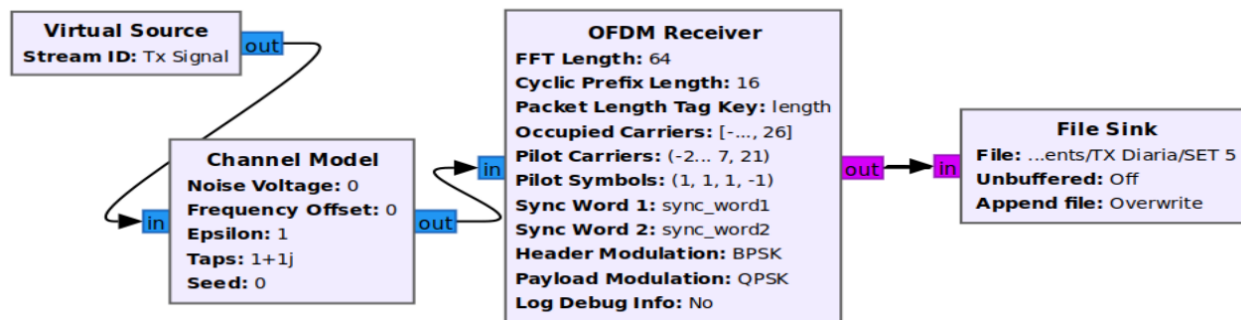


Nota: Autoría propia

### Etapa de Recepción

En esta última etapa se simula la recepción de la señal OFDM, primero se llama el Stream ID: Tx Signal, la cual viene del paso anterior, esta salta por un simulador de canal y luego por el bloque OFDM Receiver que básicamente es un demodulador de señal OFDM parametrizado con las respectivas variables con las que se moduló la señal en un comienzo, por último, la señal es entregada al bloque *File Sink* que se encarga de depurar la información e imprimir en un archivo .txt los datos recibidos por el sistema en una ubicación predeterminada, como se evidencia en la figura 13. Al verificar dicho archivo plano, se puede constatar la transmisión exitosa de los datos que ingresaron al bloque *File Source*.

### Etapa de demodulación de la señal OFDM



Nota: Autoría propia

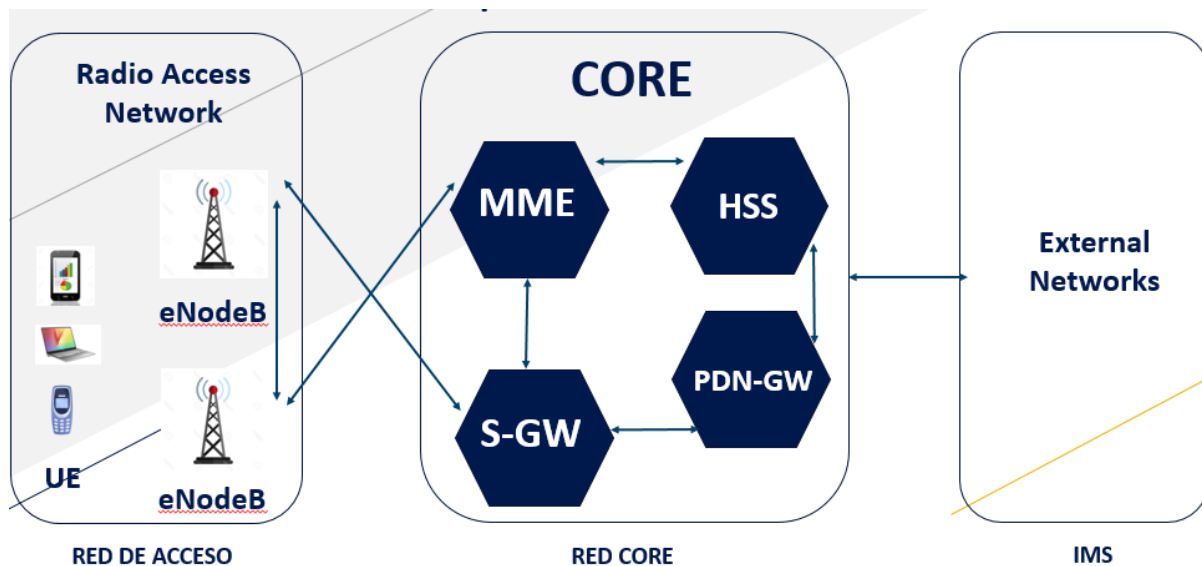
Finalmente, se evidencia que este diseño de sistema de transmisión que utiliza OFDM como tecnología de multiplexación, permite simular la conexión entre el usuario final y la estación base, todo lo anterior representa una forma teórica del proceso de envío de datos en redes móviles 4G y 5G.

### Diseño del Sistema de Transmisión NB-IoT Orientado a Redes 5G

Como introducción en este apartado, y a manera de contexto, es necesario ubicar los elementos del CORE y la RAN (Radio Access Network) en una red móvil NR (New Radio). En la figura 14 se observa que el CORE está compuesto por cuatro entidades: el MME (Mobility Management Entity), el HSS (Home Subscriber Server) y los SGW (Serving Gateway). Por otro lado, la red de acceso compuesta por el eNodeB o estación base (llamado así en generaciones pasadas) y los UE (User Equipment), estos últimos son los que se denominan dispositivos IoT (Internet of Things).

Cada una de estas entidades, se emularán en el siguiente capítulo, para conformar el sistema de transmisión de la calidad del aire.

Esquema red de acceso y del core con sus respectivas entidades



Nota: Autoría propia

A continuación, se explicarán las características de los elementos electrónicos que se utilizaron en el proyecto para emular la red móvil, los módulos de la marca Quectel transmiten las mediciones de calidad del aire por medio de una red de banda estrecha, en ella un usuario sería el sensor de medición que se conecta a la red, y este a su vez se denomina un dispositivo IoT que tiene la capacidad de interactuar con la red móvil al enviar y recibir señales y datos.

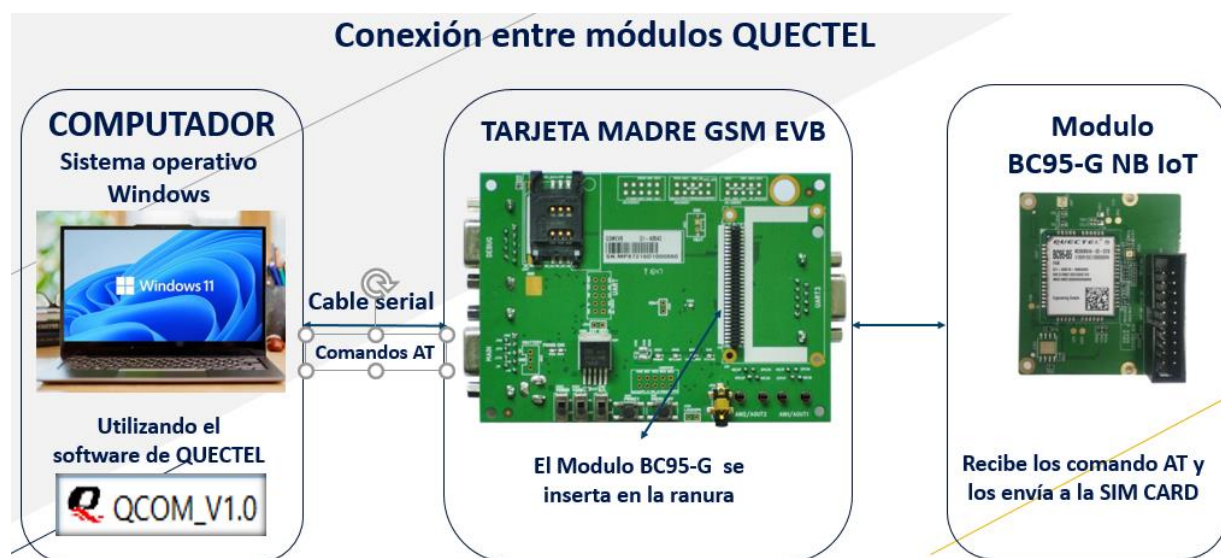
***Diseño de un Sistema de Transmisión entre la Tarjeta Madre GSM EVB KIT y el Módulo de Internet de las Cosas de Banda Estrecha BC95-G de QUECTEL.***

Con el objetivo de configurar una red con dispositivos IoT dentro de una red móvil de última generación se utilizaron dos módulos de la familia Quectel, el más robusto es un kit de desarrollo que se conoce técnicamente como GSM EVB KIT, por otro lado, para complementar el funcionamiento del kit se utilizó el módulo BC95-G que

soporta múltiples gamas de frecuencia, ideal para el desarrollo de este proyecto. En la figura 15 se puede observar el modo y orden de conexión de las tarjetas para lograr una comunicación serial entre el ordenador con sistema operativo Windows y la SIM CARD (Subscriber Identity Module).



*Orden de conexión de las tarjetas QUECTEL*



Nota: Autoría propia

El kit EVB (Evaluation Board) GSM (Global System for Mobile Communications) NB-IoT (Narrowband-Internet of Things) de Quectel Wireless Solutions está diseñado para desarrollar aplicaciones basadas en los módulos GSM, UMTS (Universal Mobile Telecommunications System) y NB-IoT (Narrowband-Internet of Things) de Quectel. El kit ofrece facilidad de uso con muchas interfaces funcionales y leds para visualizar el estado de trabajo del módulo, es compatible con el módulo BC95-G, toda la información técnica del Kit se puede encontrar en la guía de usuario (Electronics, 2022). Por otro lado, el BC95 es un módulo NB-IoT de alto rendimiento con un

consumo de energía extremadamente bajo que se acopla a las necesidades del proyecto.

### ***Instalación de Drivers en el Ordenador***

Las tarjetas de desarrollo o dispositivos que se conectan a un computador necesitan obligatoriamente un controlador para su correcto funcionamiento, para los módulos seleccionados en este proyecto, se explica paso a paso la instalación y protocolos de comunicación que se necesitaron.

Para el caso del KIT GSM EVB y módulo BC95-G NB IoT, se requiere una conexión serial para comunicarse con un ordenador, para ello, se ejecutan los controladores de cable RS232 (Recommended Standard 232), que es un tipo de comunicación serial utilizado en cortas distancias, así mismo se instala el software que controla el KIT GSM EVB; QCOM\_V1.0. En este orden, la comunicación entre el ordenador y el kit se realiza por intermedio de comandos AT, que son el lenguaje de programación para dar instrucciones a los terminales móviles o los modem.

En este caso los comando AT se ejecutan desde el software previamente instalado, para comunicarse con la SIM CARD (Subscriber Identity Module). A continuación, se puede observar en la figura 16 la interfaz gráfica que ofrece Quectel para la transmisión serial entre el ordenador y el módulo BC95-G NB IoT por intermedio del KIT GSM EVB. Una vez se instalan los programas necesarios para la comunicación serial, se realiza la conexión física por intermedio de un cable RS232

## Comunicación serial por comandos AT utilizando QCOM\_V1.0

The screenshot displays the QCOM\_V1.0 software interface, which is used for sending AT commands to a modem via a serial port. The interface is divided into several sections:

- COM Port Setting:** Configures the serial port parameters.
  - COM Port: 4
  - Baudrate: 9600
  - StopBits: 1
  - Parity: None
  - ByteSize: 8
  - Flow Control: No Ctrl Flow
- Command List:** A table of AT commands with checkboxes for execution, HEX/Enter options, and a delay in milliseconds.
 

	HEX	Enter	Delay(mS)
<input checked="" type="checkbox"/> 1: AT+NRB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	10000
<input checked="" type="checkbox"/> 2: ATi	<input type="checkbox"/>	<input checked="" type="checkbox"/>	500
<input checked="" type="checkbox"/> 3: AT+CM1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 4: AT+NCONFIG=AUTOCONNECT,TRUE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 5: AT+NCONFIG=CR_0354_0338_SCRAMBLI	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 6: AT+NCONFIG=CR_0859_SI_AVOID.FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 7: AT+CGDCONT=0,"IP",,0,0,,,,,0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 8: AT+NEARFCN=0,9448	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 9: AT+CGATT=0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 10: AT+CGATT=1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	10000
<input checked="" type="checkbox"/> 11: AT+CGDCONT?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 12: AT+CSQ	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 13: AT+NUESTATS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2000
<input checked="" type="checkbox"/> 14: AT+CDPS?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 15: AT+CGSN=1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 16: AT+CEREG?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	3000
<input checked="" type="checkbox"/> 17: AT+CGATT?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 18: AT+CGPADDR	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 19: AT+NBAND=?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input checked="" type="checkbox"/> 20: AT+CGDCONT=0,"IP",,"apn0.openair4G.eur	<input type="checkbox"/>	<input checked="" type="checkbox"/>	5000
<input checked="" type="checkbox"/> 21: AT+QPING=1,"www.google.com"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input type="checkbox"/> 22:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1000
<input type="checkbox"/> 23:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> 24:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> 25:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> 26:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> 27:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> 28:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> 29:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
- Log:** A text area showing the serial communication log. It includes timestamps and messages such as "REBOOTING", "Verified", "Protocol A... Verified", "Apps A..... Verified", "REBOOT\_CAUSE\_APPLICATION\_AT", "Neul", "JK", "Quectel", "BC95-G", "Revision:BC95GJBR02A02", "DTR", "RTS", "CTS", "RI", "HEX String", "Show In HEX", "Send With Enter", "Send Command", "Select File", "Send File", "Save Log", "D:\Tools\QCOM\_LOG.txt", "Operation", "Clear Information", "View File", "Show Time", "Load Test Script", "Run Times: 1", "Delay Time(mS): 500", "Save As Script", "Stop".

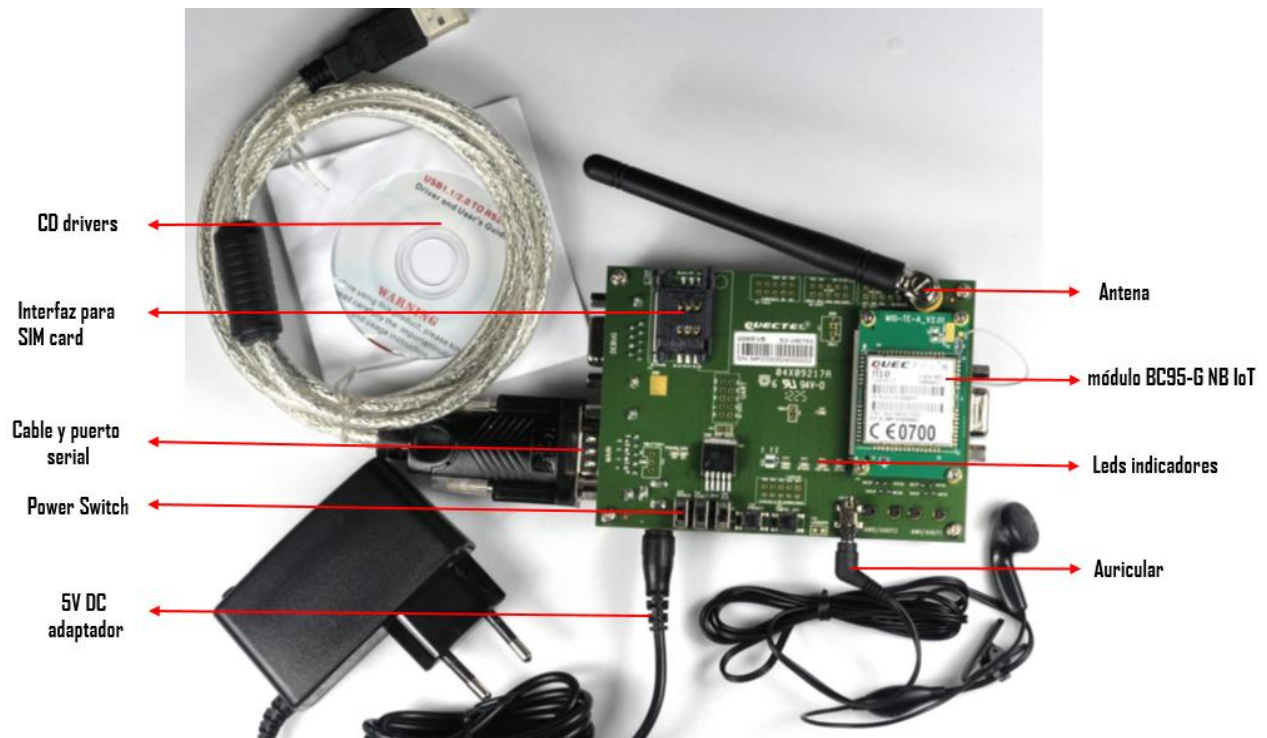
Nota: Autoría propia

La tarjeta de desarrollo de Quectel GSM EVB y el módulo NB-IoT BC95-G son utilizados para probar la tecnología de IoT en una red móvil, esto es posible al aprovechar la infraestructura de red celular que provee OpenAirInterface. Esta tecnología es actualmente la más avanzada de los proyectos de la Alianza de Software OpenAirInterface. El proyecto NB-IoT tiene según sus desarrolladores un avance del 95% y actualmente es desarrollada por EURECOM, B-COM and NTUST (Ariza Vesga,

2021), a continuación en la figura 17, se muestra en la figura el KIT GSM y el módulo BC95-G NB IoT con todos sus accesorios.



*GSM EVB KIT y módulo BC95-G NB IoT de QUECTEL*



Nota: Autoría propia

## **Resultados de las Simulaciones y Configuración de la Red Móvil**

Este cuarto capítulo expone la manera en que se configuró el Core de la red móvil, asociando cada una de las entidades, tales como; el MME (Mobility Management Entity), el HSS (Home Subscriber Server) y los SGW (Serving Gateway), para ello se necesitó un equipo de cómputo con sistema operativo Linux bajo la distribución de Ubuntu 16.04 LTS. De la misma manera, se explica los pasos para ejecutar las líneas de código para acceder a los diferentes repositorios, contenedores de Docker, base de datos de Cassandra y conectar todo el Core de la red con en eNB (Evolved Node B).

Una vez se tiene claro el concepto de arquitectura de una red móvil de última generación y las entidades que componen, se procede a ejecutar en lenguaje de programación Python el código para crear el core, lo anterior de acuerdo a las instrucciones de la página oficial de OpenAirInterface. En la sección 4.2 se explica paso a paso los repositorios, herramientas y líneas de programación que se utilizaron para ejecutar todo sobre la distribución Ubuntu 16.04 LTS.

Al final del capítulo, en la sección 4.3 se explica el paso a paso para simular el eNodeB, para ello, se debe utilizar otro repositorio de OpenAirInterface para lanzar el software del mismo y conectar físicamente la USRP al ordenador donde se está realizando todo el ejercicio de OpenAirInterface. En este apartado se especifica el tipo de USRP que se utiliza en el proyecto y se muestra la conexión al core mediante líneas de código.

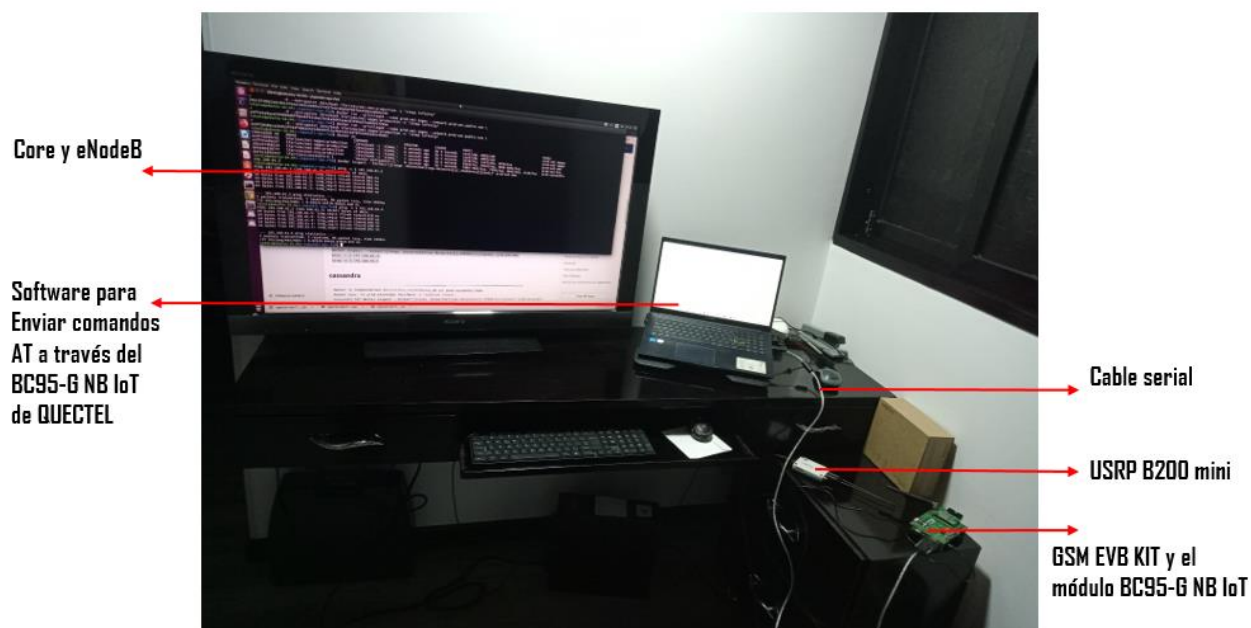
### **Configuración General de la Red Móvil**

En la figura 18 se muestra la conexión de los elementos físicos de la red móvil; A la izquierda se aprecia una pantalla donde se muestra la configuración en

OpenAirInterface que contiene el core y el eNodeB, en el portátil del centro se muestra la interfaz de Quectel con los comandos AT que se utilizan para conectar el módulo NB-IoT BC95, posteriormente, desde el portátil se conecta un cable serial con el que se controla el chip o módulo que hace las veces de User Equipment y por último la USRP que está conectada con un cable USB al ordenador con sistema operativo Linux donde se ejecuta el core y el eNodeB.



*Conexión física de los equipos del core y eNodeB*



Nota: Autoría propia

### **Configuración en OpenAirInterface del CORE y RED DE ACCESO para Transmisión de Datos en la Red Móvil**

La configuración de la RED DE ACCESO y del CORE con todas sus entidades, requiere un ordenador con sistema operativo Linux, o en su defecto hacer la instalación de Linux en el mismo computador con Dual Boot. Para que funcione la red de manera

correcta, se requiere que el sistema operativo trabaje bajo la distribución de Ubuntu 16.04 LTS, así mismo, un procesador Core i5 o superior y memoria RAM de 8 GB o superior y sistema operativo de 64 bits, esto para garantizar el rendimiento del software que se va a ejecutar simultáneamente. Para este proyecto se utilizó un ordenador con memoria de 16 GB de RAM, Intel Core i7 de 64 bits y disco duro de 300 GB.

### ***Paso a Paso Para Configurar y Lanzar el Core de la Red***

Lo primero es instalar correctamente Ubuntu 16.04 LTS, entonces se procede a hacer la configuración del Core, el primer paso es clonar un repositorio con los elementos que componen el núcleo de red, en la siguiente figura se muestra la línea de código que se debe ejecutar dentro de una terminal de trabajo.



*Clonar repositorio con los elementos que componen el núcleo de red*

```

idtolu@ubuntu-16-04:~$ git clone https://github.com/lfarizav/openair-epc-fed.git
fatal: destination path 'openair-epc-fed' already exists and is not an empty directory.
idtolu@ubuntu-16-04:~$ ls
Desktop      examples.desktop  openair-epc-fed      Pictures      Videos
Documents    Music              openairinterface5g   Public
Downloads    nbiot.conf         openairinterface5g_develop  Templates
idtolu@ubuntu-16-04:~$

```

Nota: Autoría propia

Lo siguiente es acceder a la carpeta clonada, con el comando `cd openair-epc-fed`, luego el comando `git checkout 2021.w06` para crear ramas y actualizar los componentes, y por ultimo se ejecuta el comando `/scripts/syncComponents.sh --hss-branch v1.1.1 --mme-branch 2020.w47 --spgwc-branch v1.1.0 --spgwu-tiny-branch v1.1.0` para sincronizar las entidades hss, mme, spgwc y spgwu – tiny. En la figura 20 se muestra la forma de introducir los comandos.

Acceder a la capeta clonada y verificar versión

```

idtolu@ubuntu-16-04: ~/openair-epc-fed
idtolu@ubuntu-16-04: ~$ cd openair-epc-fed
idtolu@ubuntu-16-04: ~/openair-epc-fed$ git checkout 2021.w06
M       component/oai-hss
M       component/oai-mme
M       component/oai-spgwc
M       component/oai-spgwu-tiny
HEAD is now at e8fefda... DOC: fix typo in docker build context
idtolu@ubuntu-16-04: ~/openair-epc-fed$ ./scripts/syncComponents.sh --hss-branch v1.1.1 --mme-branch 2020.w47 \
> --spgwc-branch v1.1.0 --spgwu-tiny-branch v1.1.0
-----
OAI-HSS      component branch : v1.1.1
OAI-MME      component branch : 2020.w47
OAI-SPGW-C   component branch : v1.1.0
OAI-SPGW-U   component branch : v1.1.0
-----
usage: git submodule [--quiet] add [-b <branch>] [-f|--force] [--name <name>] [--reference <repository>] [--] <reposito
ry> [<path>]
or: git submodule [--quiet] status [--cached] [--recursive] [--] [<path>...]
or: git submodule [--quiet] init [--] [<path>...]
or: git submodule [--quiet] deinit [-f|--force] [--] <path>...
or: git submodule [--quiet] update [--init] [--remote] [-N|--no-fetch] [-f|--force] [--checkout|--merge|--rebase] [-
reference <repository>] [--recursive] [--] [<path>...]
or: git submodule [--quiet] summary [--cached|--files] [--summary-limit <n>] [commit] [--] [<path>...]
or: git submodule [--quiet] foreach [--recursive] <command>
or: git submodule [--quiet] sync [--recursive] [--] [<path>...]
Submodule path 'component/oai-hss': checked out 'did0e45ec749fc49f749c65744084ee09ab1f3e5'
Submodule path 'component/oai-mme': checked out 'did0e45ec749fc49f749c65744084ee09ab1f3e5'
Submodule path 'component/oai-spgwc': checked out '381e881f64986970c2231f2768348c20ed513cc4'
Submodule path 'component/oai-spgwu-tiny': checked out '5ca383967fb80910ce92a596ad2853dd5aa520bb'
~/openair-epc-fed/component/oai-hss ~/openair-epc-fed
Previous HEAD position was did0e45... Merge pull request #62 from OPENAIRINTERFACE/hotfix-issue-59-hss-bad-opc-computat
ion
HEAD is now at 1699b6a... Merge pull request #23 from OPENAIRINTERFACE/release_v1_1_1_candidate
From https://github.com/OPENAIRINTERFACE/openair-hss
 * tag         v1.1.1         -> FETCH_HEAD
Already up-to-date.
~/openair-epc-fed
~/openair-epc-fed/component/oai-mme ~/openair-epc-fed
Previous HEAD position was did0e45... Merge pull request #62 from OPENAIRINTERFACE/hotfix-issue-59-hss-bad-opc-computat
ion

```

Nota: Autoría propia

**Crear una Red Virtual.** Posteriormente, para crear la red de trabajo en el EPC host se ejecutan el siguiente comando `docker network create --attachable --subnet 192.168.61.0/26 --ip-range 192.168.61.0/26 prod-oai-public-net`, luego, comprobar si es posible alcanzar los diferentes nodos con las siguientes líneas de código, `sudo sysctl net.ipv4.conf.all.forwarding=1`, `sudo sysctl net.ipv4.ip_forward=1` y `sudo iptables -P FORWARD ACCEPT`. En la figura 21 se muestra la forma correcta de correr los comandos

## Crear la red de trabajo

```

idtolu@ubuntu-16-04:~/openair-epc-fed$ docker network create --attachable --subnet 192.168.61.0/26 --ip-range 192.168.61.0/26 prod-oai-public-net
Error response from daemon: network with name prod-oai-public-net already exists
idtolu@ubuntu-16-04:~/openair-epc-fed$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
1c87712119bd       bridge             bridge             local
b09b4f00bd6a       host               host               local
357837c62a3b       none              null               local
de6cde43a8e2       prod-oai-public-net bridge             local
idtolu@ubuntu-16-04:~/openair-epc-fed$ sudo sysctl net.ipv4.conf.all.forwarding=1
sudo: unable to resolve host ubuntu-16-04
[sudo] password for idtolu:
net.ipv4.conf.all.forwarding = 1
idtolu@ubuntu-16-04:~/openair-epc-fed$ sudo sysctl net.ipv4.ip_forward=1
sudo: unable to resolve host ubuntu-16-04
net.ipv4.ip_forward = 1
idtolu@ubuntu-16-04:~/openair-epc-fed$ sudo iptables -P FORWARD ACCEPT
sudo: unable to resolve host ubuntu-16-04
idtolu@ubuntu-16-04:~/openair-epc-fed$

```

Nota: Autoría propia

**Crear los Contenedores.** Para crear los contenedores en el ordenador, se ejecutan los comandos que se muestran en la figura 22, así mismo, para correr los container en Docker y posteriormente con el comando *Docker ps -a* se verifica que se hayan creado con su respectivo ID e imagen.

## Ejecutar los contenedores en Docker

```

idtolu@ubuntu-16-04:~/openair-epc-fed$ docker run --name prod-cassandra -d -e CASSANDRA_CLUSTER_NAME="OAI HSS Cluster" \
-e CASSANDRA_ENDPOINT_SNITCH=GossipingPropertyFileSnitch lfzarizav/cassandra:2.1
idtolu@ubuntu-16-04:~/openair-epc-fed$ docker run --privileged --name prod-oai-hss -d --entrypoint /bin/bash lfzarizav/oai-hss:production -c "sleep infinity"
idtolu@ubuntu-16-04:~/openair-epc-fed$ docker network connect prod-oai-public-net prod-oai-hss
idtolu@ubuntu-16-04:~/openair-epc-fed$ docker network connect prod-oai-public-net prod-oai-hss
Error response from daemon: endpoint with name prod-oai-hss already exists in network prod-oai-public-net
idtolu@ubuntu-16-04:~/openair-epc-fed$ docker run --privileged --name prod-oai-mme --network prod-oai-public-net \
-d --entrypoint /bin/bash lfzarizav/oai-mme:production -c "sleep infinity"
idtolu@ubuntu-16-04:~/openair-epc-fed$ docker run --privileged --name prod-oai-spgwc --network prod-oai-public-net \
-d --entrypoint /bin/bash lfzarizav/oai-spgwc:production -c "sleep infinity"
idtolu@ubuntu-16-04:~/openair-epc-fed$ docker run --privileged --name prod-oai-spgwu --network prod-oai-public-net \
-d --entrypoint /bin/bash lfzarizav/oai-spgwu:production -c "sleep infinity"
idtolu@ubuntu-16-04:~/openair-epc-fed$
idtolu@ubuntu-16-04:~/openair-epc-fed$ docker ps
CONTAINER ID        IMAGE                COMMAND                  CREATED        STATUS        PORTS
2d3553456d97      lfzarizav/oai-spgwu:production  "/bin/bash -c 'sleep..." 12 seconds ago Up 11 seconds 2152/udp, 8805/udp
ef603b0cabf7      lfzarizav/oai-spgwc:production  "/bin/bash -c 'sleep..." 23 seconds ago Up 21 seconds 2123/udp, 8805/udp
a315731bf68f      lfzarizav/oai-mme:production    "/bin/bash -c 'sleep..." 42 seconds ago Up 41 seconds 3870/tcp, 2123/udp, 5870/tcp
59c84d25e5d5      lfzarizav/oai-hss:production    "/bin/bash -c 'sleep..." About a minute ago Up About a minute 5868/tcp, 9042/tcp, 9080-9081/tcp
ed5bff0b2f0a      lfzarizav/cassandra:2.1         "docker-entrypoint.s..." About a minute ago Up About a minute 7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp
idtolu@ubuntu-16-04:~/openair-epc-fed$

```

Nota: Autoría propia

**Verificar la Conexión de los Contenedores o Entidades.** Ahora, para verificar la conexión de las entidades se corren las siguientes líneas de código *dockerinspect--*

`format="{{range NetworkSettings.Networks}}{{.IPAddress}}{{end}}"` `prod-oai-mme`, y se hace ping a la dirección `192.168.61.3` y `192.168.61.5` para comprobar una conexión exitosa. En la figura 23 se puede observar la respuesta de conexión exitosa. Para comprobar la conexión de los otros contenedores simplemente al final de la línea de código de debe remplazar el `mme` por el `hss` o el que se desee confirmar.



### Verificar la conexión del MME

```

ldtolu@ubuntu-16-04:~/openair-epc-fed$ docker inspect --format="{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}" prod-oai-mme
192.168.61.3
ldtolu@ubuntu-16-04:~/openair-epc-fed$ ping -c 5 192.168.61.3
PING 192.168.61.3 (192.168.61.3) 56(84) bytes of data:
64 bytes from 192.168.61.3: icmp_seq=1 ttl=64 time=0.079 ms
64 bytes from 192.168.61.3: icmp_seq=2 ttl=64 time=0.034 ms
64 bytes from 192.168.61.3: icmp_seq=3 ttl=64 time=0.036 ms
64 bytes from 192.168.61.3: icmp_seq=4 ttl=64 time=0.039 ms
64 bytes from 192.168.61.3: icmp_seq=5 ttl=64 time=0.039 ms

--- 192.168.61.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.034/0.045/0.079/0.017 ms
ldtolu@ubuntu-16-04:~/openair-epc-fed$ ping -c 5 192.168.61.5
PING 192.168.61.5 (192.168.61.5) 56(84) bytes of data:
64 bytes from 192.168.61.5: icmp_seq=1 ttl=64 time=0.224 ms
64 bytes from 192.168.61.5: icmp_seq=2 ttl=64 time=0.075 ms
64 bytes from 192.168.61.5: icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from 192.168.61.5: icmp_seq=4 ttl=64 time=0.046 ms
64 bytes from 192.168.61.5: icmp_seq=5 ttl=64 time=0.082 ms

```

Nota: Autoría propia

**Utilizar la Base de Datos Cassandra.** Dentro de los contenedores de a los que se accedieron existe un contenedor con base de datos de Cassandra, en la figura 24 se muestra como copiar y ejecutar el container para acceder a la base de datos.



### Acceder a la base de datos Cassandra

```

ldtolu@ubuntu-16-04:~/openair-epc-fed$ docker cp component/oai-hss/src/hss_rel14/db/oai_db.cql prod-cassandra:/home
ldtolu@ubuntu-16-04:~/openair-epc-fed$ docker exec -it prod-cassandra /bin/bash -c "nodetool status"
datacenter: DC1
=====
Status=Up/Down
// State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens   Owns (effective)  Host ID                               Rack
UN 172.17.0.2    51.67 KB     256      100.0%            71700013-babb-44ea-a3be-9ab5f0e24b27  RAC1

ldtolu@ubuntu-16-04:~/openair-epc-fed$ Cassandra_IP=`docker inspect --format="{{range .NetworkSettings.Networks}}{{.
ldtolu@ubuntu-16-04:~/openair-epc-fed$ docker exec -it prod-cassandra /bin/bash -c "cqlsh --file /home/oai_db.cql $
ldtolu@ubuntu-16-04:~/openair-epc-fed$

```

Nota: Autoría propia

**Configuración del HSS.** Para la configuración del HSS (Home Subscriber Server), se deben tomar las siguientes líneas de comandos, con el propósito de ejecutar el container del HSS, asignarle dirección IP y conectarse a todos los protocolos de OpenAirInterface, en figura 25 se puede observar el orden de los comandos y los resultados que debe arrojar para que la base de datos esté actualizada.

### Ejecutar el HSS (Home Subscriber Server)

```

ldto@ubuntu-16-04:~/openair-epc-fed$ HSS_IP=$(docker exec -it prod-oai-hss /bin/bash -c 'ifconfig eth1 | grep inet' | sed -f ./cli-scripts/convertIPAddrFromIfconfig.sed)
ldto@ubuntu-16-04:~/openair-epc-fed$ python3 component/oai-hss/cli-scripts/generateConfigFiles.py --kind-HSS --cassandra=${Cassandra_IP} \
> --hss_s6a=${HSS_IP} --apn1=apn0.openair4G.eur --apn2=apn1.openair4G.eur \
> --users=200 --msdn=208930100001101 --realm=openair4G.eur \
> --ltek=8ba473f2f8fd09487ccbd7097c6862 --op=e734f6734007d6c5ce7a0508809e79c \
> --nb-ssid=$(cat /dev/urandom | tr -dc 'a-z0-9' | fold -w 10 | xargs echo | sha256sum | cut -d' ' -f1)
ldto@ubuntu-16-04:~/openair-epc-fed$ docker cp ./hss-cfg.sh prod-oai-hss:/openair-hss/scripts
ldto@ubuntu-16-04:~/openair-epc-fed$ docker exec -it prod-oai-hss /bin/bash -c 'cd /openair-hss/scripts && chmod 777 hss-cfg.sh && ./hss-cfg.sh'
INFO:root:208930100001170 70 41 8ba473f2f8fd09487ccbd7097c6862 mme.openair4G.eur 3 openair4G.eur 2683b376d1056746de3b254012908e0e 96 {"Subscription-Data":{"Access-Restriction-Data":{"Subscriber-Status":0,"Network-Access-Mode":2,"Regional-Subscription-Zone-Code":["0x0123","0x4567","0x89AB","0xCDEF","0x1234","0x5678","0x9ABC","0xDEF0","0x2345","0x6789"],"MSISDN":"0x10","AMBR":{"Max-Requested-Bandwidth-UL":50000000,"Max-Requested-Bandwidth-DL":100000000},"APN-Configuration-Profile":{"Context-Identifier":0,"All-APN-Configurations-Included-Indicator":0,"APN-Configuration":{"Context-Identifier":0,"PDN-Type":0,"Service-Selection":"apn0.openair4G.eur","EPS-Subscribed-QoS-Profile":{"QoS-Class-Identifier":9,"Allocation-Retention-Priority":{"Priority-Level":15,"Pre-emption-Capability":0,"Pre-emption-Vulnerability":0},"AMBR":{"Max-Requested-Bandwidth-UL":50000000,"Max-Requested-Bandwidth-DL":100000000},"PDN-GW-Allocation-Type":0,"MIP6-Agent-Info":{"MIP-Home-Agent-Address":["172.26.17.183"]},"Subscribed-Periodic-RAU-TAU-Timer":0}}},"Service-Selection":"apn1.openair4G.eur","EPS-Subscribed-QoS-Profile":{"QoS-Class-Identifier":9,"Allocation-Retention-Priority":{"Priority-Level":13,"Pre-emption-Capability":1,"Pre-emption-Vulnerability":0},"AMBR":{"Max-Requested-Bandwidth-UL":50000000,"Max-Requested-Bandwidth-DL":100000000},"PDN-GW-Allocation-Type":0,"MIP6-Agent-Info":{"MIP-Home-Agent-Address":["172.26.17.183"]},"Subscribed-Periodic-RAU-TAU-Timer":0}}}}
INFO:root:208930100001290 190 41 8ba473f2f8fd09487ccbd7097c6862 mme.openair4G.eur 3 openair4G.eur 2683b376d1056746de3b254012908e0e 96 {"Subscription-Data":{"Access-Restriction-Data":{"Subscriber-Status":0,"Network-Access-Mode":2,"Regional-Subscription-Zone-Code":["0x0123","0x4567","0x89AB","0xCDEF","0x1234","0x5678","0x9ABC","0xDEF0","0x2345","0x6789"],"MSISDN":"0x190","AMBR":{"Max-Requested-Bandwidth-UL":50000000,"Max-Requested-Bandwidth-DL":100000000},"APN-Configuration-Profile":{"Context-Identifier":0,"All-APN-Configurations-Included-Indicator":0,"APN-Configuration":{"Context-Identifier":0,"PDN-Type":0,"Service-Selection":"apn0.openair4G.eur","EPS-Subscribed-QoS-Profile":{"QoS-Class-Identifier":9,"Allocation-Retention-Priority":{"Priority-Level":15,"Pre-emption-Capability":0,"Pre-emption-Vulnerability":0},"AMBR":{"Max-Requested-Bandwidth-UL":50000000,"Max-Requested-Bandwidth-DL":100000000},"PDN-GW-Allocation-Type":0,"MIP6-Agent-Info":{"MIP-Home-Agent-Address":["172.26.17.183"]},"Subscribed-Periodic-RAU-TAU-Timer":0}}},"Service-Selection":"apn1.openair4G.eur","EPS-Subscribed-QoS-Profile":{"QoS-Class-Identifier":9,"Allocation-Retention-Priority":{"Priority-Level":13,"Pre-emption-Capability":1,"Pre-emption-Vulnerability":0},"AMBR":{"Max-Requested-Bandwidth-UL":50000000,"Max-Requested-Bandwidth-DL":100000000},"PDN-GW-Allocation-Type":0,"MIP6-Agent-Info":{"MIP-Home-Agent-Address":["172.26.17.183"]},"Subscribed-Periodic-RAU-TAU-Timer":0}}}}

```

Nota: Autoría propia

**Verificar Datos de Cassandra.** Para verificar que los datos de Cassandra están correctamente almacenados se introducen los comandos que se pueden observar en la figura 26, como resultado el sistema debe arrojar la tabla que se visualiza, que al final son los datos de los usuarios remotos.

## Verificar base de datos cassandra

```

root@edsbfff0b2f0a: /
'cacert.pem' -> '/openair-hss/etc/cacert.pem'
'hss.key.pem' -> '/openair-hss/etc/hss.key.pem'
ldtolu@ubuntu-16-04:~/openair-epc-fed$
ldtolu@ubuntu-16-04:~/openair-epc-fed$ docker exec -it prod-cassandra /bin/bash
root@edsbfff0b2f0a:/# cqlsh
Connected to OAI HSS Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 2.1.22 | CQL spec 3.2.1 | Native protocol v3]
Use HELP for help.
cqlsh> cqlsh> SELECT inst, mssidn, access_restriction, key, mmehost, opc, mmerealn, sqn FROM vhss.users_inst;
SyntaxException: <ErrorMessage code=2000 [syntax error in CQL query] message="line 1:0 no viable alternative at input 'cqlsh' ([cqlsh]...)"/>
cqlsh> SELECT inst, mssidn, access_restriction, key, mmehost, opc, mmerealn, sqn FROM vhss.users_inst;

```

inst	mssidn	access_restriction	key	mmehost	opc	mmerealn	sqn
208930100001246	146	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001290	190	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001170	70	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001261	161	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001276	176	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001120	20	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001209	109	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001289	189	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001254	154	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001105	5	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001160	60	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001150	50	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001224	124	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001286	186	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001182	82	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001137	37	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001128	28	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001203	103	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001282	182	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001133	33	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001217	117	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001157	57	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001267	167	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001106	6	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001284	184	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001269	169	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001158	58	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001296	196	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001107	7	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001226	126	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001187	87	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001149	49	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96
208930100001104	4	41	8baf473f2f8fd09487cccbd7097c6862	mme.openair4G.eur	8e27b6af0e692e750f32667a3b14605d	openair4G.eur	96

Nota: Autoría propia

**Configuración del MME, SPGWC y SPGW-U.** El proceso para la configuración de cada una de las entidades que componen el Core es muy parecido, a continuación, se detalla en la figura 27 la forma correcta y el orden de los comandos para direccionar y configurar el MME (Mobility Management Entity).

## Configuración del MME

```

idtolu@ubuntu-16-04: ~/openair-epc-fed
idtolu@ubuntu-16-04:~/openair-epc-fed$
idtolu@ubuntu-16-04:~/openair-epc-fed$
idtolu@ubuntu-16-04:~/openair-epc-fed$ MME_IP=$(docker inspect --format="{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}" prod-oai-mme)
idtolu@ubuntu-16-04:~/openair-epc-fed$ SPGW_IP=$(docker inspect --format="{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}" prod-oai-spgwc)
idtolu@ubuntu-16-04:~/openair-epc-fed$ python3 component/oai-spgwc/ci-scripts/generateConfigFiles.py --kind=MME \
> -hss_s0a=${HSS_IP} --mme_s0a=${MME_IP} \
> -mme_s1c_IP=${MME_IP} --mme_s1c_name=eth0 \
> -mme_s10_IP=${MME_IP} --mme_s10_name=lo \
> -mme_s11_IP=${MME_IP} --mme_s11_name=eth0 --spgwc0_s11_IP=${SPGW_IP} --realn=openair4G.eur \
> --from_docker_file
idtolu@ubuntu-16-04:~/openair-epc-fed$ docker cp ./mme-cfg.sh prod-oai-mme:/openair-mme/scripts
idtolu@ubuntu-16-04:~/openair-epc-fed$ docker exec -it prod-oai-mme /bin/bash -c "cd /openair-mme/scripts && chmod 777 mme-cfg.sh && ./mme-cfg.sh"
sed: no input files
Tried to replace @TAC-HB_MME_1@ with 02
sed: no input files
Tried to replace @REALM@ with openair4G.eur
sed: no input files
Tried to replace @MME_CODE@ with 3
sed: no input files
Tried to replace @TAC_1@ with 001
sed: no input files
Tried to replace @TAC-LB_SGW_TEST_0@ with 07
sed: no input files
Tried to replace @MCC@ with 208
sed: no input files
Tried to replace @HSS_FQDN@ with hss.openair4G.eur
sed: no input files
Tried to replace @MNC3_SGW_0@ with 093
sed: no input files
Tried to replace @MNC3_MME_0@ with 093
sed: no input files
Tried to replace @MME_IPV4_ADDRESS_FOR_S11@ with 192.168.61.3/24
sed: no input files
Tried to replace @TAC_0@ with 600
sed: no input files
Tried to replace @MCC_MME_0@ with 208
sed: no input files
Tried to replace @TAC-LB_SGW_0@ with 58
sed: no input files
Tried to replace @MNC@ with 93
sed: no input files
Tried to replace @MNC3_MME_1@ with 093
sed: no input files
Tried to replace @TAC-LB_MME_0@ with 59
sed: no input files
Tried to replace @OUTPUT@ with CONSOLE

```

Nota: Autoría propia

Los comandos que se deben ejecutar para correr el SPGWC son los siguientes, primero `python3 component/oai-spgwc/ci-scripts/generateConfigFiles.py --kind=SPGW-C --s11c=eth0 --sxc=eth0 --apn=apn0.openair4G.eur --dns1_ip=192.168.61.1 --dns2_ip=192.168.61.1 --from_docker_file`, para determinar la ruta de la cual la entidad toma toda la información, luego se ejecuta `docker cp ./spgwc-cfg.sh prod-oai-spgwc:/openair-spgwc`, para copiar archivos en la carpeta del contenedor, y por ultimo `docker exec -it prod-oai-spgwc /bin/bash -c "cd /openair-spgwc && chmod 777 spgwc-cfg.sh && ./spgwc-cfg.sh"`, para ejecutar el contenedor y todos sus archivos, los resultados se pueden ver en la figura 28.

## Configuración del SPGWC

```
ldt@ubuntu-16-04:~/openair-epc-fed$ python3 component/oai-spgwc/ci-scripts/generateConfigFiles.py --kind=SPGW-C --s11c=eth0 --sxc=eth0 --apn=apn0.openair4G.eur
--dns1_ip=192.168.61.1 --dns2_ip=192.168.61.1 --from_docker_file
ldt@ubuntu-16-04:~/openair-epc-fed$ docker cp ./spgwc-cfg.sh prod-oai-spgwc:/openair-spgwc
ldt@ubuntu-16-04:~/openair-epc-fed$ docker exec -it prod-oai-spgwc /bin/bash -c "cd /openair-spgwc && chmod 777 spgwc-cfg.sh && ./spgwc-cfg.sh"
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
ldt@ubuntu-16-04:~/openair-epc-fed$ docker exec -it prod-oai-spgwc /bin/bash -c "ifconfig lo:s5c 127.0.0.15 up && ifconfig lo:p5c 127.0.0.16 up"
ldt@ubuntu-16-04:~/openair-epc-fed$
```

Nota: Autoría propia

Como se puede observar en la siguiente figura 29 los comandos de configuración del SPGW-U son similares a los que se explicaron en el párrafo anterior

## Configuración del SPGW-U

```
ldt@ubuntu-16-04:~/openair-epc-fed$ python3 component/oai-spgwu-tiny/ci-scripts/generateConfigFiles.py --kind=SPGW-U \
> --sxc ip addr=${SPGW0 IP} --sxc=eth0 --siu=eth0 --from_docker_file
ldt@ubuntu-16-04:~/openair-epc-fed$ docker cp ./spgwu-cfg.sh prod-oai-spgwu:/openair-spgwu-tiny
ldt@ubuntu-16-04:~/openair-epc-fed$ docker exec -it prod-oai-spgwu /bin/bash -c "cd /openair-spgwu-tiny && chmod 777 spgwu-cfg.sh && ./spgwu-cfg.sh"
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
sed: no input files
ldt@ubuntu-16-04:~/openair-epc-fed$
```

Nota: Autoría propia

**Lanzamiento de las Entidades del EPC.** Este es el último paso en la configuración del CORE, básicamente lo que se hace es lanzar cada una de las entidades mencionadas en este apartado. La primera entidad en lanzarse es la base de datos o HSS mediante el comando:

```
docker exec -it prod-oai-hss /bin/bash
```

```
./bin/oai_hss -j ./etc/hss_rel14.json --reloadkey true
```

En la figura 30, se puede observar en la ventana una terminal abierta con dos pestañas abiertas, la primera pestaña obedece al HHS en funcionamiento y la segunda pestaña es el MME en ejecución, en este sentido, cada una de las entidades deben ejecutarse en una pestaña independiente, esto para que todas corran al mismo tiempo y se pueda unificar el trabajo del CORE.

### MME en ejecución

```

Terminal
root@202c974bb219: /openair-mme
root@b945ebd38777: /openair-hss
root@202c974bb219: /openair-mme

000099 00000:353253 7FASFF1042C0 INFO CONFIG air-mme/src/mme_app/mme_config.c:1663 XML log level..... INVALID_LOG_LEVEL (empty queue of messages)
Creating queue of message of size 128
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 256
Creating queue of message of size 1024
Creating queue of message of size 256
000100 00000:364756 7FASFF1042C0 DEBUG NAS r-mme/src/nas/emm/nas_emm_task.c:0167 Initializing NAS EMM task interface
000101 00000:365320 7FASFF1042C0 TRACE NAS-EM enair-mme/src/nas/emm/emm_main.c:0081 Entering emm_main_initialize()
000102 00000:366025 7FASFF1042C0 TRACE NAS lr-mme/src/nas/apl/mme/mme_apl.c:0113 Entering mme_apl_get_emm_config()
000103 00000:366629 7FASFF1042C0 TRACE NAS lr-mme/src/nas/apl/mme/mme_apl.c:0345 Leaving mme_apl_get_emm_config() (rc=0)
000104 00000:366876 7FASFF1042C0 TRACE NAS-EM enair-mme/src/nas/emm/emm_main.c:0102 Leaving emm_main_initialize()
000105 00000:366918 7FASFF1042C0 DEBUG NAS r-mme/src/nas/emm/nas_emm_task.c:0176 Initializing NAS EMM task interface: DONE
000106 00000:366923 7FASFF1042C0 DEBUG NAS r-mme/src/nas/esm/nas_esm_task.c:0150 Initializing NAS ESM task interface
000107 00000:366928 7FASFF1042C0 TRACE NAS-ES enair-mme/src/nas/esm/esm_main.c:0084 Entering esm_main_initialize()
000108 00000:366931 7FASFF1042C0 TRACE NAS-ES enair-mme/src/nas/esm/esm_main.c:0089 Leaving esm_main_initialize()
000109 00000:366937 7FASFF1042C0 DEBUG NAS r-mme/src/nas/esm/nas_esm_task.c:0139 Initializing NAS ESM task interface: DONE
000110 00000:367811 7FASFF1042C0 DEBUG SCTP rc/sctp/sctp_primitives_server.c:0747 Initializing SCTP task interface
000111 00000:367850 7FASFF1042C0 DEBUG SCTP rc/sctp/sctp_primitives_server.c:0761 Initializing SCTP task interface: DONE
000112 00000:367857 7FASFF1042C0 DEBUG UDP /src/udp/udp_primitives_server.c:0535 Initializing UDP task interface
000113 00000:367865 7FASFF1042C0 DEBUG UDP /src/udp/udp_primitives_server.c:0543 Initializing UDP task interface: DONE
000114 00000:367893 7FASFF1042C0 DEBUG S11 enair-mme/src/s11/s11_mme_task.c:0367 Initializing S11 interface
000115 00000:368577 7FASFF1042C0 DEBUG S11 enair-mme/src/s11/s11_mme_task.c:0346 Tx UDP_INIT IP addr 192.168.61.3:2123
000116 00000:368593 7FASFF1042C0 DEBUG S11 enair-mme/src/s11/s11_mme_task.c:0346 Tx UDP_INIT IP addr 192.168.61.3:0
000117 00000:368652 7FAS27FC700 DEBUG UDP /src/udp/udp_primitives_server.c:0239 Creating new listen socket on IPv4 address 192.168.61.3 and port 2123
000118 00000:368669 7FAS27FC700 DEBUG UDP /src/udp/udp_primitives_server.c:0256 Listened on port 2123 for IPv4.
000119 00000:368674 7FAS27FC700 DEBUG UDP /src/udp/udp_primitives_server.c:0279 (IPv4) Inserting new descriptor for task 9, sd 38
000120 00000:368687 7FAS27FC700 DEBUG UDP /src/udp/udp_primitives_server.c:0122 Received 1 events
000121 00000:368703 7FAS27FC700 DEBUG UDP /src/udp/udp_primitives_server.c:0239 Creating new listen socket on IPv4 address 192.168.61.3 and port 0

```

### Autoría propia

Ahora bien, en la figura 31 se puede observar el pantallazo de las cuatro entidades trabajando simultáneamente, en la pestaña actual de la captura se visualiza la forma de lanzar el SPGWU y su respectivo funcionamiento y en las otras pestañas, están en ejecución el HSS, MME, SPWC en orden de izquierda a derecha respectivamente:

Todo el CORE ejecución

```

root@19f5d20eca3:/openair-spgwu-tiny# ./bin/oai_spgwu -o -c ./etc/spgw_u.conf
[2022-03-17T06:21:35.865143] [spgwu] [spgwu_app] [start] Options parsed
[2022-03-17T06:21:35.865720] [spgwu] [spgwu_app] [info] SettingNotFoundException : SPGW-U.ITTI_TASKS, using defaults
[2022-03-17T06:21:35.865772] [spgwu] [spgwu_app] [info] THREAD_POOL_SIZE : 1
[2022-03-17T06:21:35.865795] [spgwu] [spgwu_app] [info] THREAD_POOL_SIZE : 1
[2022-03-17T06:21:35.865816] [spgwu] [spgwu_app] [info] THREAD_POOL_SIZE : 1
[2022-03-17T06:21:35.865836] [spgwu] [spgwu_app] [info] ==== EURECOM SPGW-U vbranch: master Abrev. Hash: 2914188 Date: Thu Mar 11 19:50:30 2021 +0000 ====
[2022-03-17T06:21:35.865839] [spgwu] [spgwu_app] [info] Configuration:
[2022-03-17T06:21:35.865840] [spgwu] [spgwu_app] [info] - Instance .....: 0
[2022-03-17T06:21:35.865842] [spgwu] [spgwu_app] [info] - PID dir .....: /var/run
[2022-03-17T06:21:35.865844] [spgwu] [spgwu_app] [info] - ITTI tasks:
[2022-03-17T06:21:35.865845] [spgwu] [spgwu_app] [info]   ITTI Timer task:
[2022-03-17T06:21:35.865847] [spgwu] [spgwu_app] [info]     CPU ID .....: 0
[2022-03-17T06:21:35.865850] [spgwu] [spgwu_app] [info]     sched policy....: 1
[2022-03-17T06:21:35.865852] [spgwu] [spgwu_app] [info]     sched priority...: 85
[2022-03-17T06:21:35.865855] [spgwu] [spgwu_app] [info]   SPGW-SIU task:
[2022-03-17T06:21:35.865858] [spgwu] [spgwu_app] [info]     CPU ID .....: 0
[2022-03-17T06:21:35.865861] [spgwu] [spgwu_app] [info]     sched policy....: 1
[2022-03-17T06:21:35.865864] [spgwu] [spgwu_app] [info]     sched priority...: 84
[2022-03-17T06:21:35.865867] [spgwu] [spgwu_app] [info]   SPGW-SX task:
[2022-03-17T06:21:35.865868] [spgwu] [spgwu_app] [info]     CPU ID .....: 0
[2022-03-17T06:21:35.865870] [spgwu] [spgwu_app] [info]     sched policy....: 1
[2022-03-17T06:21:35.865873] [spgwu] [spgwu_app] [info]     sched priority...: 84
[2022-03-17T06:21:35.865876] [spgwu] [spgwu_app] [info]   SPGW_APP task:
[2022-03-17T06:21:35.865881] [spgwu] [spgwu_app] [info]     CPU ID .....: 0
[2022-03-17T06:21:35.865885] [spgwu] [spgwu_app] [info]     sched policy....: 1
[2022-03-17T06:21:35.865890] [spgwu] [spgwu_app] [info]     sched priority...: 84
[2022-03-17T06:21:35.865893] [spgwu] [spgwu_app] [info]   ASYNC_SHELL_CMD task:
[2022-03-17T06:21:35.865895] [spgwu] [spgwu_app] [info]     CPU ID .....: 0
[2022-03-17T06:21:35.865897] [spgwu] [spgwu_app] [info]     sched policy....: 1
[2022-03-17T06:21:35.865899] [spgwu] [spgwu_app] [info]     sched priority...: 84
[2022-03-17T06:21:35.865902] [spgwu] [spgwu_app] [info] - SIU_512_54:
[2022-03-17T06:21:35.865903] [spgwu] [spgwu_app] [info]   iface .....: eth0
[2022-03-17T06:21:35.865905] [spgwu] [spgwu_app] [info]   ipw4_addr .....: 192.168.61.5
[2022-03-17T06:21:35.865908] [spgwu] [spgwu_app] [info]   ipw4_mask .....: 255.255.255.192

```

Nota: Autoría propia

### Lanzamiento del eNodeB Utilizando OpenAirInterface y la USRP B200 Mini.

Para la simulación del eNodeB se requieren dos ambientes, el primero es continuar con la ejecución del CORE que se realizó en la anterior sección, allí mismo se debe utilizar otro repositorio de OpenAirInterface para lanzar el software del eNodeB, el segundo elemento es conectar la USRP al ordenador donde se está realizando todo el ejercicio de OpenAirInterface, para ello, se utilizó una USRP (Universal Software Radio Peripheral) B200 Mini, más conocida en español como Dispositivo de Radio Definido por Software, este tipo de periféricos de alto costo proporcionan una arquitectura de RF (Radio Frecuencia) para crear prototipos e implementar sistemas inalámbricos de comunicación con procesamiento de señales. (Instruments, 2022). En la figura 32 se puede detallar físicamente una USRP B200 Mini apropiada para utilizar en este proyecto.



El último comando de la figura 33 se utiliza para llamar un archivo y editarlo, lo que se puede ver en la siguiente figura es el archivo en texto plano, al cual se le realizan ciertas configuraciones en los nombres de las interfaces y en el direccionamiento IP para su correcto funcionamiento. El archivo de configuración se puede visualizar en la figura 34.



### Editar archivo de configuración

```

Text Editor
root@4f2fc2a3547b: /openair-hss
cna...
Please, commit your changes or stash them before you
Aborting
root@ubuntu-16-04:~/openairinterface5g$ git branch
develop-nb-1st
master
root@ubuntu-16-04:~/openairinterface5g$ cd
root@ubuntu-16-04:~/openairinterface5g$ cd openairinterface5g
root@ubuntu-16-04:~/openairinterface5g$ cd
root@ubuntu-16-04:~/openairinterface5g$ cd cna...
root@ubuntu-16-04:~/openairinterface5g/cna...$ cat commands build_oai CMakeLists.txt epc_test lte_...
root@ubuntu-16-04:~/openairinterface5g/cna...$ cd
root@ubuntu-16-04:~/openairinterface5g/cna...$ build CMakeLists.txt
root@ubuntu-16-04:~/openairinterface5g/cna...$ cd ..
root@ubuntu-16-04:~/openairinterface5g/cna...$ git checkout develop-nb
root@ubuntu-16-04:~/openairinterface5g/cna...$ cd cna...
root@ubuntu-16-04:~/openairinterface5g/cna...$ ./build_oai -I --eNB -w

Archivo de configuración
El archivo de configuración se encuentra en:
EPC/CONF/enb.band28.tm1.25PRB.usrpb210.conf

*enb.band28.tm1.25PRB.usrpb210.conf (~/openairinterface5g/targets/PROJECTS/GENERIC-LTE-EPC/CONF) - gedit
# timer_reordering = (ms) [0,5, ... 250, 300, 350, ... ,500]
timer_status_prohibit = 0;

# poll_pdu = [4, 8, 16, 32 , 64, 128, 256, infinity(>10000)]
poll_pdu = 4;

# poll_byte = (kB) [25,50,75,100,125,250,375,500,750,1000,1250,1500,2000,3000,infinity(>10000)]
poll_byte = 99999;

# max_retx_threshold = [1, 2, 3, 4 , 6, 8, 16, 32]
max_retx_threshold = 4;
}

# ----- SCTP definitions
SCTP :
{
# Number of streams to use in input/output
SCTP_INSTREAMS = 2;
SCTP_OUTSTREAMS = 2;
};

////////// MME parameters:
mme_ip_address = ( { ipv4 = "192.168.61.03";
                    ipv6 = "192:168:30:17";
                    active = "yes";
                    preference = "ipv4";
                    }
);

NETWORK_INTERFACES :
{
ENB_INTERFACE_NAME_FOR_S1_MME = "enp3s0";
ENB_IPV4_ADDRESS_FOR_S1_MME = "192.168.12.242/24";
ENB_INTERFACE_NAME_FOR_S1U = "enp3s0";
ENB_IPV4_ADDRESS_FOR_S1U = "192.168.12.242/24";
ENB_PORT_FOR_S1U = 2152; # Spec 2152
};

```

Nota: Autoría propia

**Ejecución de eNodeB.** Luego de hacer los cambios necesarios en el archivo de configuración, se procede hacer la ejecución del eNodeB con los siguientes comandos: `cd ~/openairinterface5g/cmake_targets /lte_build_oai/build, y $ sudo ./lte-softmodem -O ~/openairinterface5g/targets/PROJECTS/GENERIC-LTE-PC/CONF/enb.band28.tm1.25PRB.usrpb210.conf`. En la figura 35 se puede apreciar la ejecución de dichos comandos en una terminal extra a las utilizadas por las entidades de core.

## Ejecución de eNodeB

```

root@ubuntu-16-04:~/openairinterface5g/cnake_targets/lte_build_out/build$ sudo ./lte-softnode -O ~/openairinterface5g/targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band28.tn1.25PRB.usrpb210.conf
# /dev/cpu_dma_latency set to 0us
log init done
num component carrier 1
WARNING: setting 'gtpu_log_level' not found in configuration file
WARNING: setting 'gtpu_log_verbosity' not found in configuration file
WARNING: setting 'udp_log_level' not found in configuration file
WARNING: setting 'udp_log_verbosity' not found in configuration file
WARNING: setting 'osa_log_level' not found in configuration file
WARNING: setting 'osa_log_verbosity' not found in configuration file
Downlink Frequency/ uplink offset of CC_id 0 set to 780000000/-55000000
configuring for eNB
CPU Freq ts 3.591932
reported resolution = 1 ns
[HW][I]Version: Branch: develop-nb-lot Abrev. Hash: 677bbab Date: Mon Nov 30 13:37:48 2020 +0100
[PHY][I]Initializing frame parms for N_RB_DL 25, Ncp 0, osf 1
lte_parms.cc: Setting N_RB_DL to 25, ofdm_symbol_size 512
[PHY][I]Initializing frame parms for N_RB_DL 25, Ncp 0, osf 1
lte_parms.cc: Setting N_RB_DL to 25, ofdm_symbol_size 512, PHICH Resource 1, PHICH Duration 0
[openair][LTE_PHY][INIT] common_vars->txdata[0][0] = 0x7ff37d55040 (286720 bytes)
[openair][LTE_PHY][INIT] common_vars->txdata[0][5] = 0x7ff37d5e040 (286720 bytes)
[openair][LTE_PHY][INIT] common_vars->txdata[0][10] = 0x7ff37d562040 (307200 bytes)
[openair][LTE_PHY][INIT] common_vars->txdata[0][15] = 0x7ff37d56040 (307200 bytes)
[openair][LTE_PHY][INIT] common_vars->txdata[0][20] = 0x1738e500 (61440 bytes)
[openair][LTE_PHY][INIT] common_vars->txdata[0][25] = 0x1738e500 (61440 bytes)

```

Nota: Autoría propia

Finalmente, en la figura 36 se puede observar todos los elementos del core y el eNodeB funcionando simultáneamente en el mismo ordenador, la pestaña que se muestra a continuación es la pestaña del MME, como se aprecia en el título de la misma, en la parte inferior derecha se puede ver una estadística que muestra los dispositivos que en ese momento está gestionando el core como parte de una red móvil, allí se aprecia que existe un eNB conectado y que se encuentra en funcionamiento, el eNB está representado por la USRP B200 Mini.

## Estadística de conexión en el MME

```

root@c806c6d85d6:/openair-mme$
002329 02501:551170 7F840AFFD700 DEBUG SIAP mme/src/siap/siap_mme_handlers.c:02826 Create eNB context for assoc_id: 2
002330 02501:551240 7F83967FC700 DEBUG SCTP rc/sctp/sctp_primitives_server.c:0547 [2][47] Msg of length 59 received from port 36412, on stream 0, PPID 18
002331 02501:551257 7F83967FC700 DEBUG SCTP rc/sctp/sctp_primitives_server.c:0554 SCTP RETURNING!!
002332 02501:552525 7F840AFFD700 DEBUG SIAP mme/src/siap/siap_mme_handlers.c:0301 SI-SetupRequest macroENB_ID.size 3 (should be 20)
002333 02501:552520 7F840AFFD700 DEBUG SIAP mme/src/siap/siap_mme_handlers.c:0301 New SI setup request incoming from macro eNB id: 00e00
002334 02501:552542 7F840AFFD700 DEBUG SIAP mme/src/siap/siap_mme_handlers.c:0423 Adding eNB to the list of served eNBs
002335 02501:552547 7F840AFFD700 DEBUG SIAP mme/src/siap/siap_mme_handlers.c:0438 Adding eNB id 3584 to the list of served eNBs
002336 02501:553048 7F84100C8700 DEBUG SCTP rc/sctp/sctp_primitives_server.c:0283 [47][2] Sending buffer 0x7fb374009f90 of 27 bytes on stream 0 with ppid 18
002337 02501:553097 7F84100C8700 DEBUG SCTP rc/sctp/sctp_primitives_server.c:0286
002338 02510:816458 7F840A7FC700 DEBUG MME-AP src/mme_app/mme_app_statistics.c:0039
===== MME STATISTICS =====
Current Status: Added since last display: Removed since last display:
Connected eNBs: 1 | 1 | 0
Attached UEs: 0 | 0 | 0
Connected UEs: 0 | 0 | 0
Default Bearers: 0 | 0 | 0
SI-U Bearers: 0 | 0 | 0
=====
002345 02510:816503 7F840A7FC700 DEBUG MME-AP src/mme_app/mme_app_statistics.c:0075
002346 02511:276871 7F83967FC700 DEBUG SCTP rc/sctp/sctp_primitives_server.c:0458 SCTP_SHUTDOWN_EVENT received
002347 02511:276889 7F83967FC700 DEBUG SCTP rc/sctp/sctp_primitives_server.c:0562 Sending close connection for assoc_id 2
002348 02511:277810 7F840AFFD700 INFO SIAP mme/src/siap/siap_mme_handlers.c:2772 Removing eNB with association id 2

```

Nota: Autoría propia

En anterior figura también se aprecia que no hay UE (User Equipment) conectados a la red móvil, esto debido a que dentro de los proyectos de la Alianza de Software OpenAirInterface no se ha desarrollado compatibilidad entre el core y los módulos BC95-G de Quectel u otros similares el mercado, significa que se tiene toda la infraestructura de red para conectar dispositivos, pero la tecnología que maneja el core no tiene la capacidad de asignar direcciones IP a los User Equipment que desean conectarse a la red móvil.

## Simulación del Usuario Final en GNURADIO

En este capítulo se explica detalladamente la simulación del usuario final en GNURadio, para tal fin se utilizó lo planteado en el diagrama de bloques propuesto en la sección Diseño de un sistema de transmisión OFDM de este documento, teniendo en cuenta que en una red móvil 4G o 5G lo ideal es trabajar bajo OFDM.

Adicionalmente, se utilizaron otros bloques para medir algunos indicadores de calidad KPI (Key Performance Indicator), como por ejemplo el bloque *probe rate* para medir el throughput y el bloque *BER* para medir la tasa de error de bit de las simulaciones realizadas. Para ello, se tomó como dataset un archivo que contiene 9216 bytes, cantidad de datos igual al que arroja el sensor PMS5003 tomado como referencia para estas simulaciones.

Por último, para de efectuar las simulaciones, es necesario precisar la frecuencia de las mediciones, el tamaño y procedencia del paquete de datos, así mismo, es conveniente recordar que con este proyecto se propone realizar mediciones cada cinco minutos en cada sensor, para cumplir el objetivo de obtener datos de alta confiabilidad.

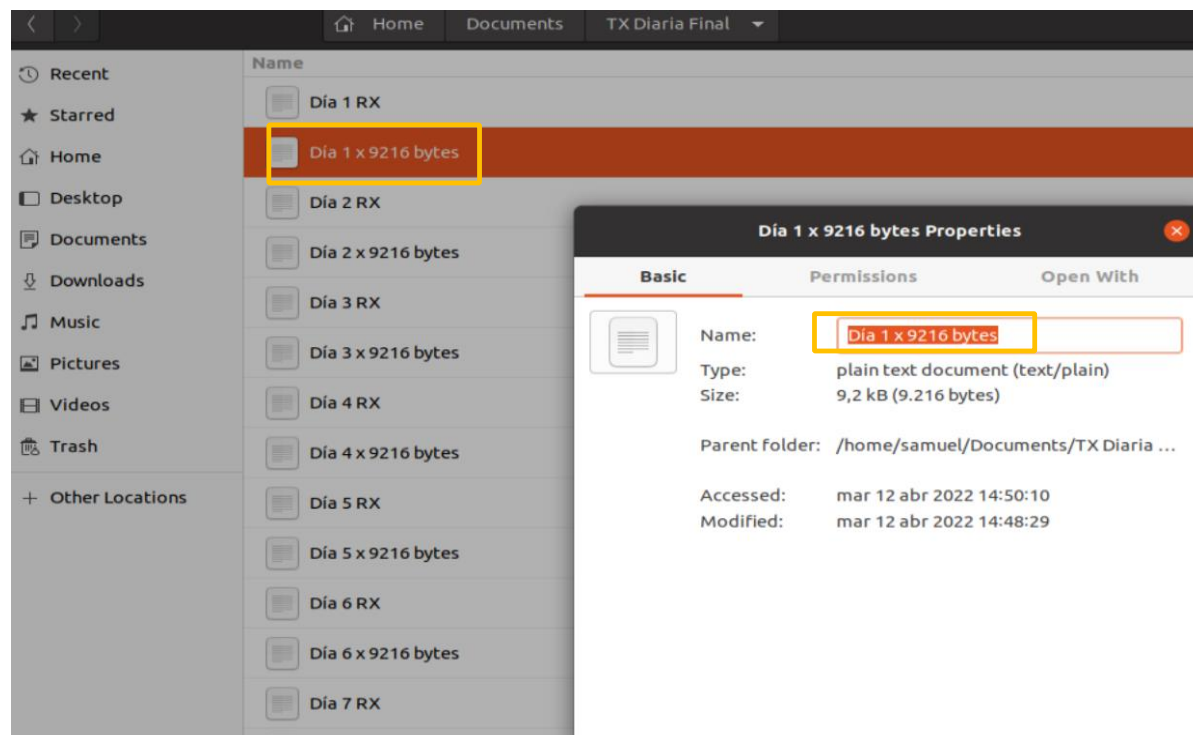
### Simulación en OFDM para un Sensor PMS5003

Según el datasheet de este sensor, cada medición efectuada tiene una longitud fija de paquete de 32 bytes, entonces si se realizan mediciones cada cinco (5) minutos, en una hora son doce (12) mediciones y en un día doscientas ochenta y ocho (288), al multiplicar 288 por 32 bytes, se tiene un total de 9.216 bytes. ( $32 \text{ bytes} * 288 \text{ mediciones} = 9216 \text{ bytes}$ ).

Con base en lo anterior, se define que los paquetes a transmitir a través de OFDM tienen una longitud fija de 9216 bytes, en la siguiente figura se verifica que el paquete cumpla con esta condición.



*Paquete de 9216 bytes listo para transmitir*



Nota: Autoría propia

Tal como se ha mencionado, el método de multiplexación OFDM es ideal para transmitir datos en 4G o 5G NB-IoT. Las simulaciones realizadas en el programa GNU radio indican un sistema de transmisión OFDM, para tal fin, se tomaron como parámetros de transmisión valores recomendados en la wiki de GNU radio, tal como se muestra en la siguiente figura, se definieron los bloques de las variables *fft\_len* con valor de 64, la variable *samp\_rate* con valor de 100k, la variable *packet\_len* con valor de 96 bytes, la variable *header\_mod* en BPSK y la variable *payload\_mod* en QPSK.

### Parámetros de transmisión en GNU radio para OFDM

<b>Variable</b> <b>Id:</b> fft_len <b>Value:</b> 64	<b>Variable</b> <b>Id:</b> samp_rate <b>Value:</b> 100k	<b>Variable</b> <b>Id:</b> length_tag_key <b>Value:</b> packet_len
<b>Variable</b> <b>Id:</b> packet_len <b>Value:</b> 96	<b>Variable</b> <b>Id:</b> header_mod <b>Value:</b> <constellation BPSK>	<b>Variable</b> <b>Id:</b> payload_mod <b>Value:</b> <constellation QPSK>

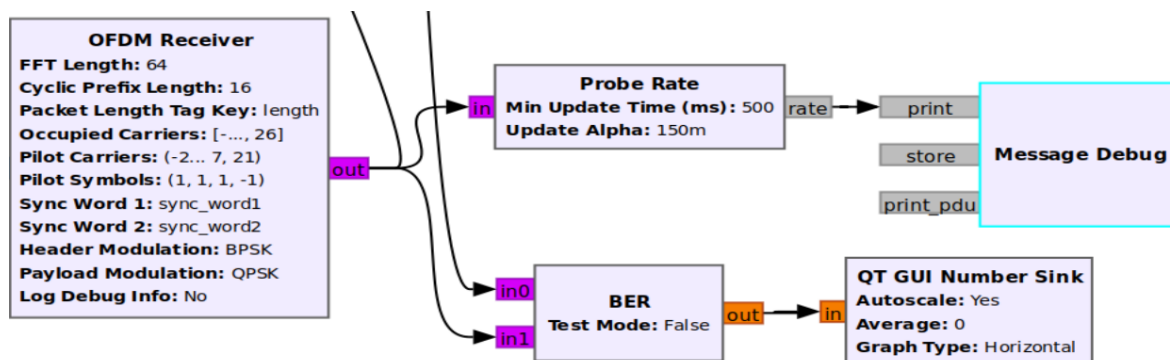
Nota: Autoría propia

El restante de parámetros que se tuvieron en cuenta para este sistema de transmisión fue explicado al detalle en la sección Diseño de un Sistema de Transmisión OFDM.

### Configuración de bloques para indicadores de calidad

Teniendo en cuenta que el throughput es la cantidad de información que se transmite por unidad de tiempo, medida en un instante. Para poder visualizar los datos de throughput, en las simulaciones realizadas, se emplearon el bloque *Probe Rate* junto con el bloque *Message Debug*, con ellos es posible al final de la simulación observar el valor promedio del throughput. Por otra parte, para medir la tasa de error de bit, se utiliza el bloque *BER* y el bloque *QT GUI Number Sink* estos dos muestran en la ventana grafica una barra con el valor del BER. El BER (Bit Error Rate) tiene mucha relevancia en cuanto a la calidad de servicio QoS (Quality of Service), este parámetro es utilizado para medir el desempeño de sistemas de comunicaciones digitales en presencia de ruido. En la figura 39 se pueden observar los bloques utilizados y sus respectivas conexiones.

## Bloques para indicadores de calidad throughput y BER

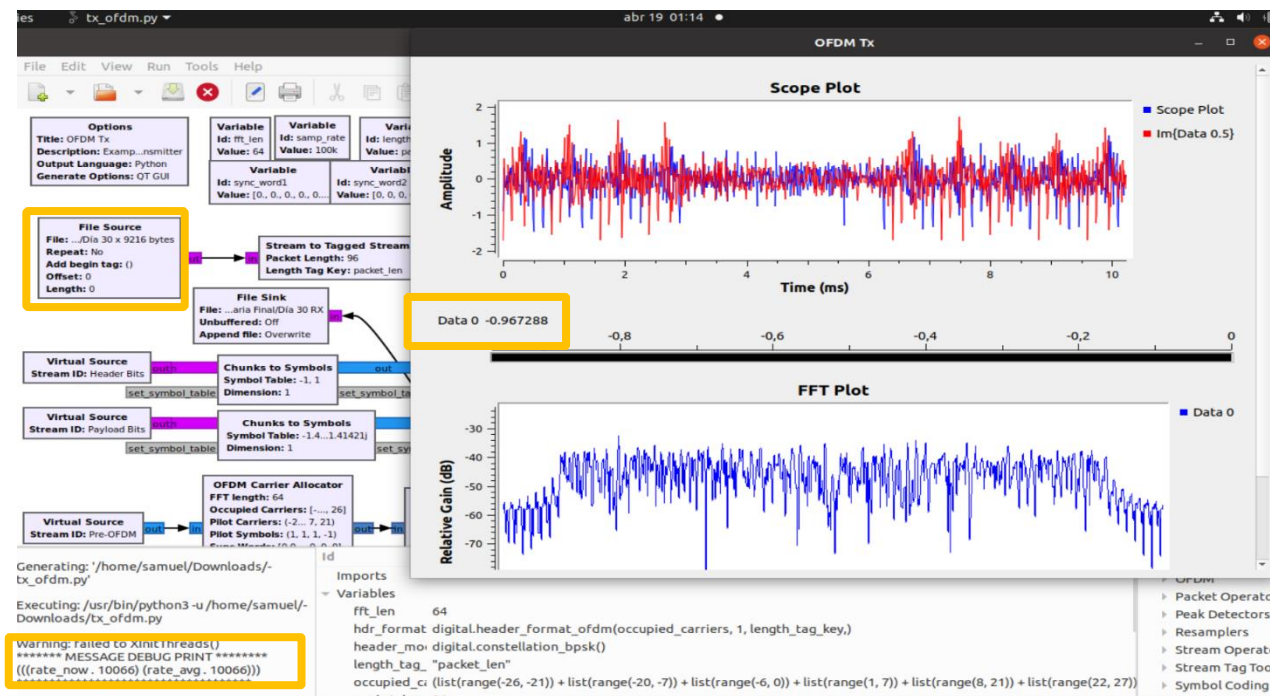


Nota: Autoría propia

## Simulaciones y recopilación de datos

En cada simulación empleando GNU radio, se obtiene gráficamente lo expuesto en la figura 40.

## Simulación del paquete enviado con GNU radio



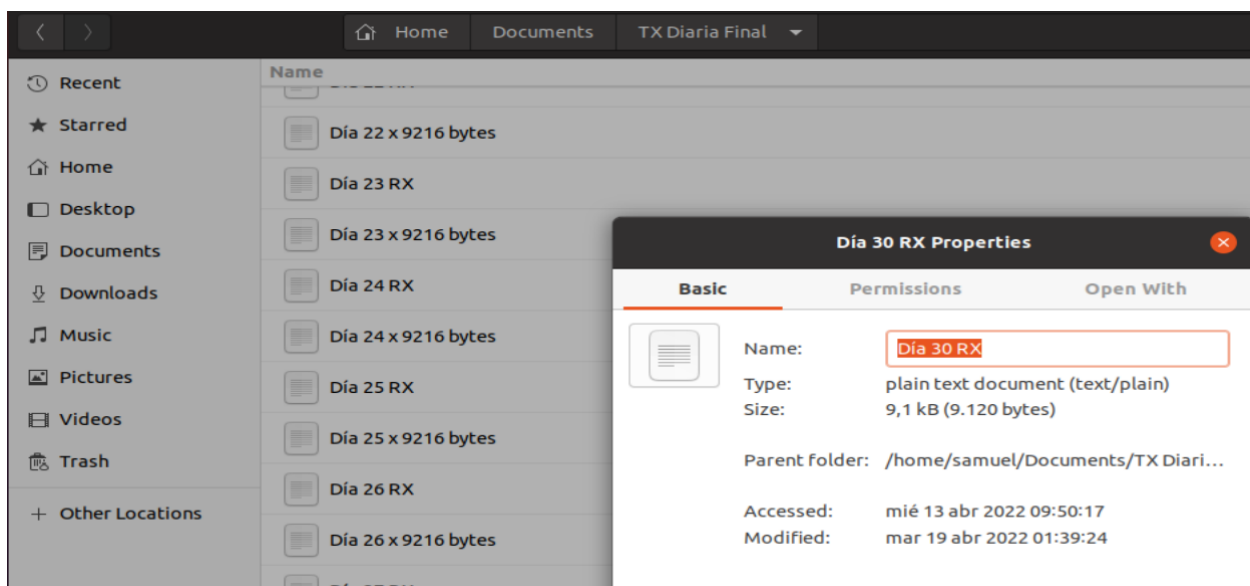
Nota: Autoría propia

En la figura 40 es posible observar el bloque llamado File Source subrayado con color amarillo en la parte superior izquierda, el cual indica la ruta donde se encuentra alojado el paquete de datos a enviar, este bloque guarda los datos recibidos en la simulación. En la parte inferior izquierda se observa el resultado del throughput y en la ventana donde aparecen las gráficas de las señales se puede visualizar una barra negra la cual pertenece al BER (Bit Error Rate).

Después de realizar el proceso de transmisión, mediante el software de simulación, se genera automáticamente un archivo plano, como se mencionó, el cual se guarda en la misma carpeta que el archivo de entrada, este contiene toda la información recibida en la simulación. En la figura 41 se muestra el peso en bytes del archivo denominado *Día 30 RX*.



*Archivo recibido por OFMD en GNU radio*



Nota: Autoría propia

Luego de realizar las 30 simulaciones en GNU radio, desde el día 1 hasta el día 30, se procede a consolidar los datos de transmisión y los resultados del throughput tal como se pueden observar en la tabla número 2.

**Tabla 2.**

*Resultados de las simulaciones en GNU radio del Throughput*

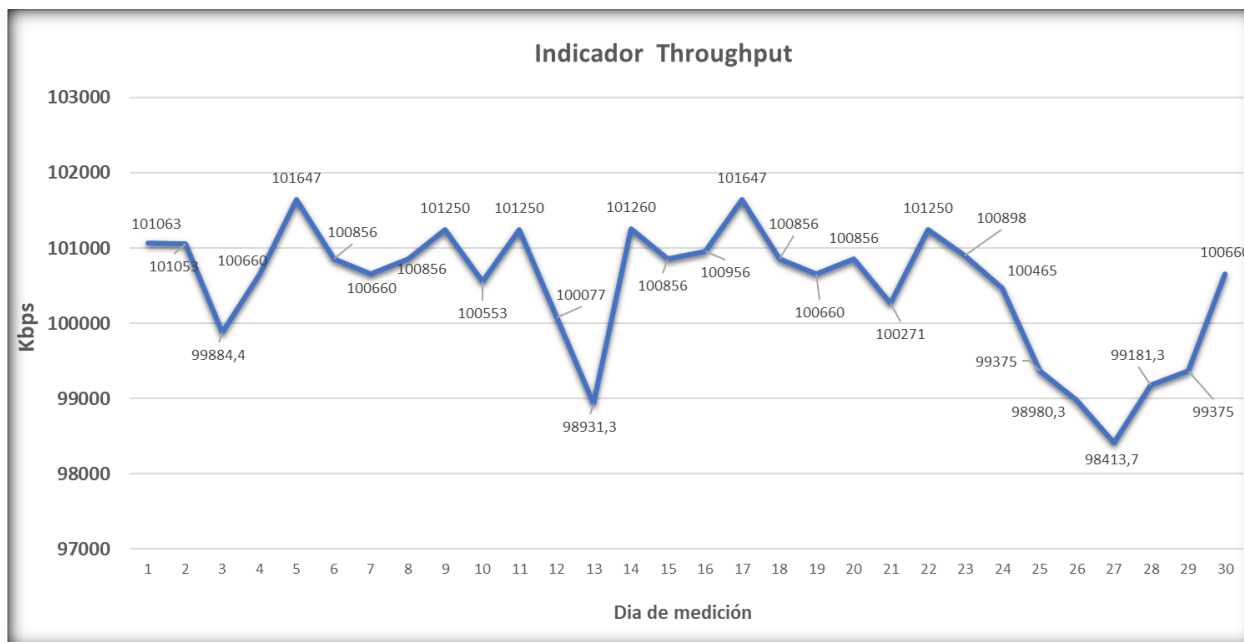
<b>Día</b>	<b>Archivo Inicial Bytes</b>	<b>Archivo Final Bytes</b>	<b>Throughput</b>
<b>1</b>	9216	9120	101063
<b>2</b>	9216	9120	101053
<b>3</b>	9216	9120	99884,4
<b>4</b>	9216	9120	100660
<b>5</b>	9216	9120	101647
<b>6</b>	9216	9120	100856
<b>7</b>	9216	9120	100660
<b>8</b>	9216	9120	100856
<b>9</b>	9216	9120	101250
<b>10</b>	9216	9120	100553
<b>11</b>	9216	9120	101250
<b>12</b>	9216	9120	100077
<b>13</b>	9216	9120	98931,3
<b>14</b>	9216	9120	101260
<b>15</b>	9216	9120	100856
<b>16</b>	9216	9120	100956
<b>17</b>	9216	9120	101647
<b>18</b>	9216	9120	100856
<b>19</b>	9216	9120	100660
<b>20</b>	9216	9120	100856
<b>21</b>	9216	9120	100271

<b>Día</b>	<b>Archivo Inicial Bytes</b>	<b>Archivo Final Bytes</b>	<b>Throughput</b>
<b>22</b>	9216	9120	101250
<b>23</b>	9216	9120	100898
<b>24</b>	9216	9120	100465
<b>25</b>	9216	9120	99375
<b>26</b>	9216	9120	98980,3
<b>27</b>	9216	9120	98413,7
<b>28</b>	9216	9120	99181,3
<b>29</b>	9216	9120	99375
<b>30</b>	9216	9120	100660

Nota: Estos son los datos recopilados en las simulaciones de 30 mediciones equivalentes a 30 días con su indicador de throughput. Autoría propia.

En la anterior tabla, la tercera columna denominada Archivo Final Bytes, corresponde al archivo que se recibe al final del proceso de transmisión en GNU radio, como se puede observar todos los archivos se recibieron con 96 Bytes menos respecto al archivo que se envía, esto se debe a la sincronización del sistema de comunicación. Por otra parte, se realiza la siguiente gráfica teniendo en cuenta los datos obtenidos con respecto al Throughput.

### Grafica indicador de Throughput



Nota: Autoría propia

Como análisis de esta gráfica se puede observar que la velocidad de transmisión es constante en todas las simulaciones, ya que desde el día 1 hasta el día 30, varía entre un máximo de 101647 Kbps y un mínimo de 98413.7 Kbps, con lo que se demuestra que el sistema es estable en cuanto a la velocidad de información con la que se transmiten los datos.

Respecto a las simulaciones que se realizaron es importante indicar que el sample rate se estableció como 100Kbps, sin embargo, al aumentar el sample rate por ejemplo a 1Mbps o 100Mbps, el Throughput fue el mismo, lo que demuestra una limitación del computador empleado para realizar la simulación, es decir la capacidad del procesador utilizado se satura con un sample rate mayor y siempre arroja el mismo throughput. Incluso con una velocidad de transmisión de 100 Kbps, se puede concluir

que en una futura implementación para una red NB-IoT se podría tener una velocidad de transmisión de acuerdo con los KPI, ya que de acuerdo con los datos una red NB-IoT, se dispone de velocidades máximas de 26 kbps en Release13 3GPP y 127 kbps en Release14. (Tech, 2022).

Por otra parte, con el propósito de medir la calidad del servicio del sistema de transmisión, se realizaron mediciones del BER (Bit Error Rate), el número que se puede observar en la cuarta columna de la tabla 3, corresponde al valor que arroja la simulación de GNU radio, al realizar las 30 simulaciones se recopilaron los datos que se evidencian a continuación.

**Tabla 3.**

*Resultados de las simulaciones en GNU radio del BER (Bit Error Rate)*

<b>Día</b>	<b>Archivo Inicial Bytes</b>	<b>Archivo Final Bytes</b>	<b>BER</b>
<b>1</b>	9216	9120	9,77758E-08
<b>2</b>	9216	9120	9,86681E-08
<b>3</b>	9216	9120	9,77758E-08
<b>4</b>	9216	9120	9,90803E-08
<b>5</b>	9216	9120	9,76583E-08
<b>6</b>	9216	9120	9,75722E-08
<b>7</b>	9216	9120	9,75719E-08
<b>8</b>	9216	9120	9,75190E-08
<b>9</b>	9216	9120	9,75044E-08
<b>10</b>	9216	9120	9,74924E-08
<b>11</b>	9216	9120	9,74751E-08
<b>12</b>	9216	9120	9,73948E-08
<b>13</b>	9216	9120	9,72782E-08

<b>Día</b>	<b>Archivo Inicial Bytes</b>	<b>Archivo Final Bytes</b>	<b>BER</b>
<b>14</b>	9216	9120	9,71475E-08
<b>15</b>	9216	9120	9,69810E-08
<b>16</b>	9216	9120	9,78790E-08
<b>17</b>	9216	9120	9,67863E-08
<b>18</b>	9216	9120	9,66642E-08
<b>19</b>	9216	9120	9,67111E-08
<b>20</b>	9216	9120	9,65718E-08
<b>21</b>	9216	9120	9,64781E-08
<b>22</b>	9216	9120	9,64781E-08
<b>23</b>	9216	9120	9,64781E-08
<b>24</b>	9216	9120	9,76142E-08
<b>25</b>	9216	9120	9,76803E-08
<b>26</b>	9216	9120	9,76509E-08
<b>27</b>	9216	9120	9,74167E-08
<b>28</b>	9216	9120	9,72855E-08
<b>29</b>	9216	9120	9,75629E-08
<b>30</b>	9216	9120	9,67288E-08

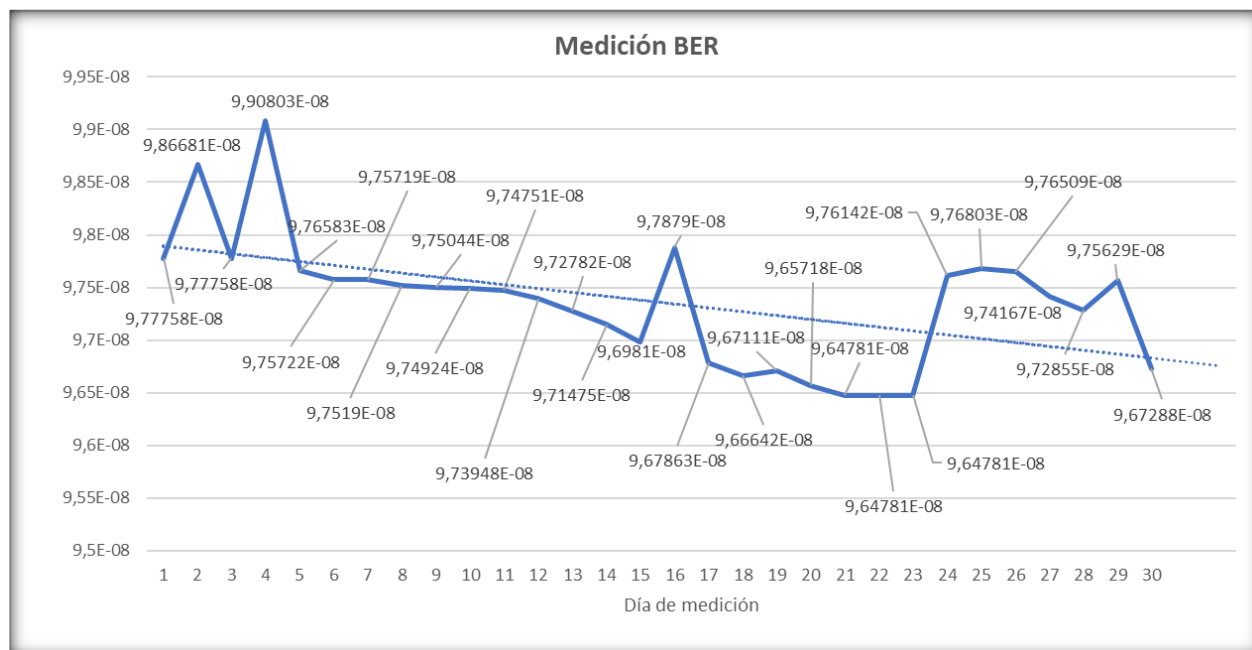
Nota: Estos son los datos recopilados en las simulaciones de 30 mediciones equivalentes a 30 días con su indicador de calidad BER. Autoría propia.

A continuación, con los datos del BER de la tabla 3, se configura la siguiente gráfica, para ello se seleccionaron la cantidad de simulaciones realizadas en el eje “x”, y en el eje “y” se seleccionaron los valores de BER, sin embargo, para representarlos y obtener los valores que a continuación se grafican, fue necesario ingresar a las configuraciones del eje “y”, se establece el valor mínimo y valor máximo a graficar, luego en la opción unidades de visualización se elige el ítem millones y por último se

elige mostrar los datos en escala logarítmica, de esa manera se obtiene la siguiente representación gráfica del BER.



*Grafica Bit Error Rate*



Nota: Autoría propia

Como se observa en la gráfica, no hay mucha variación en el BER, las mediciones realizadas oscilan entre un máximo de  $9,90803E-08$  y un mínimo de  $9,64781E-08$ , lo más importante es que el valor se acerca notablemente a cero, quiere decir que la tasa de error de bit es baja y existe una gran confiabilidad de los datos como la que se exige en 4G o 5G NB-IoT. Lo anterior hace referencia a URLLC. (Hameed & Rehman, 2020) ultra-reliable low latency communications, este es uno de los grandes pilares de las redes móviles de última generación, que consiste en ofrecer servicios avanzados de conexión para habilitar aplicaciones en un amplio espectro, dentro de los cuales está Internet de las cosas.

## Conclusiones

En el proceso de caracterizar los elementos del sistema de medición de la calidad del aire en Bogotá, se analizaron las redes móviles de última generación, con esto se llegó a la conclusión que la tecnología NB IoT, es la más apropiada para un sistema que involucre cientos de sensores midiendo calidad del aire y transmitiendo los datos a través de las redes móviles, adicionalmente NB IoT es una tecnología de baja potencia, que tiene un consumo bajo de energía, es espectralmente eficiente, tiene baja complejidad y se espera que su costo disminuya rápidamente a medida que aumenta la demanda.

Las redes móviles de última generación están compuestas de un core y una red de acceso (RAN), en el proyecto se emuló una red NB-IoT que contiene dichos elementos. Se concluyó que el software OpenAirInterface es la mejor opción disponible para emular los equipos del operador, por ser un software de código abierto y por la cantidad de información que se encuentra disponible en internet.

En cuanto a la simulación del sistema de transmisión de información de calidad del aire a través de NB-IoT, se concluyó que en la implementación es importante tener en cuenta la correcta instalación del software en cada uno de los equipos, las versiones apropiadas, los controladores de los periféricos, adicionalmente, el hardware utilizado también debe soportar y brindar capacidad de procesamiento para una red móvil.

La emulación del User Equipment(UE) se realizó mediante comandos AT desde módulo BC95-G al eNodeB, sin embargo, la conexión al core no fue exitosa, debido a que el software OpenAirInterface no cuenta con un protocolo que tenga la capacidad de

asignar direcciones IP a los usuarios finales. Por lo anterior, se realizaron simulaciones mediante GNURadio de los datos de calidad del aire que se transmiten al UE.

Con los datos técnicos de la simulación, se elaboró una gráfica para analizar el BER y Throughput del sistema de transmisión, llegando a la conclusión que el sistema de comunicación simulado cumple con los indicadores de desempeño de las redes de última generación y las redes NBloT, siendo un sistema confiable en cuanto a la información enviada, debido a que las mediciones del BER oscilan entre un máximo de  $9,90803E-08$  y un mínimo de  $9,64781E-08$  y cumple con la velocidad de transmisión para este tipo de tecnología, debido a que las mediciones muestran un Throughput entre un máximo de 101647 Kbps y un mínimo de 98413.7 Kbps.

## Referencias

- Agiwal, M. (2016). Redes inalámbricas 5G de próxima generación: una encuesta completa. IEEE Xplore.
- Andrews, J. G., Buzzi, S., Choi, W., Hanly, S. V., Lozano, A., Soong, A. C., & C, J. (Junio de 2014). What Will 5G Be? Obtenido de IEEEExplore:  
<https://ieeexplore.ieee.org/document/6824752>
- Ariza Vesga, L. (2021). Introducción a la tarjeta de desarrollo Quectel BC95 G. Obtenido de <https://gitlab.com/comunidad-sucree-investiga/sucree-investiga/-/wikis/Introducci%C3%B3n-a-la-tarjeta-de-desarrollo-Quectel-BC95-G>
- Baldauf, R., Heist, D., Isakov, V., Perry, S., Hagler, G., Kimbrough, S., . . . Brixey, L. (21 de septiembre de 2012). Air quality variability near a highway in a complex urban environment. Obtenido de  
<file:///C:/Users/samue/Downloads/1.%20Trabajo%20de%20Grado/1-s2.0-S1352231012009259-main.pdf>
- Boris Galvis, N., & Rojas. (junio de 2006). Relación entre PM<sub>2,5</sub> y PM<sub>10</sub> en la ciudad de Bogotá. Obtenido de  
[http://www.scielo.org.bo/scielo.php?script=sci\\_arttext&pid=S1683-07892006000100013&lng=es&tlng=es](http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892006000100013&lng=es&tlng=es)
- Castiblanco, C. (2021). Bogotá. Obtenido de Contaminación en el aire de Bogotá ha caído un 47 %: <https://bogota.gov.co/mi-ciudad/ambiente/disminucion-del-material-particulado-en-el-aire-de-bogota>

Digital, S. I. (28 de Noviembre de 2018). La FM. Obtenido de Colombia, segundo país de Latinoamérica con más contaminación del aire:

<https://www.lafm.com.co/medio-ambiente/colombia-segundo-pais-de-latinoamerica-con-mas-contaminacion-del-aire>

Electronics, M. (2022). Mouser Electronics. Obtenido de

<https://co.mouser.com/new/quectel/quectel-gsm-nb-iot-evb-kit/>

ETMY. (junio de 2019). ¿Cuáles son las definiciones de banda ancha y banda estrecha, y cuál es la diferencia? Obtenido de [http://es.handheld-](http://es.handheld-twowayradio.com/info/what-are-the-definitions-of-wideband-and-narrowband/)

[twowayradio.com/info/what-are-the-definitions-of-wideband-and-narrowband-31954225.html](http://es.handheld-twowayradio.com/info/what-are-the-definitions-of-wideband-and-narrowband/)

europapress. (mayo de 2020). En 2023 habrá 29.300 millones de dispositivos conectados a Internet, según Cisco. Obtenido de

<https://www.europapress.es/portaltic/internet/noticia-2023-habra-29300-millones-dispositivos-conectados-internet-cisco-20200517112937.html>

Gavidia, P. (31 de octubre de 2019). Meteored. Obtenido de

<https://www.tiempo.com/noticias/actualidad/dia-mundial-de-las-ciudades.html>

GNURadio. (04 de Junio de 2021). Acerca de GNU Radio. Obtenido de GNURadio:

<https://www.gnuradio.org/about/>

GNURadio, W. (2022). [https://wiki.gnuradio.org/index.php/OFDM\\_Transmitter](https://wiki.gnuradio.org/index.php/OFDM_Transmitter). Obtenido

de OFDM Transmitter: [https://wiki.gnuradio.org/index.php/OFDM\\_Transmitter](https://wiki.gnuradio.org/index.php/OFDM_Transmitter)

González Manrique, C. C., & Rodríguez Chisco, C. M. (2020). Prototipo IoT para la medición de la polución atmosférica en la Universidad Católica de Colombia.

Obtenido de

[https://repository.ucatolica.edu.co/handle/10983/24967#:~:text=El%20proyecto%20contempla%20el%20dise%C3%B1o,%20y%20ozono%20\(O3\).](https://repository.ucatolica.edu.co/handle/10983/24967#:~:text=El%20proyecto%20contempla%20el%20dise%C3%B1o,%20y%20ozono%20(O3).)

Guevara Córdoba, F. A. (2019). Comparativo entre la tecnología de redes 4G y 5G y los beneficios de su implementación en Colombia. DSPACE.

Hameed , U., & Rehman, Z. (2020). Technical Challenges in Achieving Ultra-Reliable & Low Latency Communication in 5G Cellular- V2X Systems. LC INTERNATIONAL JOURNAL OF STEM .

Instruments, N. (marzo de 2022). <https://www.ni.com/es-co/shop/hardware/products/usrp-software-defined-radio-device.html>. Obtenido de Dispositivo de Radio Definido por Software USRP: <https://www.ni.com/es-co/shop/hardware/products/usrp-software-defined-radio-device.html>

KeepCoding. (2021). KeepCoding Tech School. Obtenido de <https://keepcoding.io/blog/que-son-datasets/>

L, A., K, A., Savarinathan, P., & Jayapalan, A. (Enero de 2021). Data Transfer through OFDM transceiver with the aid of GNU Radio. Obtenido de IEEEExplore: <https://ieeexplore-ieee-org.bibliotecavirtual.unad.edu.co/document/9402541>

Machado Fernández, J. R. (29 de septiembre de 2014). Software Defined Radio: Basic Principles and Applications. Obtenido de [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0121-11292015000100007](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-11292015000100007)

Martínez Villalobos. (2020). Desarrollo de un prototipo para la medición de partículas PM2.5 en la ciudad de Bogotá. Obtenido de <https://repository.ucatolica.edu.co/handle/10983/25540>

- Mircea, M., Stoica, M., & Bogdan. (Febrero de 2021). ResearchGate. Obtenido de [https://www.researchgate.net/publication/349508412\\_Investigating\\_the\\_Impact\\_of\\_the\\_Internet\\_of\\_Things\\_in\\_Higher\\_Education\\_Environment](https://www.researchgate.net/publication/349508412_Investigating_the_Impact_of_the_Internet_of_Things_in_Higher_Education_Environment)
- Nazarenko. (2021). ir quality standards for the concentration of particulate matter 2.5, global descriptive analysis. Bulletin of the World Health Organization. [bibliotecavirtual.unad.edu.co](http://bibliotecavirtual.unad.edu.co).
- OpenAirInterface. (Junio de 2021). Open AirInterface. Obtenido de GRUPO DE PROYECTOS OAI MOSAIC5G: <https://openairinterface.org/mosaic5g/>
- OpenAirInterface.org. (2021). Proyecto de red de acceso de radio OpenAirInterface 5G. Obtenido de <https://openairinterface.org/oai-5g-ran-project/>
- OpenAirInterface.org. (2021). Proyecto Red Core5G. Obtenido de <https://openairinterface.org/oai-5g-core-network-project/>
- OPS. (2021). Organización panamericana de la salud. Obtenido de <https://www.paho.org/es/temas/calidad-aire>
- Pachon Quinche, J. E. (2018). Análisis de inmisión de material particulado PM10 y PM2.5 en dos puntos estratégicos de la ciudad de Bogotá. Obtenido de [https://ciencia.lasalle.edu.co/ing\\_ambiental\\_sanitaria/1092/](https://ciencia.lasalle.edu.co/ing_ambiental_sanitaria/1092/)
- Paluga, M. (2021). Fritsch. Obtenido de LA TEORÍA DE MIE: <https://www.fritsch.es/medicion-de-particulas/conocimientos-de-fritsch/la-teoria-de-mie/>
- Peterson, L., & Sunay, O. (2020). 5G Mobile Networks: A Systems Approach. Obtenido de <https://5g.systemsapproach.org/>

- Plokiko. (agosto de 2019). ¿Qué significa que existan dos tipos de 5G? Diferencias y compatibilidades entre 5G NSA y 5G SA. Obtenido de <https://www.xataka.com/moviles/que-significa-que-existan-dos-tipos-5g-diferencias-compatibilidades-5g-nsa-5g-sa>
- PMS5003. (2021). Digital universal particle concentration sensor. Obtenido de series data manual: [http://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual\\_v2-3.pdf?sfvrsn=2](http://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual_v2-3.pdf?sfvrsn=2)
- quectel. (2021). [www.quectel.com](http://www.quectel.com). Obtenido de [https://www.quectel.com/wp-content/uploads/pdfupload/Quectel\\_GSM\\_EVB\\_User\\_Guide\\_V3.4.pdf](https://www.quectel.com/wp-content/uploads/pdfupload/Quectel_GSM_EVB_User_Guide_V3.4.pdf)
- Restrepo Mantilla, Ó. J. (29 de marzo de 2019). Publicaciones El Carro Colombiano. Obtenido de <https://www.elcarrocolombiano.com/notas-de-interes/contaminacion-en-las-ciudades-de-colombia-cortinas-y-colchones-de-humo/>
- Rojano, R., Angulo, L., & Restrepo, G. (septiembre de 2012). Niveles de Partículas Suspensas Totales (PST), PM10 y PM2.5 y su Relación en Lugares Públicos de la Ciudad Riohacha, Caribe Colombiano. Obtenido de [https://www.scielo.cl/scielo.php?script=sci\\_arttext&pid=S0718-07642013000200006&lng=en&nrm=iso&tlng=en](https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642013000200006&lng=en&nrm=iso&tlng=en)
- Secretaria de Ambiente de Bogotá. (2021). Secretaria de Ambiente de Bogotá. Obtenido de Red de Monitoreo de Calidad del Aire de Bogotá - RMCAB: <https://www.ambientebogota.gov.co/calidad-del-aire>

Sensor, H.-A. L. (2021). HK-A5 Laser PM2.5/10 Sensor. Obtenido de

<https://drive.google.com/file/d/0BzaKjvCRihgbdXVQYWISVzc5N0E/view?resourcekey=0-C8o8AmtizKPV5gC6TLzPug>

Shafi, M., Molisch, A. F., Smith, P. J., Haustein, T., Zhu, P., Silva, P. D., & Tufvesson, F. (junio de 2017). 5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice. Obtenido de IEEEExplore:

<http://ieeexplore.ieee.org/document/7894280>

Tech, N. (2022). <https://nctech.com.mx/>. Obtenido de [https://nctech.com.mx/blog/iot-](https://nctech.com.mx/blog/iot-industrial/nb-)

[iot/#:~:text=IoT%20ofrece%20una%20velocidad%20de,kbps%20en%20la%20Verisi%C3%B3n%2014.](https://nctech.com.mx/blog/iot-industrial/nb-)

TOP-electronics. (2022). TOP-electronics. Obtenido de [https://www.top-](https://www.top-electronics.com/en/nb-iot-module-multiband-on-adapt-board)

Velazco Ortiz, J., & Pinzon Torres, M. (octubre de 2018). Prototipo basado en internet de las cosas para el monitoreo ambiental como contribución a la orientación de Bucaramanga (Colombia) hacia Smart City. Obtenido de <https://repository.unab.edu.co/handle/20.500.12749/7271>

wikipedia.org. (22 de marzo de 2022). wikipedia.org. Obtenido de [https://en.wikipedia.org/wiki/Bit\\_error\\_rate](https://en.wikipedia.org/wiki/Bit_error_rate)