UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

Previo a la obtención del título de:

INGENIERO EN TELECOMUNICACIONES

Implementación de estación 4G-LTE mediante radio definida por software (SDR) para el laboratorio de telecomunicaciones

AUTORES:

Balón Tigrero Henry Erick

González Beltrán José Daniel

DOCENTE TUTOR:

Ing. Daniel Armando Jaramillo Chamba, MSc.

LA LIBERAD – ECUADOR

Año 2025

APROBACIÓN DE DOCENTE TUTOR

En mi calidad de docente tutor del trabajo de Integración Curricular denominado: "Implementación de estación base 4G/LTE mediante radio definida por software (SDR) para el laboratorio de Telecomunicaciones ", elaborado por José Daniel González Beltrán y Henry Erick Balón Tigrero, estudiantes de la Carrera de Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, previo a la obtención del título de Ingeniería en Telecomunicaciones, me permito declarar que, tras supervisar el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos. En consecuencia, lo considero apto en todos sus aspectos y listo para ser evaluado por el docente especialista.

Atentamente

Ing. Daniel Jaramillo Chamba, Mgtr

DOCENTE TUTOR

DECLARACIÓN DE AUTORIA DE LOS ESTUDIANTES

El presente trabajo de Integración Curricular con el título **"Implementación de estación base 4G/LTE mediante radio definida por software (SDR) para el laboratorio de Telecomunicaciones"**, declaramos que la concepción análisis y resultados son originales a la actividad educativa en el área de Telecomunicaciones.

Atentamente

José Daniel González Beltrán
C.I: 0928072453

Henry Erick Balón Tigrero
C.I: 2400152241

DECLARACIÓN DE DOCENTE ESPECIALISTA

En mi calidad de docente especialista del trabajo de Integración Curricular, **"Implementación de estación base 4G/LTE mediante radio definida por software (SDR) para el laboratorio de Telecomunicaciones"**, elaborado por José Daniel González Beltrán y Henry Erick Balón Tigrero, estudiantes de la carrera de Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, previo a la obtención del título de Ingeniería en Telecomunicaciones, me permito declara que, tras supervisar el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos. En consecuencia, lo considero apto en todos sus aspectos y listo para la sustentación del trabajo.

Atentamente

Ing. Manuel Montaño Blacio, Mgtr

DOCENTE ESPECIALISTA

DECLARACIÓN DE RESPONSABILIDAD

Quienes suscriben, **José Daniel González Beltrán con C.I. 0928072453 y Henry Erick Balón Tigrero con C.I. 2400152241**, estudiantes de la carrera de Telecomunicaciones, declaramos que el trabajo de titulación denominado, **"Implementación de estación base 4G/LTE mediante radio definida por software (SDR) para el laboratorio de Telecomunicaciones"** pertenece y es exclusiva responsabilidad de los autores y pertenece al patrimonio intelectual de la Universidad Estatal Península de Santa Elena.

Atentamente

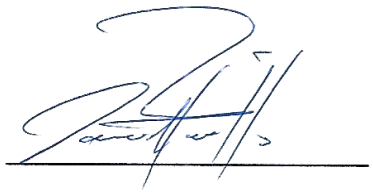
José Daniel González Beltrán
C.I: 0928072453

Henry Erick Balón Tigrero
C.I: 2400152241

TRIBUNAL DE SUSTENTACIÓN

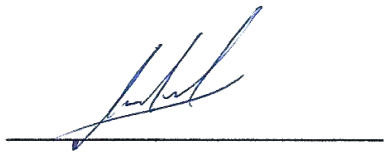
Ing. Ronald Rovira Jurado. Ph.D.

DIRECTOR DE LA CARRERA



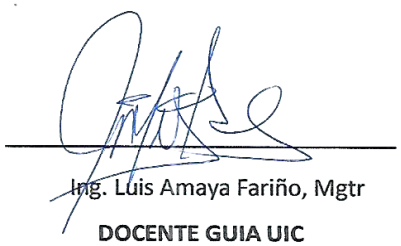
Ing. Daniel Jaramillo Chamba, Mgtr

DOCENTE TUTOR



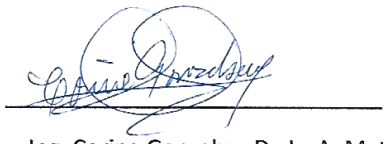
Ing. Manuel Montaño Blacio, Mgtr

DOCENTE ESPECIALISTA



Ing. Luis Amaya Fariño, Mgtr

DOCENTE GUIA UIC



Ing. Corina Gonzabay De La A, Mgtr

SECRETARIA

DEDICATORIA

Dedico este proyecto de titulación va dedicado a mis padres Eduardo González y Gloria Beltrán, a mi hermano Eduardo, por ser parte de este proceso, por su esfuerzo y por siempre me brindarme su apoyo incondicional en todo momento.

A mis queridos amigos Steven, Nohelia, por apoyarme en toda esta travesía, dándome su apoyo incondicional y palabras de ánimo para seguir adelante con toda mi trayectoria estudiantil hasta llegar a ser profesional.

José Daniel González Beltrán

Dedico este trabajo primeramente a las personas más importantes de mi vida, y de las cuales he recibido un apoyo incondicional durante la trayectoria de este proceso, mis padres Henry Heriberto Balón González y Maritza Aracely Tigrero Yagual quienes me apoyaron de manera incondicional, tanto económica como emocionalmente, siendo esto la manera de agradecerles tanto esfuerzo de su parte y que gracias a ellos estoy en esta etapa tan importante de mi vida.

A mis hermanos Carlos Enrique Chichande Tigrero y Thays Balón Tigrero, quienes también me apoyaron incondicionalmente, en el largo proceso de mi carrera.

A mi compañero este trabajo de integración José Daniel González Beltrán con quien estuve todo este proceso, con altas y bajas, pero siempre juntos apoyándonos, más que un compañero, un gran amigo que como yo también se convertirá en Ingeniero en Telecomunicaciones

Henry Erick Balón Tigrero

AGRADECIMIENTOS

En primer lugar, quiero agradecer a Dios por darme fuerza y sabiduría para llegar a la meta propuesta desde hace tiempo.

A toda mi familia, en especial a mis padres por estar en este proceso, darme el apoyo emocional y económico para continuar.

A los Ingenieros de la facultad de Sistemas y Telecomunicaciones por ser buenos amigos y guías en mi carrera estudiantil, en especial al Ingeniero. Daniel Jaramillo por ser el guía en el desarrollo de este proyecto, y brindarnos su tiempo.

Así como a mis compañeros de la carrera de Telecomunicaciones, por brindarme su apoyo y conocimientos para poder seguir adelante en especial a Geomayra, Melanie y Gabriela por el tiempo compartido en todos estos años, ayudándonos a crecer y poder seguir aprendiendo cada día.

Y a mi amigo Henry Balón, agradecerle por sus conocimientos y paciencia en toda la carrera universitaria, además por ser mi compañero en este proyecto.

José Daniel González Beltrán

En primer lugar, agradeciéndole a Dios por permitirme vivir este momento tan importante, que es solo una etapa de esta aventura a la que llamamos vida.

A mis padres y hermanos que siempre me apoyaron en las buenas y en las malas y que a pesar de todos los problemas y dificultades estuvieron ahí para mí, sé que no fue fácil, pero de todo corazón Gracias por no rendirse conmigo.

A mis amigos Harold y Bryan, quienes siempre estuvieron dándome ánimos en este proceso y escuchando mis locuras y experiencias en la universidad.

A los ingenieros de la facultad de Sistemas y Telecomunicaciones, quienes siempre estuvieron al tanto en este proceso, dándome apoyo y ofreciéndome sus conocimiento y experiencia en todo mi proceso académico.

A mis amigos de universidad por estar presente en todo momento de la carrera y darme palabras de aliento y brindándome consejos que me ayudaron a realizar este proyecto.

A mi Docente Tutor Daniel Armando Jaramillo quien nos brindó sus conocimientos en el área, además de guiarnos en el proceso.

A mi docente especialista quien nos apoyó con sus conocimientos y sugerencias en el proyecto que realizamos.

A mi gran amiga Maeba Nazareno que me brindo su apoyo en el proceso, escuchando mis problemas con el proyecto y dándome palabras de aliento**.**

Henry Erick Balón Tigrero

Índice General

Índice de Figuras

Índice de Tablas

Glosario

3GPP: 3rd Generatión Partnership Project, 16

CF: Core Framework, 39

CORBA: Common Object Request Broker Arquitecture, 39

DAC: Digital Analog Converter, 38

EDGE: Enhanced Data Rates for GSM Evolution, 16

EMM: EPS Mobility Management, 144

FCI: • Framework Control Interfaces, 39

FSI: • Framework Service Interfaces, 39

Resumen

El presente proyecto de investigación desarrolló una arquitectura 4G-LTE simulada mediante radio definida por software (SDR) para el laboratorio de Telecomunicaciones. El objetivo general es fomentar la educación practica experimental en los laboratorios de telecomunicaciones mediante los equipos SDR y softwares de código abierto basadas en redes móviles de última generación para los estudiantes de ingeniera en telecomunicaciones.

Primero se realizó un estudio de los diferentes equipos SDR que se pueden encontrar en el mercado que soporten frecuencias de LTE (Long Term Evolutión) y diferentes aplicaciones, en este caso se utilizó un equipo SDR BladeRF. Además, se configuro computadores disponibles en el laboratorio con el fin de simular la arquitectura, que será la encargada de simular las diferentes entidades por medio de softwares Open5GS y srsRAN.

Se realizaron pruebas de seguridad, conexión, tráfico de paquetes por medio de softwares especializados como Wireshark que nos ayudaron a definir limitantes del proyecto y mejoras que se pueden realizar en casos de estudios o docentes que hagan uso del equipo BladeRF, finalmente este proyecto busca ofrecer una herramienta practica experimental que permita entender el funcionamiento de las redes móviles.

Palabras Clave: Redes móviles, Software de código abierto, 4G-LTE, BladeRF, Arquitectura.

Abstract

This research project developed a simulated 4G-LTE architecture using software-defined radio (SDR) for the Telecommunications Laboratory. The overall objective is to promote hands-on experimental training in telecommunications laboratories using SDR equipment and open-source software based on next-generation mobile networks for telecommunications engineering students.

First, a study was conducted of the different SDR devices available on the market that support LTE (Long Term Evolution) frequencies and different applications. In this case, a BladeRF SDR device was used. Additionally, computers available in the laboratory were configured to simulate the architecture, which will be responsible for simulating the different entities using Open5GS and srsRAN software.

Security, connection, and packet traffic tests were performed using specialized software such as Wireshark, which helped us define project limitations and improvements that can be made in case studies or by teachers who use BladeRF equipment. Ultimately, this project seeks to provide a practical experimental tool that allows us to understand how mobile networks work.

Keywords: Mobile networks, Open-source software, 4G-LTE, BladeRF, Architecture.

Introducción

Las tecnologías de red móvil han tenido un gran impacto desde la segunda mitad del siglo pasado, aunque no se dio a conocer de manera pública comercialmente hasta los años 80 con la primera generación (1G), que utilizada el estándar AMPS (Advanced Mobile Phone System) [1]. En esa época se pudo permitir la transmisión de voz, aunque no muy segura. La tecnología empleada era totalmente analógica usada bajo multiplexación FDMA (Frequency División Múltiple Access).

Debido a la poca seguridad de 1G, en los años 90 se implementó 2G denominada como GSM (Global System for Mobile Communications), lo que no solo aumento la seguridad, sino también transmitir datos digitales además de voz.

Para aumentar la velocidad se implementa el estándar GPRS (General Packet Radio Service), lo que permitió poder realizar mensajes cortos, los llamados SMS (Short Message Service) y la agregación de contenido multimedia en los mismos.

La gran revolución llego en el año 2003 con el Nacimiento de UMTS (Universal Mobile Telecommunications System), caracterizada por la gran taza de transferencia combinada con la adquisición de varios protocolos de radio (W – CDMA (Wideband Code División Multiple Access), esta nueva generación 3G pudo alcanzar velocidades de hasta 2 Mbps y dio paso a la navegación por internet y la reproducción de contenido multimedia.

Esto no se detuvo aquí y a finales del año 2009 la 3GPP anuncio un récord de abonados para UMTS, por lo que ese mismo año se lanzó un nuevo estándar que mejoraría aún más las redes móviles como se conocían en aquel entonces, mejorando la velocidad de transferencia de datos y la experiencia de usuario por medio del protocolo IP basado en paquetes. Este estándar se denominó LTE, el cual fue la base de la siguiente generación 4G [1].

La tecnología 4G cumple el objetivo de proveer velocidad, calidad, seguridad y servicios a un bajo costo, brindando al usuario manejo de alta calidad en voz, datos y contenido multimedia, en base a paquetes de internet por medio del protocolo IP [2].

En la actualidad LTE seguirá presente en las redes móviles, aunque la nueva generación 5G ya está más cerca que nunca y muchos países ya comienzan a implementarla, sus terminales son costosas y hacen que su adquisición se dificulte, también dependerá de cada país y la entidad que regule la liberación de espectro.

En Ecuador 5G aun no es una realidad y aun pasaran varios años para ver una red 5G completa en el país, por lo tanto las redes móviles 4G son de mucha importancia para el país, ya que serán la tecnología predominante en el país, según datos del ARCOTEL(Agencia de Regulación y Control de las Telecomunicaciones), la cobertura 4G va en aumento con la cantidad de radio bases de las diferentes empresas de telecomunicaciones, entre ellas CNT E.P(Corporación Nacional de Telecomunicaciones), CONECEL S:A(Corporacion Nacional de Celulares – CLARO) y OTECEL(Operadora de Telecomunicaciones del Ecuador - MOVISTAR) S. A, siendo CONECEL la que posee la mayor cantidad de radio bases instaladas [2]

El Software Defined Radio (SDR) es una tecnología que ha destacado mucho en los últimos años. Aunque es relativamente nueva al tener su origen a finales del siglo pasado, su desarrollo se ha dado en los últimos años.

El gran impacto de esta tecnología se debe a su capacidad para el despliegue de protocolos de radiocomunicación simplemente con la utilización de un software especifico que solo requerirá ciertas actualizaciones dependiendo de que se requiera, sin tener la necesidad de cambiar el hardware.

La flexibilidad de esta tecnología permite que empresas puedan tener un ahorro significativo, ya que no se requiere la obtención de hardware específico para cada sistema de radio, sino que se puede implementar cada sistema de radio en un solo dispositivo.

Esto sería ideal en un ambiente de laboratorio, sobre todo en educación, la idea de poder implementar diferentes tipos de escenarios basados en sistemas de comunicación móvil y softwares de código abierto proporcionaría a los estudiantes un ambiente más real, lo cual les ayudaría a comprender de mayor manera como funcionan las redes móviles actuales.

En este proyecto se ha desplegado una arquitectura de red LTE simulada por medio de Open5GS y SrsRAN, y el equipo BladeRF micro 2.0 xA4, con el objetivo de evaluar dicha tecnología en diferentes escenarios.

Capítulo I

1.1. Antecedentes.

A mediados de los años 90, Joseph Mitola III, quien se dedicó a la idea del desarrollo de nuevos sistemas inalámbricos, investigó y desarrolló un nuevo concepto denominado SDR, naciendo así los equipos de radio definidos por software. Los SDR son dispositivos de comunicación cuya funcionalidad está basada en el software y no requiere modificaciones de hardware durante actualizaciones tecnológicas [1].

El departamento de defensa de los Estados Unidos poseía más de 200 proyectos dedicados a radiocomunicaciones culminado el año de 1990. Entre estos proyectos nació SCA (Software Communications Architecture), que fue un intento por crear un estándar entre varios participes del departamento de defensa de los estados unidos de América, los proyectos más destacables para esto se vieron en JTRS (Joint Tactical Radio System) y JPOES (Joint Program Executive System), ambos basados en componentes de prueba que se usaron para la tecnología SDR[1].

Se propuso el diseño basado en SDR para acceso y modulación en banda base, llevado a cabo en el Centro de Investigación Científica y Educación Superior de Ensenada. El autor usa comunicación de largo alcance entre boyas, proponiendo una red ad-hoc operada en la banda HF(High Frequency), agrupando la información de todas las boyas en un solo nodo [2].

En la universidad UTTyler (Universidad de Texas en Tyler) en Estados Unidos, utilizan un SDR para la implementación y análisis del rendimiento de las redes LTE. Se profundizo en técnicas de acceso múltiple tales como OFDMA (Orthogonal Frequency División Multiple Access) y SCFDMA (Single Carrier Frequency División Multiple Access), haciendo su implementación por medio del software GNU Radio. También se evaluó el desempeño mediante métricas de tasa de error de bit (BER) y la Relación de Pico a Promedio de Potencia (PAPR) [3].

En la Universidad de Cantabria en España, se desarrolló la implementación de una red LTE, con soluciones SDR. En específico, se utilizó un USRP B210, teniendo un gran rango en la utilización de frecuencias en estándares móviles. EL autor creo una estación base LTE y un UE (User Equipment) con ayuda de dos computadores con alto nivel de procesamiento en sus especificaciones, lo que lo ayudo a simular la arquitectura, todo esto sobre el software de código abierto srsLTE que actualmente lleva el nombre de srsRAN y la implementación de un SDR para la transmisión [4].

Así mismo en la Universidad de Jaén, también en España, se implementó una red LTE con soluciones SDR, en esta ocasión se usó el BladeRF X40, utilizando dos computadores como bases para simular toda la arquitectura, utilizando el software de código abierto OAI (OpenAirInterface). Además, se hizo la utilización de USIM para la utilización de dispositivos móviles como terminales (UE), para tener un ambiente más real [5].

En Ecuador también se han presentado proyectos con soluciones SDR. Tal es el caso de la Universidad Nacional de Loja, donde se implementó una red GSM (2G) por medio de SDR, para servicio de voz y mensajería SMS, este proyecto fue basado en YateBTS (empresa que ofrece componentes para redes móviles GSM y LTE para MVNOs y MNOs), la misma que utiliza como hardware BladeRF, software Omnicom fue elegido para simular la red [6].

1.2. Descripción del Proyecto.

El presente proyecto está enfocado en la implementación de una estación base 4G-LTE mediante radio definida por software (SDR) para el laboratorio de telecomunicaciones, con el fin de proporcionar un entorno práctico a los estudiantes donde se puedan realizar diferentes pruebas de red, como conectividad, seguridad, entre otras.

Se propone utilizar computadoras con la capacidad de emular la arquitectura 4G-LTE, y que soporten la distribución de Linux Ubuntu 20.04.6 LTS, la cual brinda herramientas de configuración necesarias para la simulación de la arquitectura.

Se diseñará la arquitectura con base en software de código abierto basados en redes móviles de última generación, que contienen herramientas para redes LTE donde se emulará la EPC con componentes como MME (Mobility Management Entity), SPGW (Serving and Packet Data Network Gateway), HSS (Home Subscriber Server), PCRF (Policy and Charging Rules Función), entre otros; y el eNodeB, donde se creará la estación base por medio del equipo conectado al SDR. Este dispositivo, con capacidad para operar en frecuencias compatibles con la tecnología LTE, será el encargado de emular la estación base, teniendo además la capacidad de simular UE durante la simulación de la red.

La MME será la encargada de la seguridad del sistema, ya que esta entidad proporcionará claves que se van a comparar con las que envían los equipos de los usuarios (UE), y si están coinciden, se establecerá la conexión. Se registrarán datos que estarán en la base del sistema. Se conectarán las computadoras que emularán las entidades por medio de cables USB 3.0, para simular las interfaces correspondientes. La entidad PGW es el punto de conexión a las redes IP externas, y otorga direcciones IP a los equipos de usuario.

La función de la entidad HSS es la de una base de datos, en esta entidad se encuentra la información de cada usuario, equipos terminales y velocidad máxima de conexión. Los equipos de usuarios (UE) serán los dispositivos que generan el tráfico de la red y serán los encargados de comprobar si la red es estable y verificar el acceso a la estación base

Esta arquitectura será ensamblada con dos computadoras, que simularán al EPC y el eNodeB de manera separada, dando la impresión de ser una red más real.

Se realizarán pruebas mediante software especializado para realizar un análisis del funcionamiento de la red LTE simulada, para casos de estudio o mejoras que se puedan realizar en un futuro para los estudiantes de la carrera de telecomunicaciones.

1.3. Objetivos.

Objetivo General

Desarrollar una estación base LTE utilizando Software Defined Radio (SDR) y software de código abierto como herramienta experimental para laboratorio de telecomunicaciones

Objetivos específicos

1. Realizar un estudio del estado del arte de redes LTE (4G), con el fin de comprender su funcionamiento y arquitectura, además, investigar sobre equipos SDR y software que soporte esta tecnología.
2. Diseñar de manera experimental la arquitectura LTE conforme a las especificaciones técnicas de la 3GPP Reléase 12-17 para implementar las entidades MME, HSS, SGW/ PGW.
3. Seleccionar softwares de código abierto compatibles con tecnologías 4G y SDR para configurar los elementos de red que incluyen el EPC (MME, HSS, SGW, PGW, PCRF) y el E – UTRAN (eNodeB).
4. Implementar una arquitectura 4G LTE simulada bajo las especificaciones de la 3GPP Reléase 12-17, por medio de software de código abierto y la implementación de un SDR como estación base.
5. Programar UE virtuales y registrar dispositivos finales en las configuraciones correspondientes, esto con el fin de proporcionar servicios de autenticación y seguridad para el acceso a la red simulada, estableciendo así protocolos de seguridad dictados por la 3GPP.

6. Realizar pruebas experimentales con el fin de analizar parámetros relevantes como velocidad de conexión, estabilidad y seguridad.
7. Proporcionar al laboratorio de Telecomunicaciones herramientas educativas de bajo, fomentando el uso de tecnologías SDR y uso de software de código abierto que puedan ser usadas en actividades practico-experimentales por estudiantes y docentes de la carrera
8. Utilizar software de análisis de protocolos de red y paquetes, con el fin de monitorear, analizar y detectar fallos en la red, representando resultados que nos ayuden a identificar fallos y soluciones para nuestra red simulada.

1.4. Justificación.

El impacto de la radio definida por software (SDR) es de gran relevancia en el contexto comunicaciones inalámbricas, lo que permitió utilizar funciones que antes eran dependientes del hardware del equipo.

Los trabajos de Josep Mitolla III y el departamento de defensa de Estados Unidos con la creación del estándar SCA, además de los proyectos JTRS y JPOES, demuestran soluciones de flexibles y escalables para múltiples necesidades de comunicaciones inalámbricas

Si comparamos proyectos anteriores propuestos por diferentes universidades, la utilización de SDR como solución efectiva a diferentes problemáticas, un ejemplo es el Centro de Investigación Científica y de Educación Superior de Ensenada que realizo una comunicación entre boyas en la banda HF, para una comunicación a largo alcance, por otro lado, la Universidad de Texas en Tyler (UTTyler) desarrollo un análisis orientado a redes 4G LTE donde se estudió técnicas de acceso OFDMA y SC-FDMA, por medio de software como GNU RADIO como herramienta experimental para este proceso.

Por otro lado, la Universidad de Cantabria en España, implementó una red LTE usando un USRP B210 y el software srsRAN, logrando una simulación completa de una arquitectura LTE. De manera similar, la Universidad de Jaén también en España, utilizó un BladeRF X40 y el entorno OpenAirInterface (OAI) para construir una red LTE, pero incluyendo el uso de USIM reales para integrar dispositivos móviles comerciales como terminales (UE), logrando un ambiente de pruebas más cercano a escenarios reales.

Finalmente, en el contexto latinoamericano, la Universidad Nacional de Loja en Ecuador desarrolló una red GSM utilizando BladeRF y YateBTS, mostrando que las tecnologías SDR también son

viables para redes de generaciones anteriores (2G/3G) en países en vías de desarrollo, adaptándose a necesidades de comunicación más básicas como voz y SMS.

Estas comparaciones muestran que, aunque todos los proyectos se basan en la misma filosofía SDR, cada uno adapta el concepto a sus necesidades específicas, utilizando diferentes combinaciones de hardware y software, variando desde aplicaciones en ambientes extremos como el mar hasta redes móviles comerciales LTE y GSM, con fines académicos, experimentales o de desarrollo de soluciones reales.

1.5. Alcance del Proyecto.

El presente proyecto de titulación está enfocado generalmente en desarrollar un eNodeB LTE o estación base 4G con ayuda de SDR (Software Defined Radio) y software libre basado en redes móviles como 3G, 4G e incluso5G, con el fin de proporcionar herramientas práctico-experimentales en el laboratorio de Telecomunicaciones.

El proyecto se implementa en un entorno de prueba controlado, en donde se configura computadores de escritorio que simularan la red LTE con software de código abierto y el uso de SDR para la implementación de una estación base, en donde se conectaran nuestros dispositivos simulados.

Este proyecto simulará la red de núcleo EPC con la arquitectura LTE, que incluirá componentes esenciales como MME, HSS, SGW y PGW, PCRF. Esto permitirá la realización de pruebas de autenticación, movilidad y gestión de datos.

Revisando los proyectos basados en esta tecnología, como la comunicación de largo alcance de las boyas en el Centro de Investigación Científica de Ensenada y el análisis de LTE en la Universidad de Texas, van de acuerdo con implementaciones en redes LTE. De igual manera en la Universidad de Cantabria y la Universidad de Jaén, teniendo un enfoque hacia laboratorios, además de proyectos basados directamente en implementación de redes móviles, como en Loja Ecuador, donde se implementa una red GSM 2G, en un entorno académico con la ayuda de equipos SDR. El proyecto busca brindar plataformas de aprendizaje en entornos de red móvil, sobre todo redes de última generación como LTE, lo que fortalece las capacidades técnicas de los estudiantes en redes de telecomunicaciones, además de fomentar el uso de tecnologías como SDR como herramienta innovadora en este campo

1.6. Metodología.

Investigación Exploratoria

Este tipo de investigación se basa en la recopilación de conceptos mediante un estudio donde se revisarán repositorios académicos y fuentes bibliográficas relacionadas con proyectos que utilicen tecnología 4G-LTE, y Radio Definida por Software (SDR) [7]. Para ello, se estudiará la evolución de las redes de celulares, con el enfoque en la arquitectura y funcionamiento de LTE.

Se realizará un estudio de plataformas de código abierto con el fin de seleccionar la más adecuada para el diseño de la arquitectura de la red 4G-LTE, identificando las ventajas, compatibilidades y limitaciones de cada una.

Investigación Aplicativa

Este tipo de investigación se centra en la prueba y error para el presente proyecto se configura una arquitectura 4G/LTE por medio de software libre, se hará uso de conocimiento en programación adquiridos a lo largo de la carrera para poder simular de manera correcta [8].

Para el desarrollo del presente proyecto se han establecido cuatro fases, diseñadas para cumplir los objetivos específicos planteados y garantizar la culminación del proyecto.

FASE 1: Estudio del arte de las tecnologías móviles, investigando con profundidad 4G-LTE, este análisis tiene como objetivo comprender su arquitectura y el funcionamiento general de la tecnología 4G. Se investigarán equipos SDR y herramientas de software que serán utilizados para la creación de la red simulada.

Se realizará un estudio del arte de las diferentes tecnologías móviles desarrolladas a lo largo de la historia, como 2G/GSM, 3G/UMTS, 4G/LTE y 5G/NR, incluyendo aquellas que se han implementadas en Ecuador.

Se recopilará información teórica de trabajos de investigación académicos, repositorios y fuentes bibliográficas confiable relacionadas con el presente proyecto, con el fin de comprender el funcionamiento y arquitectura de las redes LTE, cumpliendo con las especificaciones técnicas versión 12 hasta la versión 17 establecidas por la 3GPP. Con esto se cumplirá el objetivo específico número 1.

Por último, se investigarán equipos SDR y software de código abierto existentes, analizando su capacidad de simular, configuración y ejecución de pruebas experimentales.

FASE 2: Diseño experimental de la arquitectura LTE, EPS y sus componentes (EPC: MME, HSS, SGW, PGW) y la eNodeB

En la fase se enfoca en el diseño experimental de la arquitectura LTE, siguiendo las especificaciones técnicas de la 3GPP (versiones 12 a 17). La tecnología LTE se basa en una arquitectura donde el EPC gestiona funciones como la autenticación de usuarios, conexión y acceso a datos. Estas funciones están divididas en entidades que simplifican el proceso con respecto a otras tecnologías móviles.

Este diseño permitirá su implementación en la red simulada y su configuración adecuada para su funcionamiento. Cumpliendo así con el objetivo específico numero 2

FASE 3: Selección de software y equipos SDR para la configuración de la arquitectura, implementación y acceso a dispositivos finales en la red 4G/LTE simulada.

Una vez escogido el software, se configurarán las distintas entidades del EPC (MME, HSS, SGW, PGW, PCRF), y se empleará otro software capaz de simular la E – UTRAN (eNodeB) utilizando un SDR como estación base. Este equipo deberá operar dentro de los parámetros que se establecidos por la 3GPP, cumpliendo con los estándares de la red. Cumpliendo así con el objetivo específico número 4.

Finalmente, cuando todos componentes de la red este completamente configurados, se programarán tarjetas USIM programables para garantizar la autenticación y el acceso a la red LTE a los equipos de usuario. Se realizará mediante la configuración de la base de datos y la comunicación con el HSS, entidad encargada de verificar las credenciales de los usuarios y verificar su correcta conexión. Así cumpliendo correctamente con el objetivo específico numero 5

FASE 4: Realizar pruebas de rendimiento, se analiza el rendimiento de la red simuladas y fomentara el uso de tecnologías de código abierto

En esta etapa se realizarán pruebas, se analizará el rendimiento de la red y se fomentará el uso de tecnologías de código, cumpliendo así con los objetivos específicos 6, 7 y 8.

Primero, se realizarán pruebas de rendimiento como tráfico de red, conexión y seguridad de la red. Para verificar el correcto funcionamiento, se utilizarán aplicaciones como Wireshark para capturar el tráfico, así como herramientas aplicaciones softwares capaces de brindarnos datos relevantes sobre la red que se simulara.

Una vez que obtenidos los resultados, analizaremos el rendimiento de la red, identificando posibles limitaciones tanto en el equipo SDR como el software especializado que se estará usando. Se

realizarán tablas ilustrativas para facilitar la interpretación y se propondrán soluciones para mejorar la red o migrar a otras tecnologías de bajo costo. Por último, se fomentará el uso de herramientas educativas, las cuales ayudaran a los estudiantes a comprender estas tecnologías y a realizar prácticas experimentales en el laboratorio de telecomunicaciones.

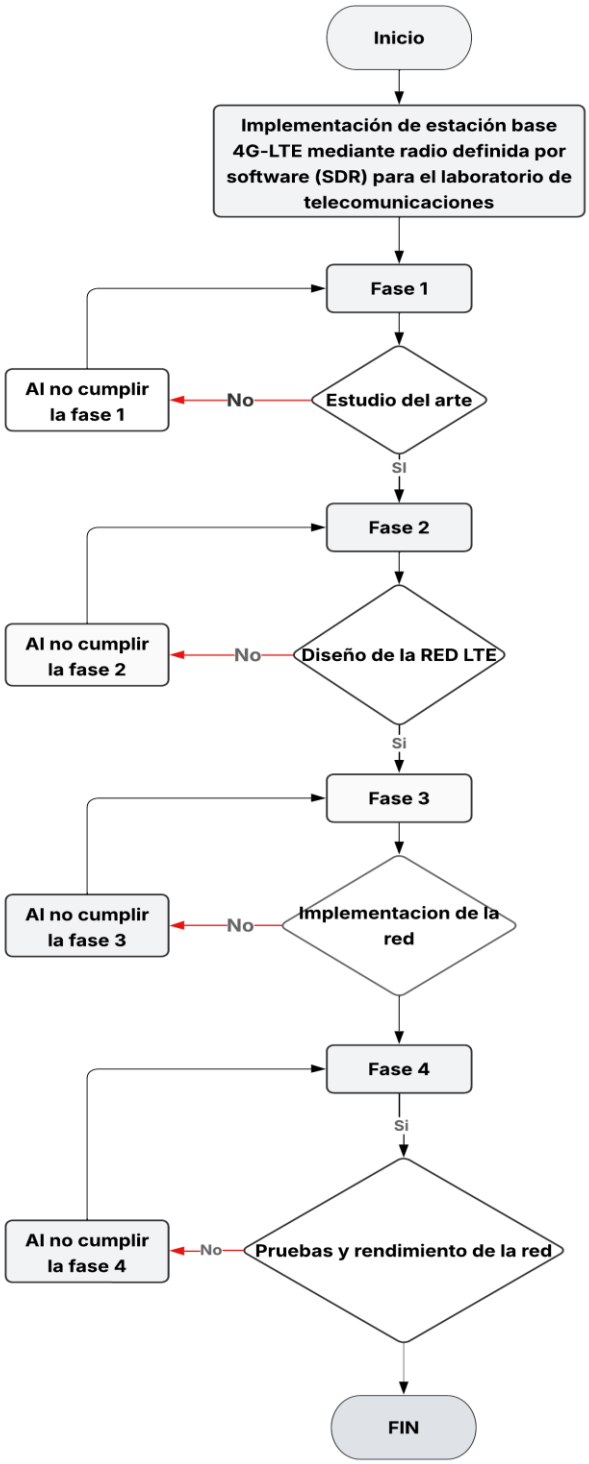


Figura 1 Diagrama de Flujo de la red a simular

1.7. Resultados Esperados

En este proyecto se espera lograr la simulación de una estación base 4G-LTE por medio de un equipo SDR para el laboratorio de Telecomunicaciones. Esta propuesta se centra en acceder a una tecnología de última generación que pueda ser manipulada para comprender su funcionamiento con mayor detalle.

- Se realizará una investigación del tema propuesto, es decir sobre la tecnología SDR, para posteriormente crear la red en base de software de código abierto que nos permita realizar configuraciones en red EPC y eNodeB
- Se desarrollará la red siguiendo los estándares de la 3GPP. Se programará la red con el objetivo de desplegar una estructura funcional y accesible. Este proceso incluirá líneas de códigos y la solución de posibles errores que puedan surgir durante la instalación de las diferentes herramientas de código abierto.
- Se configuran tarjetas SIM LTE programables con el objetivo de establecer conexión con la red desplegada. Estás actuarán como equipos finales, ya que se utilizarán smartphones para poder conectarse, autenticarse y tener acceso a la red.
- Cuando se termine la configuración de los diferentes componentes de la red, se iniciará la red completa y con ayuda de la tarjeta SDR, emulará una estación base, la cual proporcionará la capacidad de establecer conexión entre diferentes equipos de usuarios con las tarjetas SIM configuradas anteriormente.
- Se realizará pruebas de seguridad de la red, tráfico, conexión para evaluar las limitaciones y beneficios que tiene la estación base simulada mediante el equipo SDR.
- Se analizarán posibles mejoras para el proyecto, como el despliegue de una tecnología más avanzada, o implementación de nuevas pruebas, como verificación de claves o la simulación de más equipos de usuarios para comprobar el funcionamiento correcto de la estación base y red LTE.

Capítulo II

2.1. Marco Contextual

Se implementará una arquitectura simulada basada en rede LTE 4G bajo las normativas de la 3GPP, en base a software de código abierto y uso de SDR para implementación de una estación base simulada en donde se conectarán nuestras UE virtuales. A través de la simulación, será posible llevar a cabo pruebas conexión, tráfico, latencia, facilitando el acceso a tecnología móvil sin recurrir en altos costos.

2.2. Marco Conceptual

En este apartado se abordan conceptos esenciales para comprender el funcionamiento de la red que se implementará en el laboratorio de telecomunicaciones. Se estudiarán temas como la evolución de las redes celulares, las tecnologías de acceso, los dispositivos SDR, el software de programación, los programas utilizados para controlar los equipos SDR y las pruebas de la red LTE-4G.

2.3. Evolución de las redes celulares

Las redes de celulares han evolucionado rápidamente desde su aparición, pasando por distintas generaciones, desde la primera generación (1G) hasta la actual quinta generación (5G) y se encuentra en desarrollo la sexta generación (6G).

- **1G**

La primera generación de telefonía se desarrolló en la década de 1980. Esta red se caracteriza por ser completamente analógica, lo que significa que se usó principalmente para la transmisión de voz, su función consistía en convertir el sonido a señales eléctricas, estableciendo la comunicación entre el emisor y receptor durante una llamada a través de un circuito físico dedicado [9].

- **2G**

La segunda generación, conocida como GSM (Global System for Mobile Communications), apareció por primera vez en 1990. Permitía la transmisión voz y datos mediante la técnica de conmutación de paquetes, una técnica que divide la información se divide en pequeños paquetes transmitidos independientemente por la red.

Utiliza una arquitectura más compleja, con elementos como la BTS (Base Transceiver Statión), el núcleo de red (Core), la MS (Mobile Statión), y los nodos SGSN (Serving GPRS Support Node), entre otros [10].

Operan en bandas de 900 MHz y 1800 MHz que se utiliza en diferentes partes del mundo. Con una de frecuencia son de 200 KHz entre separación de portadoras, lo que permitió incluir servicios digitales adicionales [10].

- **Generación intermedia 2.5 G**

Tras la implementación de GSM, surgieron mejoras denominados como GPRS Y EDGE, consideradas como tecnologías 2.5 G. Aunque nunca fueron oficialmente reconocidos como una generación, representaron una transición a la tecnología a la red UMTS o denominando mundialmente como tecnología 3G [11].

Estas mejoras incrementaron la capacidad de servicios como SMS, permitiendo enviar hasta 30 mensajes por minuto frente a los 6 o 10 permitidos por GSM.

EDGE (Enhanced Data Rates for GSM Of Evolutión), también denominada EGPRS, elevo la velocidad de transferencia hasta 384 Kbps [11].

- **3G**

La tercera generación o 3G, bajo el estándar 3GPP(3rd Generatión Partnership Project), supuso la evolución de la tecnología GSM. Su característica principal esta tecnología fue el soporte para archivos multimedia, voz y video a altas velocidades [11].

La arquitectura UMTS reemplazo a la red de acceso de GSM, aunque mantuvo algunos elementos del núcleo y se reemplazó la red de acceso de radio por una nueva.

Usa canales de 5Mhz frente a los 200 kHz de GSM, lo que permite conectar a más usuarios. Implementa la técnica W-CDMA (Wideband Code División Multiple Access), donde todos los usuarios transmiten simultáneamente información no existe una separación de tiempo ni frecuencia, a cada usuario se le asigna un código un código. La infraestructura de una red UMTS se conforma principalmente por los siguientes componentes:

User Equipment (UE): Se compone de los equipos terminales móviles que estará conectado con la tarjeta SIM.

UTRAN (Universal Terrestrial Radio Access Network): Red de Acceso en UMTS entre terminales móviles y el Núcleo de la red (Core).

Núcleo de la red: Gestiona tráfico, transporte y señalización, e incluye elementos como MSC (Mobile Switching Center), SGSN (Serving GPRS Support Node), GGSN (Gateway GPRS Support Node) [12].

- **4G**

La cuarta generación o LTE (Long-Term Evolutión), se caracteriza de esta tecnología por ser completamente IP, eliminando la técnica de conmutación de circuitos. Soporta velocidades relativas de 100 Mbps para un dispositivo terminal en movimiento, y hasta 1 Gbps en condiciones estacionarias.

Utilizan técnicas TDD (Dúplex por división de tiempo) y FDD (Dúplex por división de Frecuencia) [13]. Así como OFDMA (Orthogonal Frequency-División Multiplexing), en LTE es de 20 MHz por portadora [13].

La arquitectura en LTE se forma principalmente de tres componentes esenciales: EPC (Núcleo de paquetes evolucionado), RAN (Red de acceso radioeléctrico), UE (Equipos de usuario).

El EPC es el encargo de gestionar la mayor parte de la red LTE. Se dividen en varias entidades que como MME (Entidad de gestión de Movilidad), SGW (Nodo de soporte de servicio), PGW (Puerta de enlace a la red de datos por paquetes), y el HSS (Servidor de Suscriptores Locales), y los nodos principales llamado eNodeB (Envolved Nodo B) son los componentes básicos para el funcionamiento correcto de una red LTE [14].

- **5G**

La tecnología 5G, lanzada comercialmente en el año 2019, bajo el estándar de Reléase 15 por 3GPP, incluye una interfaz completamente nueva llamada NR (New Radio), opera con dos modos: modo Non- Standalone, en este modo las tecnologías 4G y 5G pueden funcionar de manera simultánea, y modo Standalone desarrollado únicamente para las tecnologías de 5G, las frecuencias de operación de esta tecnología operan de 3 GHz y 300 GHz, puede alcanzar velocidades relativas superiores a 1 Gbps [15].

La UIT (Unión internacional de telecomunicaciones) clasifica 5G en tres categorías dependiendo su uso:

eMBB (Enhanced Mobile Broadband) para aplicaciones que ocupan un ancho de banda masivo como realidad virtual, transmisión de video, realidad aumentada, entre otros. mMTC (Massive Machine-Type Communications) se utiliza para aplicaciones IOT como realizar mediciones en campos específicos, monitoreo, detección masiva, etc. Y por último URLLC (Ultra-Reliable and Low Latency Communications) es indispensable para servicios que latencia precisa como telemedicina, drones, conducción autónoma, etc. [16].

- **6G**

La tecnología 6G, es la sexta generación de conectividad inalámbrica, se lanzó un plan piloto en china donde se planea realizar pruebas en 2026, a comparación con 5G esta ofrece mayores anchos de banda y velocidades de descarga hasta 100 Gb/s, y se habla de latencias cercanas a cero, tendrá aplicaciones como el desarrollo de IOT, hogares inteligentes, vehículos autónomos, y sistemas de monitoreo en tiempo real [17].

Figura 2 Evolución de las redes celulares en la actualidad

2.4. Introducción a Cuarta Generación

La tecnología anterior a LTE es la 3G, basada en la arquitectura UMTS, esta tecnología fue mejora de la tecnología 2G, UMTS incluye componentes como el núcleo de la red y nueva red de acceso de radio llamado UTRAN, donde los Nodo B y está conectada a la RNC (Radio Network Controller) encargada de gestionar los nodos B. A su vez, el RNC se conectada con redes públicas o privadas como redes de conmutación de circuitos o de paquetes, lo que permite el soporte de redes basadas en protocolo IP. A medida que se avanza se propuso una nueva tecnología [12].

El desarrollo en el año 2004 de la tecnología LTE, pero en el 2009 cuando se lanzó comercialmente por la empresa de telecomunicaciones sueca- finlandesa TeliaSonera, el 14 de diciembre activo por primera una red comercial LTE en las ciudades de Estocolmo y Oslo [18].

La primera red a nivel mundial de LTE fue en Estados Unidos en el año 2010 por la empresa Verizon Communications, cubriendo alrededor de 38 ciudades importantes. Para el año 2016 ya se había

desplegado la tecnología 4G de manera comercial en 170 países, pero a finales de 2022 el número de redes había aumentado a más 800 [18].

A diferencia de UMTS, LTE no se considera su evolución directa, sino más bien una renovación tecnológica, ya que su arquitectura de red está basada en IP (Internet Protocol). Esto significa que solo enfoca dominio de paquetes, soporta más velocidad de datos en lo que se refiere UPLINK y DOWNLINK, permite al usuario utilizar aplicaciones mejoradas a la velocidad, procesamiento de datos [19].

La principal diferencia de la tecnología UMTS y LTE radica en su arquitectura. LTE presenta una más simplificada, conocida como EPS (Sistema de Paquetes Evolucionado), compuesta por dos elementos: EPC (Evolved Packet System) y la E-UTRAN (Evolved Universal Terrestrial Radio Access Network), que representa la evolución del acceso por radio de la tecnología 3G, y optimiza la interconexión entre sus componentes [20].

2.4.1. Arquitectura LTE

La arquitectura de LTE se basa en una arquitectura simplificada y eficiente. Está compuesta por el EPC y E-UTRAN, cuya combinación conforma el sistema EPS, las interfaces S1 que conectaran los e-NB con las entidades y X2 que son las que conectaran los e-NB entre sí, y UE (User Equipment) que son los equipos terminales de los usuarios [19].

La red troncal esta divida en tres subsistemas principales:

Circuit Switched (CS) utilizada tradicionalmente para servicios de voz y video, PS (Packet Switched) dominio basado en la conmutación de paquetes, en el cual se apoya LTE y el subsistema IP Multimedia IMS (IP Multimedia Subsystem) encargado de gestionar servicios multimedia sobre IP, como llamada VoIP y videollamadas [19].

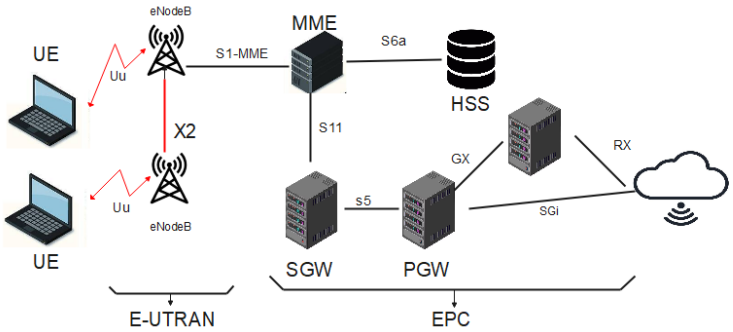


Figura 3 Arquitectura LTE

2.4.2. Arquitectura E-UTRAN

La arquitectura de la red de acceso en LTE, conocida como E-UTRAN, se diferencia de UMTS en los nodos principales, que pasan de llamarse Nodo B a eNodeB (Evolved NodeB). Este componente asume funciones que antes pertenecían al RNC (Radio Network Controller) en UMTS. En LTE es la red es más autónoma ya que los eNodeB asumen estas funciones directamente, lo que reduce las interferencias y mejora la calidad de servicio, permitiendo a los usuarios una experiencia más rápida y con menor latencia, administra los recursos de radio ya que mantiene una conexión vía radio entre eNodeB y el equipo de Usuario.

Los eNodeB se comunican entre si mediante la interfaz lógica X2, que permite ejecutar funciones como el Handover (traspaso de usuario entre celdas), garantizando la movilidad sin interrupciones. Esta interfaz también se utiliza para gestionar las interferencias entre celdas y garantizar una movilidad sin riesgo de perdidas al transferir paquetes entre celdas vecinas. La red de radio LTE está directamente conectada con el núcleo de red EPC, mediante la interfaz lógica S1. El núcleo de red EPC se divide en Plano de control y el Plano de usuario [21].

El plano de control se gestiona a través de S1-MME, mientras que el plano de usuario utiliza S1-U. Esta estructura de red se ha diseñado de esa manera para mantener separadas las pilas de protocolos que se asocien a las interfaces de la red LTE, además de permitir la comunicación entre los eNodeB y distintos nodos de la red [22].

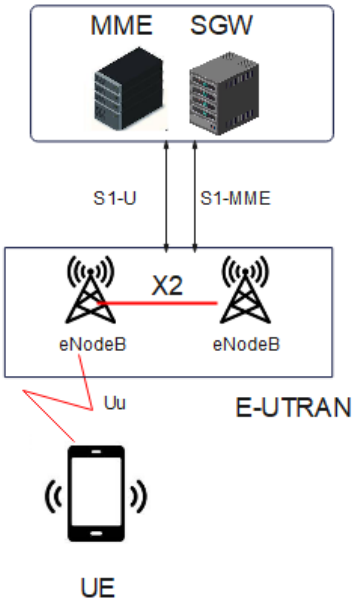


Figura 4 Arquitectura de la red de acceso E-UTRAN

2.4.3. Arquitectura de la red troncal EPC

El EPC (Evolved Packet Core) representa el núcleo de la red LTE, cuya principal característica es la gestión de todos los servicios a través del protocolo IP lo que optimiza su transporte. Además, permite la separación entre planos de usuario y de control permitiendo una mayor eficiencia en la gestión del tráfico. El EPC está compuesto por varias entidades, las cuales se describirán a continuación [23].

2.4.3.1.MME

Es la responsable de gestionar funciones del plano de control, facilitando el acceso de los suscriptores y la administración de sesiones [23].

Sus principales funciones incluyen:

- Seguridad: Autenticara a los usuarios que se conecten a la red, verificando la protección y el cifrado de los datos [24].
- Gestión de sesiones: Controla la señalización que se relacionan con QoS (Calidad de Servicio), creando paquetes de datos [24].
- **Localización de equipos inactivos:** Supervisa la actualización de área para permitir que los terminales accedan a la red cuando tienen sesiones activas[24].

2.4.3.2. HSS

Otra entidad dentro de la arquitectura LTE es el HSS (Home Subscriber Server), un servidor encargado de almacenar la información de los suscriptores de la red móvil. Integra funciones de AAA (Autenticación, autorización, contabilidad) [25].

EL HSS es responsable de autenticar a los usuarios y proporcionar la información necesaria a otras entidades cuando un equipo terminal se conecta a la MME, compara los datos recibidos con los almacenados en la base de datos [22]. Entre los datos que gestiona el HSS se encuentran:

- **Identificación y direccionamiento de dispositivos**: Almacena códigos IMSI y números de teléfono de los usuarios [25].
- **Información del usuario:** Incluye datos sobre la subrepción a servicios y la calidad de servicio (QoS) asignada [25].

2.4.3.3. SGW

El SGW (Serving Gateway) es el encargado del enrutamiento del tráfico de datos en la red LTE, asegurando su correcta entrega al destino. Esta entidad se comunica directamente con la MME para

establecer y mantener la conexión entre los dispositivos móviles y la red 4G LTE. Cuando un equipo terminal se conecta a la MME, esta asigna la SGW para encargar de administrar el tráfico de datos del dispositivo [25].

2.4.3.4. PGW

EL PGW (Packet Data Network Gateway) se encarga de asignar de direcciones IP a los dispositivos móviles y gestionar de la calidad de servicio (QoS) en la transmisión y recepción de datos. Se comunica con el SGW para garantizar la movilidad y continuidad de la sesión de los dispositivos. Administra las políticas de red y facturación [25].

En el roaming de datos internacional, el PGW facilita la internet para los usuarios que viajan al extranjero, permitiendo la comunicación entre operadoras de diferentes países. Establece túneles GTP (GPRS Tunneling Protocol) entre el SGW en la red visitada y el PGW de la red de origen asegurando la prestación del servicio [25].

2.4.3.5. PCRF

El PCRF (Policy and Charging Rules Functión) es una entidad encargada de administrar la calidad de servicio (QoS) al definir políticas específicas para cada usuario.

Entre estas políticas se incluyen el ancho de banda asignado al usuario, la prioridad de servicios y las restricciones, optimizando así la experiencia del usuario [25].

Permite diferenciar los servicios y aplicar políticas personalizadas según los acuerdos establecidos con cada usuario. El PCRF se comunica con el PGW, intercambiando información en tiempo real sobre las políticas y tarifas, lo que permite un control sobre los servicios proporcionados [25].

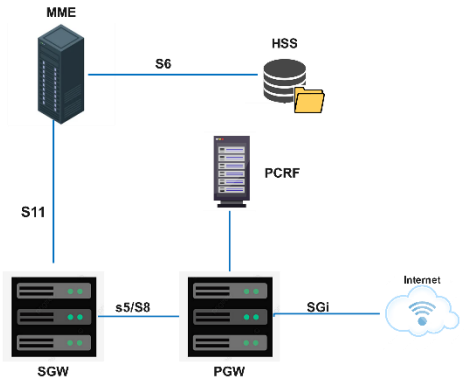


Figura 5 Red EPC

2.4.4. Bandas de Frecuencia

Las bandas de frecuencias se dividen en regiones específicas, con respecto a LTE también puede variar según el estado u operador que distribuya el servicio [26]. En el mundo las bandas de frecuencia se reparten por continentes y ocupan diferentes bandas según se requiera [27].

Banda (FDD)	Nombre	Frecuencia (MHz)	Región 1 (EMEA)	Región 2 (América)	Región 3 (Asia Pacifico)
1	2100	2100	✓		✓
2	PCS	1900		✓	
3	1800	1800	✓		✓
4	AWS	1700/2100		✓	
5	850	850		✓	✓
7	2600	2600	✓		
8	900	900	✓		✓
12	700 a	700		✓	
13	700 c	700		✓	
17	700 b	700		✓	
20	800 DD	800	✓		
28	700 APT	700		✓	✓

Tabla 1 Asignación de Bandas FDD

En América del SUR las bandas de frecuencias que se ocupan son las siguientes:

Banda (TDD)	Nombre	Frecuencia (MHz)	Región 1 (EMEA)	Región 2 (América)	Región 3 (Asia-Pacifico)
38	2600	2600	✓	✓	✓
40	2300	2300	✓		
41	2500	2500-2690	✓	✓	✓
42	3500	3400-3600	✓		✓
43	3700	3600-3800	✓		

Tabla 2 Asignación de Bandas TDD

2.4.4.1. Banda de 700 MHz

Este rango de frecuencias se popularizó en servicios móviles tras el dividendo digital que consistió en la transición de la televisión analógica a digital. Al operar por debajo de 1 GHz, ofrece 28 canales de propagación y permite grandes anchos de banda, características que lo hacen ideal para la cobertura en interiores y un área extensa [28]. Esta banda fue licitada en 2016 y se ha convertido en una de las más utilizadas por los operadores de internet móvil [29].

2.4.4.2. Banda de 1900 MHz

Conocida como la banda 2 en LTE, esta frecuencia es utilizada debido a su capacidad para proporcionar servicios de voz y datos en redes LTE. Es compatible con tecnologías anteriores como UMTS y GSM, lo cual facilita su interoperabilidad y uso eficiente del espectro. Aunque se emplea principalmente en áreas urbanas y rurales, su desempeño es más eficiente en entornos con una densidad moderada de usuarios [30].

2.4.4.3. Banda de 1700-2100 MHz

Conocida como AWS-1, esta banda opera en 1700 MHz para enlace ascendente (UPLINK) y 2100 para enlace descendente (DOWNLINK). Es utilizada en Estados Unidos y países de Latinoamérica debido a su buen equilibrio entre capacidad y cobertura. Puede emplearse como una banda única o en configuraciones de red multibanda, permitiendo la agregación de portadoras para mejorar el rendimiento de la red [31].

Figura 6 Países que utilizan LTE B4

FUENTE: IMAGEN TOMADA DE [31]

2.4.4.4. Banda de 2500 MHz

Denominada como banda de frecuencia 4G LTE B41, comprende frecuencias que van desde los 450 MHz hasta los 3.8 GHz.

Permite anchos de banda hasta 20 MHz mediante la agregación de portadoras. Soporta LTE tecnología por división de tiempo (TDD), lo cual permite un uso más eficiente del espectro. La disponibilidad y uso de banda de frecuencia varía según la región y el operador [32].

2.4.4.5. Tecnología LTE en Ecuador

El consejo nacional de telecomunicaciones (CONATEL) aprobó el despliegue de la tecnología 4G con las frecuencias de 700 MHz, 1900 MHz, 1700-2100 MHz (AWS) en el territorio ecuatoriano. Diferentes operadores utilizan este espectro para ofrecer diferentes servicios como Movistar, Claro, CNT y Tuenti. Este último no se encuentran registros al ser un operador virtual por parte de Movistar [34].

La Corporación nacional de Telecomunicaciones (CNT EP) comenzó a desplegar la desplegar tecnología 4G con 30 MHz de espectro en la banda de 700 MHz y 40 MHz en la banda de AWS esta banda corresponde a 1700- 2100 MHz por autorización de la CONATEL [34].

Movistar implementó su red 4G en Ecuador en el 2015 en la banda 1900 MHz con 50 MHz de espectro, mientras que a Claro se le concedió 40 MHz en la banda de 1700 (AWS) y 20 MHz en la banda de 1900 para el despliegue de esta tecnología [34].

Operador	Banda	Nombre común	Frecuencia	Tipo
Movistar	2	PCS	1900 MHz	FDD
Claro	4	AWS	1700/2100 MHZ	FDD
Claro	2	PCS	1900 MHz	FDD
CNT	4	AWS	1700/2100 MHz	FDD
CNT	28	APT	700 MHz	FDD
Tuenti	2	PCS	1900 MHz	FDD

Tabla 3 Operadores de redes móviles en Ecuador

2.4.5. Interfaces en LTE

En LTE, las interfaces permiten la comunicación eficiente entre los distintitos componentes de la red EPC y E-UTRAN, cada interfaz cumple funciones específicas que garantizan el correcto funcionamiento de la red [25], estas son:

- SG1: Es responsable de conectar el PGW con las redes externas basadas en IP, como internet.
- S5: La función principal es transmitir paquetes de datos entre el SGW y el PG.
- S11: Coordina la transferencia de canales portadoras durante los procedimientos de handover inter-nodos.
- S10: Permite que un MME transfiera el contexto de un usuario a otro MME
- S6a: Facilita la transferencia de información de la base de datos de usuario entre el HSS y el MME
- S1-U: Proporciona el transporte de información del plano de usuario entre eNodeB y el EPC
- S1-MME: Permite la comunicación entre eNodeB y el MME, transmitiendo paquetes de control necesarios para establecer, modificar y liberar recursos en la interfaz de radio, además de asistir en los procesos de Handover [25].

2.4.6. Canales en LTE

La estructura de los canales en LTE se organiza en tres capas principales: Canales físicos, Canales de Transporte y Canales Lógicos. Cada uno de ellos cumple con funciones específicas para garantizar la comunicación eficiente entre el UE y la red

2.4.6.1. Canales Físicos

Los canales físicos se generan en la capa 1 y se encargan principalmente de la sincronización del sistema. Su función es transportar datos provenientes de capas superiores, incluyendo la información de control, configuración y datos de usuario Los canales físicos varían entre el enlace ascendente y descendente [35]. Entre los canales físicos en el enlace descendente podemos encontrar:

- PBCH (Canal de transmisión física): Transporta la información maestra (MIB), asignada desde el BCH
- PDSCH (Canal compartido de enlace descendente físico): Lleva datos de usuario e información de control, mapeada desde el DL-SCH
- PDCCH (Canal de control de enlace descendente físico): Transporta información de programación para los equipos de usuario
- PCICH (Canal indicador HARQ físico): Transmite mensajes ACK/NACK para la transmisión en el enlace ascendente

- PCFICH (Canal indicador de formato de control físico): Indica la cantidad de símbolos OFDM utilizados para la información de control
- PMCH (Canal de difusión física): Transporta servicios de multidifusión.

Los canales físicos en el enlace descendente se dividen en:

- PUSCH (Canal compartido de enlace ascendente físico): Transporta datos del usuario y señalización al enlace ascendente
- PUCCH (Canal de control de enlace ascendente físico): Lleva información de control de enlace ascendente, como los mensajes ACK/NACK, indicadores de calidad del canal y programación
- PRACH (Canal físico de acceso aleatorio): Transmite preámbulos de acceso aleatorio para el acceso inicial y las transferencias

2.4.6.2. Canales de Transporte

Los canales de transporte ofrecen información de transferencia a la capa de control de acceso al medio (MAC) y a las capas superiores [36], estos son:

- PCH (Canal de búsqueda): Transporta mensajes de búsqueda asignados desde el PCCH
- BCH (Canal de transmisión): Lleva información del sistema asignados desde el BCCH
- DL-SHC (Canal compartido de enlace descendente): Encargado del transporte de datos del usuario, asignado desde DTCH y DCCH
- MCH (Canal de Multidifusión): Transporta servicios de multidifusión, asignados desde el MTCH y MCCH

Para el enlace ascendente los canales encargados son:

- UL-SCH (Canal compartido de enlace ascendente): Utilizado para datos de usuario y control
- RACH (Canal de acceso aleatorio): Se emplea para el acceso inicial y la resolución de contenciones.

2.4.6.3. Canales Lógicos

Estos canales se clasifican en canales de control y canales de tráfico [36], se describen como:

- **Canales de control:**

- PCCH (Canal de control de paginación): Pagina al UE en modo inactivo
- BCCH (Canal de control de transmisión): Transmite información del sistema

- CCCH (Canal de control común): Utilizado para la configuración inicial y acceso aleatorio
- DTCH (Canal de tráfico dedicado): Transporta datos de usuarios dedicados
- DCCH (Canal de control dedicado): Transmite información de control para un UE especifico.
- MTCH (Canal de trafico de multidifusión): Transporta datos de multidifusión
- MCCH (Canal de control de multidifusión): Encargado de transmitir información de control para servicios de multidifusión.

➢ **Canales de tráfico:**

- CCCH (Canal de control común): Se utiliza para solicitudes de conexión y acceso aleatorio inicial
- DTCH (Canal de tráfico dedicado): Transporte datos de usuarios dedicados
- DCCH (Canal de control dedicado): Se utiliza para la señalización de control de enlace ascendente

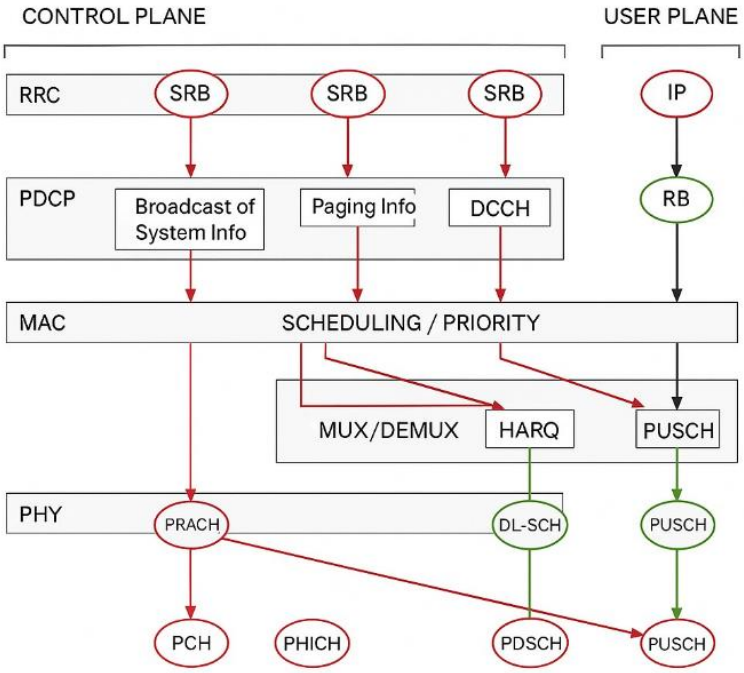


Figura 7 Arquitectura de Canales en LTE

FUENTE: IMAGEN TOMADA DE [37]

2.4.7. Técnicas en LTE

En LTE, se utilizan diferentes técnicas de transmisión para maximizar el rendimiento del ancho de banda. Esta diferenciación en las técnicas busca reducir la perdida de paquetes y mitigar los efectos negativos de la propagación por múltiples trayectorias, las técnicas de acceso que usa LTE, son OFDMA para el enlace descendente y, SC-FDMA para el enlace ascendente [25].

2.4.7.1. OFDMA

La técnica de acceso OFDMA asigna bloques temporales de subportadoras a cada usuario, conocido como resource block (PRB) y ofrece una mayor flexibilidad para gestionar la calidad de servicio (QoS) [37].

Cada subportadora está separada por 15 kHz y se caracteriza por tener un PARP (Peak-to-Average Power Ratio) elevado, lo que requiere el uso de amplificadores con un alto consumo energético.

Por esta razón, OFDMA se utiliza en el canal de bajada, las operadoras disponen de equipos capaces de soportar alto consumo energético [37].

Esta técnica divide el canal de comunicación en múltiples sub-canales, permitiendo que varios usuarios transmitan simultáneamente a través de frecuencias específicas, mejorando el rendimiento de la transmisión[38]. OFDMA facilita la implementación de la técnica de Scheduling, permitiendo asignar más recursos para transmitir datos, con solo modificar los símbolos introducidos en los PRB.

La principal ventaja OFDMA es su tiempo de respuesta reducido, mejorando la eficiencia del sistema. Durante el proceso de Scheduling, se considera la calidad del canal de cada usuario o la calidad de servicio, optimizando el uso de recursos disponibles [37].

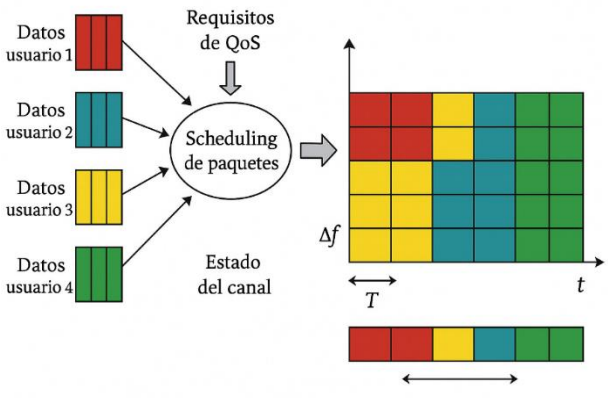


Figura 8 Asignación de recursos en OFDMA

FUENTE: IMAGEN TOMADA DE [37]

2.4.7.2. SC-FDMA

La técnica de acceso SC-FDMA fue desarrollada como una alternativa al elevado consumo energético que posee OFDMA, la organización 3GPP estableció el uso de SC-FDMA para el canal de subida (UPLINK) [37].

Para reducir el consumo energético, SC-FDMA utiliza una única portadora como canal de subida, y aplica una recodificación en los símbolos transmitidos. Este método emplea la transformada discreta de Fourier (DFT) seguida de la transformada inversa de Fourier (FFT), permite disminuir el PAPR (Peak-to-Average Power Ratio) en la señal final señal final, mejorando así la eficiencia energética y optimizando el rendimiento del sistema [37].

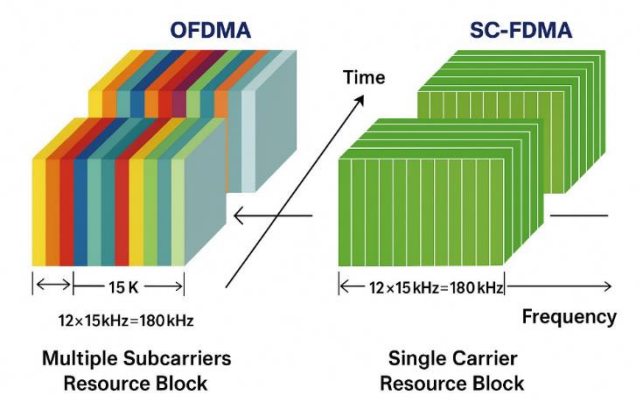


Figura 9 Comparativa entre OFDMA y SC-FDMA

FUENTE: IMAGEN TOMADA DE [37]

2.4.7.3. Enlace descendente y enlace ascendente

Para el enlace descendente en LTE, mediante la técnica de OFDMA fragmenta la información transmitida en múltiples flujos de datos más lentos, los cuales se envían a simultáneamente través de subportadoras ortogonales.

El uso de subportadoras de banda estrecha contribuye a la estabilidad dentro de cada sub-banda, debido a la ortogonalidad de las subportadoras, estas pueden solaparse sin interferencia, lo que mejora la eficiencia espectral. Esto quiere decir que, el equipo terminal del usuario modula únicamente las subportadoras necesarias, y dejando libre aquellas que no utiliza para que otros usuarios puedan aprovecharlas, optimizando así el uso del espectro

La técnica de SC-FDMA empleada en el enlace ascendente de LTE es una versión precodificada, es decir que los símbolos en el dominio del tiempo se modulan utilizando esquemas como QPSK,

16QAM, o 64QAM. Luego se convierten al dominio de la frecuencia mediante la DFT (Transformada discreta de Fourier) se distribuyen a través de las subportadoras dentro del ancho de banda del usuario. Esta técnica reduce el PARP (Peak-to-Average Power Ratio) lo que mejora la eficiencia del HPA (Amplificador de alta potencia) y reduce un menor consumo la batería del dispositivo.

Con respecto a OFDMA, transmite datos en paralelo por subportadora, la función de SC-FDMA los va a transmitir en serie a una tasa cuatro veces mayor, con cada símbolo ocupando 15 kHz, lo que permite una mayor eficiencia en el enlace ascendente en LTE [25].

2.4.8. Link Budget

Este proceso considera tanto ganancias como las perdidas pérdidas de la señal transmitida por un medio de transmisión, evaluando su rendimiento en los enlaces ascendentes y descendentes, para garantizar la comunicación entre la estación base (eNodeB) y el equipo de usuario (UE) [39].

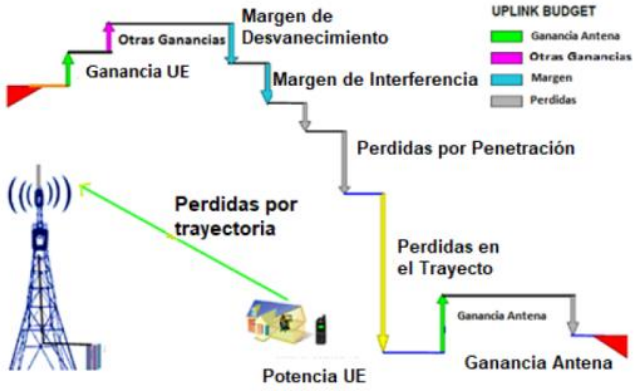


Figura 10 Link Budget para UPLINK

FUENTE: IMAGEN TOMADA DE [39]

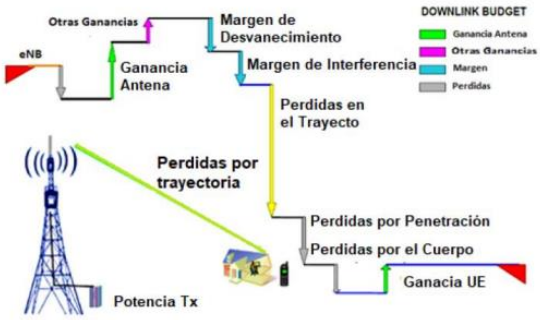


Figura 11 Link Budget para-DOWNLINK

FUENTE: IMAGEN TOMADA DE [39]

2.4.9. Arquitectura del protocolo de radio en LTE

La arquitectura de protocolos en LTE se dividen en dos planos los cuales son: plano de control y plano de usuario

2.4.9.1.Plano de usuario

Este plano se responsable de la transmisión de información entre el UE y eNodeB, así como entre eNodeB y EPC. [25] Los protocolos empleados en la comunicación entre el UE y el eNodeB incluyen [40]:

- Capa física (PHY): Es la capa más baja, encargada de transmitir la información a través del canal radio. Gestiona la modulación, codificación y asignación de recursos de radio.
- Packet Data Convergence Protocol (PDCP): Se encarga de la compresión de paquetes IP y facilita la transferencia de estos a través de la red.
- Radio Link Control (RLC): Responsable de la segmentación, control de errores, y transmisión segura paquetes PDCP entre eNodeB y el equipo de usuario.
- Medium Access Control (MAC): Administra el acceso de los usuarios al medio de transmisión y realiza la multiplexación de los datos de distintos usuarios.

2.4.9.2. Plano de Control

En el plano de control se sitúa entre el equipo de usuario y la red, operando en la misma capa de enlace de datos y en la capa física en el plano de usuario, los protocolos utilizados en el plano de control son:

- Radio Resource Control (RRC): Establece y gestiona la conexión entre el UE y la estación base [25].
- NAS: Protocolo utilizado entre el MME y el terminal móvil. Se encarga de la autenticación, autorización y gestión de movilidad de los dispositivos [23]

Para la comunicación entre eNodeB y el EPC, se emplea la interfaz S1-MME, donde se implementan protocolos como S1-AP (S1- Applicatión Part), SCTP (Stream Control Transmissión Protocol) e IP (Internet Protocol) [25]

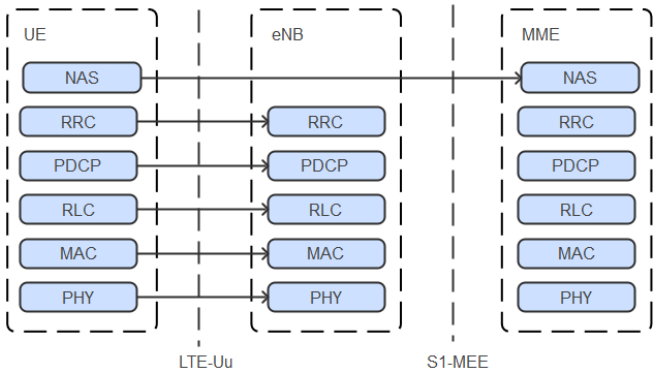


Figura 12 Protocolos utilizados en LTE

2.4.10. Estructura de una trama en LTE

Un bloque de recursos representa la unidad mínima de recursos que puede asignar un e nodo B a un terminal móvil en un enlace ascendente o descendente. En la capa física, se le denomina PRB (Physical Resource Block) [41].

Cada bloque de recursos tiene una duración de 0,5 ms, está compuesto por 12 subportadoras, las cuales están separadas por 15 KHz, dando un total de 180 KHz en el dominio de la frecuencia. Esta relación se está relacionada con el ancho de banda que existe en cada PRB. [42].

$$BW_{PRB} = 12\ Subportadoras\ \cdot 15\frac{kHz}{Subportadoras} = 180\ Khz$$

El ancho de banda de operativo en LTE es flexible y depende de la cantidad de bloques de recursos (PRB) que se asignen, ya que al modificar el número de PRB, se modifica el ancho de banda disponible, lo que permite adaptarlo a distintos requerimientos dentro de la red según las necesidades específicas [42].

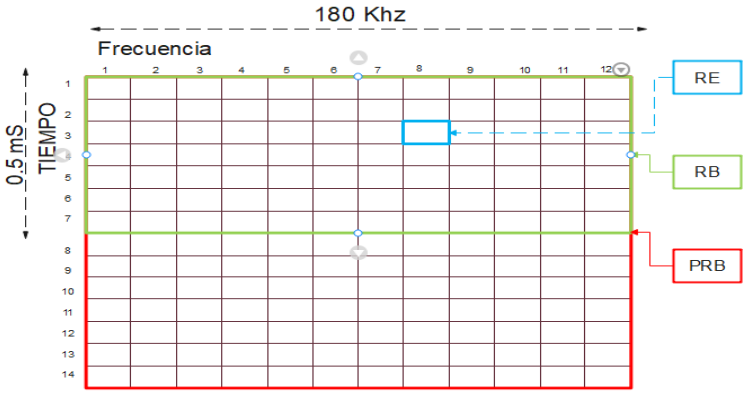


Figura 13 Estructura de una Trama

2.4.11. Seguridad acceso a la red

Es importante establecer políticas de seguridad al tener una red que contiene información importante y sensible para garantizar la integridad y privacidad de los datos a través de las redes LTE por ello se crearon diferentes mecanismos para salvaguardar esta información.

2.4.11.1. Autenticación en LTE

El proceso de autenticación en LTE es un mecanismo en LTE basado en el protocolo EPS-AKA (Evolved Packet System Autenticación and Key Agreement). Esta entidad recibe claves de seguridad de alto nivel desde el HSS para garantizar el acceso a la red [43]. Para autenticar la identidad del usuario, el equipo de usuario (UE) y la MME generan una clave compartida denominada K_ASME, que sirve como base para derivar otras claves utilizadas en la protección de datos y la integridad de mensajes intercambiados [44].

2.4.11.2. Seguridad NAS (Non Access Stratum)

Este mecanismo protege los mensajes de señalización entre el equipo de usuario y la MME. Una vez completada la autenticación, se generan claves K_NASenc y K_NASint a partir de K_ASME [43]. Se derivan en dos claves:

- La función de la clave K_NASenc se emplea para el cifrar mensajes NAS, garantizando la confidencialidad de la comunicación
- K_NASint se utiliza para proteger la integridad de los mensajes, asegurando no sean modificados durante la transmisión [44].

2.4.11.3. Seguridad AS (Access Stratum)

La seguridad AS (Access Stratum) es la encargada de establecer y la proteger la conexión en el plano de usuario y en el plano de control entre el equipo de usuario y eNodeB [45]. Una vez que configurada la seguridad NAS, se generan claves específicas para cifrar y proteger los datos trasmitidos en el enlace de radio, evitando su alteración o intercepción [44].

2.5. Equipos de Usuario (UE)

Los equipos de usuario son componentes importantes en la tecnología LTE, permiten la conexión a través de la interfaz de radio. Estos equipos facilitan las comunicaciones dentro de la red LTE [23].

2.5.1. Teléfonos móviles

En 1973 se inventó el primer teléfono móvil, desarrollado por motorola en 1973, pesaba aproximadamente un 1,1 KG, permitía 30 minutos de llamadas y requería unas 10 horas de carga, en la actualidad los teléfonos son más compactos son óptimos de transportar a cualquier lugar [46].

Los teléfonos móviles comenzaron a popularizarse, alcanzando alrededor de 11 millones de usuarios, en los años 90, esto hizo crecer de una manera acelerada a la industria de teléfonos móviles grandes empresas empezaron a crear sus propias versiones, como Nokia que lanzo el Mobira Cityman 900 [46].

En el año 1992, se introdujo el primer teléfono con acceso a GSM, el Nokia 1011, y en 1994 surgió el primer smartphone con pantalla táctil y aplicaciones. Con el paso de los años, se incorporan innovaciones como pantallas en color, cámaras fotografías, y acceso a servicios avanzados como YouTube y Google Maps. Los teléfonos inteligentes integran en la actualidad inteligencia artificial y una amplia variedad de aplicaciones que permiten la conectividad a internet, llamadas y múltiples funciones avanzadas [46]

2.5.2. Módulo de suscripción de usuario (SIM)

La tarjeta SIM (Subscriptor Indentity Module), es un circuito integrado con su propio sistema operativo y estructura de archivos. La función principal de las tarjetas SIM es almacenar la información requerida para la autenticación, y acceso a la red móvil, incluyendo algoritmos de seguridad e identificadores como el IMSI (Identidad Internacional de Suscritor móvil) [47].

El IMSI es un identificador único a nivel global compuesto por aproximadamente 15 dígitos. Su propósito es facilitar la señalización y la transmisión de mensajes de redes de telefonía móvil [47].

2.6. Software Defined Radio (SDR)

El SDR es un radio reconfigurable, que define su funcionalidad por medio de software[50]. El SDR nos permite utilizar un mismo hardware para diferentes funciones en distintos momentos.
El procesamiento de señales en un radio convencional es realizado en hardware por lo que no se puede cambiar, ni alterar su diseño. Por otro lado, el SDR resulta muy práctico para un rango muy extenso de aplicaciones, es más existen casos excepcionales en donde la capacidad de alterar la funcionalidad de radio en tiempo real es muy alta.
Como ya se mencionó en el Capítulo I en el apartado Antecedentes esta tecnología fue propuestas por Joseph Mitola III y tuvo su desarrollo en el departamento de defensa de EE. UU. para lograr

comunicarse en diferentes bandas de frecuencia con un solo equipo, asegurando así compatibilidad entres sistemas de los diferentes frentes.

2.6.1. Filosofía y Evolución

El propósito de un SDR es no depender de la parte electrónica en dispositivos de radiocomunicación, es decir que las funciones que realizaba el hardware de estos equipos ahora estén a cargo de un software. Esto nos proporciona una disminución de gastos en equipos que se emplearan, además de la posibilidad de trabajar en un mismo dispositivo con diferentes protocolos.
Gracias a su arquitectura hibrida, uniendo hardware y software en un solo dispositivo, haciendo que funciones, como la modulación y demodulación ya no dependan de la electrónica del dispositivo en cuestión.[23]

La Figura 14 Simplifica la composición de un transceptor, y explica cuáles son aquellos que se desarrollan a través de software, además de aquellos que siguen siendo componentes físicos.

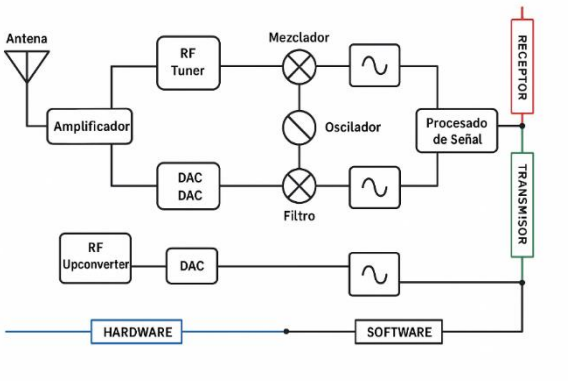


Figura 14 Estructura del transceptor: elementos y modos de implementación

FUENTE: IMAGEN TOMADA DE [23]

Observamos que el transmisor se ubica en la parte superior, mientras que el receptor se ubica en la parte inferior.

Comenzando con la transmisión, el primer paso es el procesamiento de la señal a transmitir, estamos haciendo referencia al proceso de modulación y codificación. A continuación, se trasladará la señal de banda base hacia una frecuencia intermedia y mediante el Digital Analog Converter (DAC), la señal digital pasará a ser analógica.

Luego, la señal analógica pasara al RF Converter para convertirla en frecuencia del espectro RF (este va de los 3 Hz hacia los 300 GHz), para luego ser incrementada por el amplificador y ser entregada a la antena para su envió.

Para la recepción, la señal recibida es enviada hacia un amplificador de bajo ruido (Low Noise Amplifier (LNA)). Tras finalizar esto el Tuner RF transformará la frecuencia RF a una frecuencia intermedia y el ADC será el encargado de digitalizar la señal. Una vez hecho esto se traslada la señal a banda base y pasa por un filtro. Para Finalizar se realizan el procesamiento de la señal.

Cabe recalcar que toda esta parte de software que pertenece a la arquitectura SDR, no es más que la SCA, que tal y como se menciono tuvo gran impacto en las comunicaciones militares de los Estados Unidos de América, siendo considerada un estándar en las Telecomunicaciones militares.

SCA se compone por módulos de software que provienen de aplicaciones ya creadas anteriormente y cuyas dependencias están definidas.

Es una arquitectura distribuida capaz de ser ejecutar sus módulos en diversos componentes, esto es gracias a su estándar Common Object Request Broker Arquitecture (CORBA (Millware estandarizado por el grupo Object Management Group (OMG)))[51].

SCA posee un entorno para sus módulos, además de definir un protocolo en los mismo. El Core Framework (CF) especifica sus servicios e interfaces, esto bajo un fichero Extensible Markup Language (XML), que decreta sus parámetros y formato.

Las interfaces de CF tienen una clasificación de cuatro grupos:

- Base Applicatión Interfaces: Nos permite controlar y gestionar módulos de implementaciones realizadas.
- Base Device Interfaces: Ayuda a controlar y gestionar el hardware de manera más sencilla.
- Framework Control Interfaces (FCI): Permite la gestión de las aplicaciones, además de la opción de eliminar a las mismas
- Framework Service Interfaces (FSI): Aporta funciones que nos permiten implementar ficheros.

A continuación, la Figura 15 nos presenta las capas que pertenecen a la arquitectura SDR.

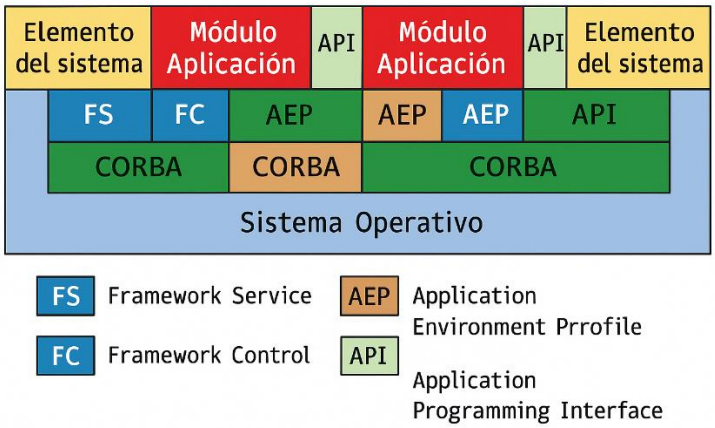


Figura 15 Arquitectura SDR

FUENTE: IMAGEN TOMADA DE [25]

Para Finalizar mencionaremos los retos que enfrentan los SDR[10]:

- **Requisitos Computacionales:** Es el desafío más importante de los SDR. La importancia de tener una capacidad computacional de acorde a lo que queramos, sin comprometer el consumo energético, el peso y el tamaño.

Si se desea desplegar terminales móviles por medio de SDR, las condiciones adecuadas serian: un consumo energético bajo, para garantizar un tiempo prolongado de las baterías en los dispositivos. De la misma manera el equipo debe ser pequeño para mayor movilidad y lo suficientemente grande para evitar sobrecalentamientos.

Por otro lado si lo que se desea es el despliegue de una Estación Base, estas condiciones pierden relevancia, esto porque dicho elemento de la red celular puede permanecer estático. Es por lo que se podría conectar directamente a la red eléctrica, siendo irrelevante su consumo energético.

- **Plataformas de hardware:** Los SDR permiten el despliegue de redes por medio de software, pero, aunque posee esta ventaja, no deja de depender de un circuito Físico con electrónica dedicada a funciones determinadas.

- **Seguridad:** La tecnología SDR permite la alteración de su software. Debido a esto la seguridad es altamente vulnerable, esto porque existe el riesgo de modificación por parte de terceros

Otro problema es la instalación de software malicioso capaz de corromper el sistema operativo del terminal donde se aloja, además de la posibilidad de extenderse al resto del sistema. Además de la implementación de código malicioso que puede usarse de puerta trasera para la filtración de datos de usuario, que residen en el terminal.

La mitigación de este problema es realizada oír medio de la criptografía asimétrica, la cual genera pares de claves publica-privadas que garantizan la autenticidad del creador y la integridad de su código. Antes del envío del código se realiza un hash de este y se firma con una clave generada previamente. A continuación, la Figura 16 muestra este proceso:

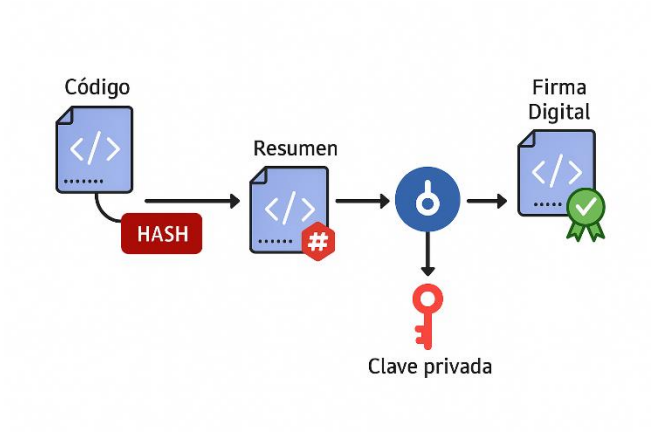


Figura 16 Proceso de firma del código SDR

FUENTE: IMAGEN TOMADA DE [26]

La plataforma envía el código y su firma para la validación, asegurando así que el software no fue modificado por un tercero.

El proceso de validación se verifica por medio de 2 procesos: primero se realiza el hash del código recibido, como segundo proceso se obtiene de la firma un resumen del código original. Para finalizar la comprobación se debe verificar que los 2 resúmenes del código sean idénticos.

En caso de diferir, el programa detectara la modificación o alteración del código hecha por un tercero.

Esto se puede resumir en la Figura 17:

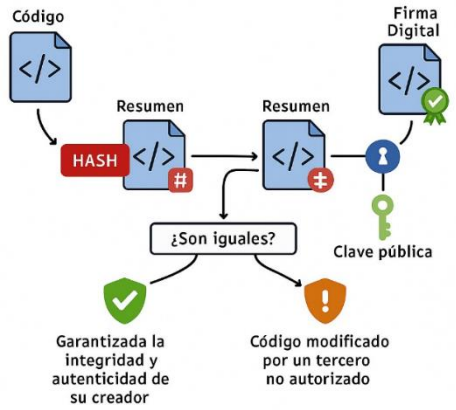


Figura 17 Verificación de la firma del código SDR

FUENTE: IMAGEN TOMADA DE [26]

2.6.2. Equipos SDR

Los equipos SDR permiten la implementación de estaciones base LTE mediante software, eliminando la necesidad de hardware propietario y costoso.

A continuación, se detallan los modelos BladeRF x40, BladeRF x9, BladeRF x5 y USRP.

2.6.2.1. Blade RF micro 2.0 XA4

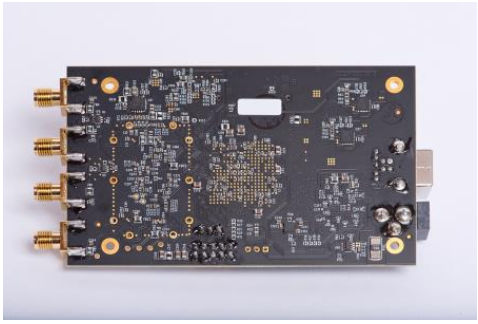


Figura 18 Blade RF micro 2.0 xA4

FUENTE: IMAGEN TOMADA DE NUAND.COM

El BladeRF Micro 2.0 XA4 es una radio definida por software (SDR) de alto rendimiento, compacta y potente, diseñada para aplicaciones avanzadas de comunicación inalámbrica como LTE, 5G, GSM, WiFi, y GNSS.

Es ideal para investigadores, desarrolladores y profesionales que requieren capacidades SDR robustas en un factor de forma reducido.

Gracias a su potencia de procesamiento, amplio rango de frecuencias y compatibilidad con software libre, el BladeRF Micro 2.0 XA4 es ideal para quienes buscan desarrollar o experimentar con tecnologías LTE, 5G y redes inalámbricas avanzadas con bajo costo relativo frente a equipos comerciales[52].

2.6.2.2.Blade RFx40

Figura 19 Blade RFx40

FUENTE: IMAGEN TOMADA DE NUAND.COM

El BladeRF x40 es una radio definida por software de bajo costo diseñada para aplicaciones de comunicación inalámbrica como LTE y GSM[53].

Es ideal para estudiantes, investigadores y desarrolladores que buscan implementar y probar redes celulares.

Características principales:

- Rango de frecuencia: 300 MHz – 3.8 GHz
- Ancho de banda de muestreo: 40 MHz
- FPGA: Cyclone IV de 40K LE
- Interfaz USB 3.0 con baja latencia
- Admite MIMO 1x1 (expandible a 2x2 con módulo adicional)
- Compatible con OpenAirInterface (OAI) y srsRAN

Su bajo costo, buena documentación y compatibilidad con software libre, es ideal para aplicaciones LTE de bajo presupuesto.

2.6.2.3. Blade RFx9

Figura 20 BladeRF 2.0 micro xA9

FUENTE: IMAGEN TOMADA DE NUAND.COM

El BladeRF x9 es una versión mejorada del x40, con capacidades más avanzadas para aplicaciones LTE y 5G. Su mayor ancho de banda lo hace adecuado para sistemas con mayor demanda de datos[54].

Características principales:

- Rango de frecuencia: 47 MHz – 6 GHz
- Ancho de banda de muestreo: 56 MHz
- FPGA: Cyclone V de 301K LE
- Interfaz USB 3.0 y PCIe Gen 2
- Admite MIMO 2x2
- Mejor rendimiento en LTE y redes experimentales 5G

Posee una mayor cobertura de frecuencia, ideal para estaciones base LTE avanzadas y experimentación con 5G.

2.6.2.4. Blade RFx5

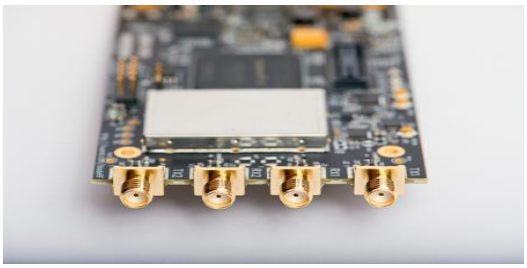


Figura 21 BladeRF 2.0 micro xA5

FUENTE: IMAGEN TOMADA DE NUAND.COM

El BladeRF x5 es un modelo con menor capacidad que el x40, pero sigue siendo útil para redes pequeñas o experimentales[55].

Características principales:

- Rango de frecuencia: 300 MHz – 3.8 GHz
- Ancho de banda de muestreo: 28 MHz
- FPGA: Cyclone IV de 5K LE
- Interfaz USB 2.0
- MIMO 1x1 (no expandible)

Tiene un bajo costo, útil para pruebas básicas de LTE, pero limitado en capacidad de ancho de banda y procesamiento.

2.6.2.5. USRP

Figura 22 USRP (Universal Software Radio Peripheral)

FUENTE: IMAGEN TOMADA DE ETTUS.COM

El USRP es una línea de SDR de alta gama desarrollada por Ettus Research.

Se utiliza en aplicaciones avanzadas de telecomunicaciones, incluyendo estaciones base LTE.

Posee un alto rendimiento, adecuado para estaciones base LTE comerciales, investigación y aplicaciones 5G[56].

A continuación, se presenta una tabla 4, observamos una comparación entre los diferentes equipos, y su compatibilidad con redes LTE.

Característica	BladeRF x40	BladeRF 2.0 micro xA4	BladeRF 2.0 micro xA5	BladeRF 2.0 micro xA9	USRP
Rango de frecuencia	300 MHz – 3.8 GHz	47 MHz – 6 GHz	47 MHz – 6 GHz	47 MHz – 6 GHz	DC – 6 GHz (según modelo)
Ancho de banda (IBW)	Hasta 28 MHz	Hasta 56 MHz	Hasta 56 MHz	Hasta 56 MHz	Hasta 160 MHz
Frecuencia de muestreo	Hasta 40 MSPS	Hasta 61.44 MSPS	Hasta 61.44 MSPS	Hasta 61.44 MSPS	Kintex-7 o Zynq
FPGA	Cyclone IV (40K LE)	Cyclone V (49K LE)	Cyclone V (77K LE)	Cyclone V (301K LE)	USB 3.0 / Gigabit Ethernet / PCIe
Interfaz	USB 3.0	USB 3.0	USB 3.0	USB 3.0	2x2 / 4x4
MIMO	1x1 (2x2 opcional)	2x2 nativo	2x2 nativo	2x2 nativo	srsRAN, OAI, GNU Radio
Compatibilidad	GNURadio, srsRAN, SoapySDR, MATLAB	GNURadio, srsRAN, SoapySDR, MATLAB	GNURadio, srsRAN, SoapySDR, MATLAB	GNURadio, srsRAN, SoapySDR, MATLAB	Alto ($1500 – $5000)
Precio estimado	$520	$540	$670	$860	Estaciones base LTE comerciales

Tabla 4 Comparativa entre Equipos SDR

2.7. Software de código abierto

Estos programas permiten a los investigadores y empresas personalizar y mejorar el software según sus necesidades, reduciendo costos y evitando depender de licencias propietarias.

Para este trabajo de titulación se tomó en cuenta 2 software potenciales dedicados específicamente en redes móviles, los cuales ofrecen una gran cantidad de librerías y soluciones que serán de gran utilidad, para cumplir los objetivos propuestos[57].

Los softwares propuestos son los siguientes:

- OAI (Open Air Interface)
- srsRAN
- Open5gs

2.7.1. Open Air Interface (OAI)

Es una interfaz de código abierto francesa desarrollada por la escuela de postgrado y centro de investigación EUROCOM en el departamento de Comunicaciones Móviles.

Gracias a esta herramienta es posible la implementación de una red LTE de Reléase 8.6, además de agregar ciertos aspectos de la versión 10[58].

Reléase es un sistema de numeración, que pertenece a los estándares que realiza el grupo 3GPP, La versión 8 se lanzó en 2008 y fue la que estandarizo las redes LTE.

OAI tiene una librería de código escrita en C para el despliegue de una red LTE, en donde la totalidad de las características necesarias ya se encuentran disponibles en su totalidad. Además de esto es un proyecto que actualmente se está orientando en aplicaciones 5G, ejemplos de esto son funciones como Machine to Machine (M2M) y Cloud RAN[51].

Esta interfaz es compatible con USRP y también puede ser ejecutada en Nuand BladeRF.

2.7.2. srsRAN

Es una interfaz de código abierto realizada por SRS, una empresa irlandesa, que dedico este proyecto al despliegue de una red LTE de Reléase 8.

srsRAN proporciona una librería en lenguaje C, además tiene un uso frecuente de Single Instruction Multiple Data (SIMD), el cual se encarga del envió de datos múltiples bajo una misma instrucción, incrementando su rendimiento.

Su interfaz es compatible con la USRP desarrollada por Ettus Ruffus Research, pero también puede operar con dispositivos de Nuand BladeRF.

Proporciona herramientas especializadas para el despliegue de LTE, sin embargo, es una plataforma aun en desarrollo, por lo que solo determinara puntos inestables[59].

2.7.3. Open5GS

Uno de los softwares más completos que podemos tener para simular LTE es Open5Gs se trata de un software de código abierto que sigue las normas de la versión 17, sirve tanto para desplegar 4G (LTE), como 5G (NR). Ya que este software solo servirá para netamente lo que es configurar el EPC se podría combinar con otro software para simular eNodeB, lo cual también lo hace una herramienta fácil de configurar [59].

2.7.4. OAI vs srsRAN vs Open5gs

OAI, srsRAN y Open5gs son interfaces parecidas que siguen el mismo propósito, la elección de una con la otra dependerá de la aplicación especifica que le queramos dar.

A continuación, se muestra en la Tabla 5 una comparación de ambas con sus características:

Característica	OpenAirInterface (OAI)	srsRAN	Open5GS
Enfoque Principal	Despliegue completo de redes LTE y 5G (eNB, EPC, UE)	Implementación flexible de eNB y UE en LTE	Núcleo de red EPC/5GC para LTE y 5G
Componentes Soportados	eNB, EPC, UE (red completa)	eNB y UE (no incluye EPC completo)	MME, HSS, SGW, PGW, AMF, SMF, UPF, AUSF, NRF, PCRF, etc.
Compatibilidad con Hardware SDR	USRP, BladeRF, LimeSDR, Xilinx RFSoC	USRP, BladeRF	No aplica directamente (funciona en conjunto con SDR)
Soporte para MIMO	Sí, hasta 2x2 y 4x4	Sí, pero más limitado (principalmente 2x2)	No aplica (es núcleo de red, no parte de radio)
Capacidad de Implementación	Red LTE/5G completa con interconexión a redes reales	LTE básico sin EPC, requiere integración externa	EPC/5GC completo para redes móviles
Flexibilidad y Personalización	Alta, con soporte 5G y múltiples opciones de configuración	Moderada, más orientado a pruebas LTE	Alta, arquitectura modular con API REST

Escalabilidad	Alta, ideal para despliegues grandes	Moderada, orientada a investigación	Alta, apto para redes privadas o comerciales
Requerimientos de Hardware	Elevados, necesita CPU potente, RAM y buen SDR	Moderados, se puede correr en laptops o mini PC	Moderados, funciona en servidores o PC estándar
Facilidad de Configuración	Complejo, requiere conocimiento avanzado	Sencillo, ideal para pruebas rápidas	Moderado, documentación clara y moderna
Casos de Uso	Investigación avanzada, redes privadas, 5G académico/industrial	Experimentación LTE, educación, entornos de prueba	Despliegue del núcleo de red LTE/5G
Comunidad y Soporte	Amplia, respaldada por academia y operadores	Activa pero más pequeña	En crecimiento, con buena documentación y GitHub activo
Licencia	OAI Public License (basada en Apache 2.0 con cláusulas extra)	Apache 2.0	Apache 2.0

Tabla 5 Comparativa entre Interfaces de Código abierto basadas en LTE

2.8. Sistemas Operativos

La implementación de una estación base LTE con OpenAirInterface (OAI) o srsRAN, requiere de un procesamiento computacional exigente, como ya mencionamos antes, la arquitectura SDR basa su funcionamiento sobre todo en software, por lo que esas funciones dependen mucho de la capacidad computacional.

Es fundamental la elección del sistema operativo para que nuestro entorno de simulación aproveche los recursos de manera eficiente y no cause errores.

2.8.1. Windows

Windows no es el sistema operativo ideal para el despliegue de redes LTE con SDR, ya que muchas herramientas de código abierto, como OAI, srsRAN, están diseñadas para funcionar en entornos Linux.

Windows, aunque muy fácil de manejar no es la mejor opción, debido a que los softwares de código abierto son basados en Linux y según su documentación funcionan mejor en Linux, por otro lado esta WSL, que podría ayudarnos, pero no sería ideal, ya que nos impediría usar las computadoras a su máximo rendimiento[60].

2.8.2. Ubuntu for WSL

WSL (Windows Subsystem for Linux) permite ejecutar una distribución Ubuntu dentro de Windows sin necesidad de una máquina virtual. Esto facilita el uso de herramientas de desarrollo para SDR en Windows.

Beneficios de Ubuntu for WSL para SDR:

- Permite ejecutar OpenAirInterface y srsRAN sin salir de Windows.
- Compatible con librerías y herramientas de Linux.
- Fácil instalación sin necesidad de particionar el disco duro.

Limitaciones:

- No soporta comunicación directa con hardware SDR sin configuraciones adicionales.
- Problemas de latencia en aplicaciones de tiempo real.
- Requiere una versión avanzada de Windows 10 o superior.

2.8.3. Linux

Linux es el sistema operativo más recomendado para la implementación de estaciones base LTE con OAI y srsRAN, ya que ofrece compatibilidad total con hardware SDR y optimización en la gestión de redes[61].

2.8.3.1. Ubuntu for Linux

Ubuntu es la distribución de Linux más utilizada para el despliegue de estaciones base LTE debido a su compatibilidad con OAI y srsRAN.

Beneficios de Ubuntu para SDR:

- Totalmente compatible con OAI y srsRAN.
- Gran comunidad de soporte.
- Facilidad de instalación y configuración.
- Soporte directo para hardware SDR sin necesidad de drivers adicionales.

2.8.4. Comparativa Windows vs Linux

A continuación, veremos una comparación mostrada en la tabla 6

Característica	Windows	Linux (Ubuntu)
Compatibilidad con OAI	No recomendado	Totalmente compatible
Compatibilidad con srsRAN	Parcial con WSL	Compatible y optimizado
Soporte para Hardware SDR	Limitado	Nativo para BladeRF, USRP, LimeSDR
Rendimiento en redes LTE	Bajo	Alto
Facilidad de instalación	Fácil (pero con limitaciones)	Requiere configuración inicial
Latencia en el procesamiento de señales	Alta	Baja (mejor rendimiento en tiempo real)
Flexibilidad y personalización	Limitada	Muy alta
Uso recomendado	Desarrollo básico y pruebas	Despliegue de estaciones base LTE

Tabla 6 Comparativa entre Windows y Linux

2.9.Marco Teórico

Con el desarrollo de la Radio Definida por Software (SDR) en el Ecuador, se han llevado a cabo investigaciones que impulsen la evolución de las redes móviles LTE, ya que el uso de esta tecnología permite configurar y adaptar sistemas de telecomunicaciones de manera flexible, optimizando la implementación de estaciones base y mejorando la eficiencia del espectro radioeléctrico. Se han desarrollado soluciones basadas en SDR para el despliegue de redes móviles en zonas rurales y de difícil acceso, facilitando la conectividad en comunidades remotas, el soporte a infraestructuras de emergencia y la creación de entornos de prueba para la investigación en 5G y futuras generaciones de redes celulares.

Capítulo III

3.1. Desarrollo de la Propuesta

Aquí en este capítulo se expone los pasos que se tomaron en cuenta para realizar el proyecto propuesto en este documento.

Como primer punto tenemos la sección 3.1, donde se detallan la elección de componentes necesarios para este proyecto.

A continuación, en la sección 3.1 y 3.2 se nos presenta la arquitectura y diseño que tomara nuestra red LTE. En la sección 3.4 de Implementación se nos explicara paso a paso la instalación del software que se utilizó para realizar el siguiente proyecto.

En la sección 3.5 veremos el costo de los equipos para el proyecto.

3.2. Componentes de la Propuesta

3.2.1. Componentes Físicos

Aquí presentamos los componentes de hardware que se necesitaron para la realización de este proyecto de titulación.

3.2.1.1. PC de Escritorio

Nuestra red LTE será simulada en su mayor parte, siendo la enoje el único componente que se simulará de una manera un poco más real gracias a la ayuda de un SDR.

Por esta razón el poder de procesamiento que necesitaremos será muy grande, por ello se deicidio dividir esta simulación en dos computadores, aunque esta se podría realizar en una sola computadora, no contamos con una computadora con esa capacidad de procesamiento.

En este caso se nos proporcionó por parte de la facultad de Sistemas y Telecomunicaciones, dos computadores con un nivel de procesamiento medio, lo cual será suficiente para nuestra simulación de red.

Además, el uso de dos computadoras podría ser beneficioso para simular un ambiente más real, ya que como sabemos las interfaces de EPC y eNodeB están físicamente separadas en una red LTE real.

Por otro lado, la simulación con dos máquinas físicas conectadas a una red IP, hace que la simulación de la interfaz S1 sea más realista.

- **HP Prodesk (EPC)**

Figura 23 HP-Prodesk 400 G4 Desktop mini

Este computador proporcionado por la facultad de Sistemas y telecomunicaciones es un equipo portable y productivo, con un procesador Intel i5, con 8 GB de RAM.

Característica	Detalle
Nombre del dispositivo	HP Prodesk 400 G4 Desktop Mini
Memoria RAM	8 GB (7.6 GiB utilizable)
Procesador	Intel® Core™ i5-8500T CPU @ 2.10GHz × 6 núcleos
Gráficos	Intel® UHD Graphics 630
Capacidad del disco	500 GB HDD o SSD (según configuración)
Sistema Operativo	Windows 10 Pro (64 bits)
Tipo de SO	64 bits
Versión de Windows	Windows 10 Pro, versión empresarial
Sistema de ventanas	Windows Desktop Manager
Actualizaciones de software	Windows Update automáticas

Tabla 7 Características de HP Prodesk

El EPC contine componentes como MME, HSS, SGW, PGW que pueden beneficiarse al trabajar en una sola máquina,

Se eligió este computador como EPC, ya que los procesadores Intel presentan mayor estabilidad en tareas de este tipo.

- **ASRock Ryzen 7 (eNodeB)**

Este computador proporcionado por la facultad de Sistemas y telecomunicaciones es un equipo portable y productivo, con un procesador AMD Ryzen 7, con 8 GB de RAM.

Característica	Detalle
Marca	ASRock
Nombre del dispositivo	facsistel-desktop
Memoria RAM	8 GB (7.8 GiB utilizable)
Procesador	AMD® Ryzen 7 2700 (8 núcleos / 16 hilos) @ 3.2 GHz
Gráficos	Radeon Vega / Radeon RX (depende de la GPU instalada)
Capacidad del disco	1 TB HDD o SSD
Sistema Operativo	Windows 10 Pro (64 bits)
Tipo de SO	64 bits
Versión de Windows	Windows 10 Pro, versión empresarial
Sistema de ventanas	Windows Desktop Manager

Tabla 8 Parámetros de Pc ASRock

Se eligió este equipo por su distribución en los recursos, el eNodeB al estar conectado con SDR, necesita procesar intensivamente la señal en tiempo real. La arquitectura de la Ryzen 7 con más núcleos que la Intel y un mejor rendimiento en procesamiento paralelo es ventajosa para estas tareas.

3.2.1.2.BladeRF micro 2.0 XA4

Figura 24 BladeRF micro 2.0 xA4

FUENTE: IMAGEN TOMADA DE NUAND.COM

El modelo xA4 es el equipo que ofrece las características necesarias para simular nuestra red LTE, además de ser un modelo portátil, su rango de precios está en la media a comparación con los demás, lo que lo hace más accesible, por otro lado, los equipos BladeRF son compatibles con las interfaces Open5GS, OpenAirInterface y srsRAN.

La tabla 9 resume las principales características del BladeRF micro 2.0 xA4.

Parámetro	Valor
Rango de frecuencias	47 MHz – 6 GHz
Ancho de banda (BW) Analógico	56 MHz
Resolución ADC/DAC	12 bits
Muestras ADC/DAC	Hasta 61.44 Mbps
Precisión de frecuencia	± 1 ppm (con reloj interno)
Potencia de salida	Aproximadamente 8 dBm máximo
Figura de ruido del receptor	< 8 dB (típico, depende de la frecuencia)
Modelo FPGA	Intel Cyclone V (301K LE, 5CEBA2F23C8N)
Conectividad con host	Puerto USB 3.0
Alimentación	Puerto USB 3.0 o Fuente de alimentación externa de 6V
Otros	Half of Full Duplex, MIMO 2x2
Usos	Redes celulares LTE, 5G NR experimentales, WiFi, SDR general

Tabla 9 Características del BladeRF xA4

3.2.1.3. Antenas Tri Banda

Figura 25 Antena Tri -banda

FUENTE: IMAGEN TOMADA DE NUAND.COM

La antena tribanda 700–2600 MHz es muy buena opción para el proyecto de la red LTE, ya que trabaja en las bandas que se utilizan en comunicaciones móviles.

Tiene un diseño flexible con un conector SMA macho que facilita su implementación en diferentes configuraciones, además de ser compatible con diferente equipo, con su ganancia de 5 dBi y su bajo VSWR (Voltage Standing Wave Ratio), nos ofrece buena transmisión y recepción.

Parámetro	Valor
Tipo	Antena de goma duradera
Interfaz	Conector macho SMA
Rango de frecuencia	700 MHz – 2600 MHz
Ganancia	5 dBi
VSWR	No mayor a 1.5
Longitud	11 pulgadas
Montaje	Flexible (rango de movimiento de 100 grados)
Compatibilidad	BladeRF 1.0, BladeRF 2.0, BT-100, BT-200
Usos recomendados	Investigación y experimentación en bandas ISM

Tabla 10 Parámetros de Antena Tri Banda

3.2.1.4. Cable Multipar Cat 5

Se utilizará cable multipar categoría 5 (Cat 5) para las conexiones de red necesarias entre los distintos componentes de la arquitectura LTE experimental, como el eNodeB, el EPC y otros dispositivos de apoyo. Este cable soporta transmisiones de datos a velocidades de hasta 100 Mbps (Fast Ethernet) y en algunos casos hasta 1 Gbps, proporcionando una infraestructura de comunicación física confiable y de bajo costo para el proyecto.

Figura 26 Cable Multipar

FUENTE: IMAGEN TOMADA DE AMAZON

Se uso el cable multipar categoría 5, disponibles en el laboratorio de telecomunicaciones para conectar el computador ASROCK (Ryzen 7), a la red LAN debido que el mismo no cuenta con tarjeta WIFI disponible, y necesitábamos asignarle una IP estática por lo que tiene que estar obligatoriamente conecta a nuestra red LAN.

3.2.1.5. NARDA SRM 3006

Figura 27 Equipo Narda

FUENTE: IMAGEN TOMADA DE NARDA-STS

Utilizaremos el equipo Narda disponible en la facultad de sistemas y telecomunicaciones, para realizar mediciones de radio en frecuencias LTE, que van a hacer generadas por el equipo BladeRF XA4, al simular la estación base LTE

3.2.1.6. Router inalámbrico N 300Mbps WR840N

Figura 28 Router N 300Mbps WR840N

FUENTE: IMAGEN TOMADA DE TP-LINK

Se utilizo un router TP-Link N 300Mbps WR840N para realizar una red local donde se conectarán las dos computadoras del laboratorio de telecomunicaciones, para comunicarse entre sí, elegimos

crear una LAN debido a que, si nos conectábamos a la red principal de la UPSE, no íbamos a saber qué dirección IP se nos iba a asignar automáticamente, es por eso que al crear una red LAN tuvimos la facilidad de asignar la IPs de nuestra preferencia a las computadoras correspondientes.

Parámetro	**Valor**
Producto	Router inalámbrico N 300Mbps WR840N
Formatos	IEEE 802.11b/g/n, con soporte para 2.4 GHz
Compatibilidad	Compatible con dispositivos Wi-Fi 802.11b/g/n
Características	Velocidad inalámbrica de hasta 300 Mbps, 2 antenas externas fijas de 5 dBi, 4 puertos LAN 10/100 Mbps, 1 puerto WAN 10/100 Mbps
Aplicación	Redes domésticas, pequeñas oficinas, conexión de múltiples dispositivos Wi-fi.

Tabla 11 Parámetros de TP-Link

3.2.2. Componentes Lógicos

3.2.2.1. Ubuntu 20.04 LTS

El sistema operativo Ubuntu 20.04 LTS será la plataforma base sobre la cual se desplegarán las aplicaciones necesarias para la configuración del eNodeB y el EPC. Ubuntu es una distribución Linux de amplio soporte en la comunidad de telecomunicaciones y es compatible con la mayoría de las herramientas de código abierto utilizadas en redes LTE experimentales

- **Instalación del sistema Operativo Ubuntu 20.04 LTS**

Como las computadoras ya cuentan con Windows 10 como sistema operativo, tendremos que instalar el Ubuntu 20.04. LTS, con ayuda de USB bo oteable.

Paso 1:

Como no queremos eliminar Windows, ya que los equipos son prestados, procederemos a hacer un dual boot, para poder tener 2 sistemas operativos en un mismo computador.

Primero creamos una partición donde instalaremos el sistema de Linux Ubuntu, para eso abrimos el administrador de discos en Windows, ya sea por la caja de comando CMD con el comando:

```
diskmgmt.msc
```

O dándole clic derecho en el icono de Windows y administrador de archivos.

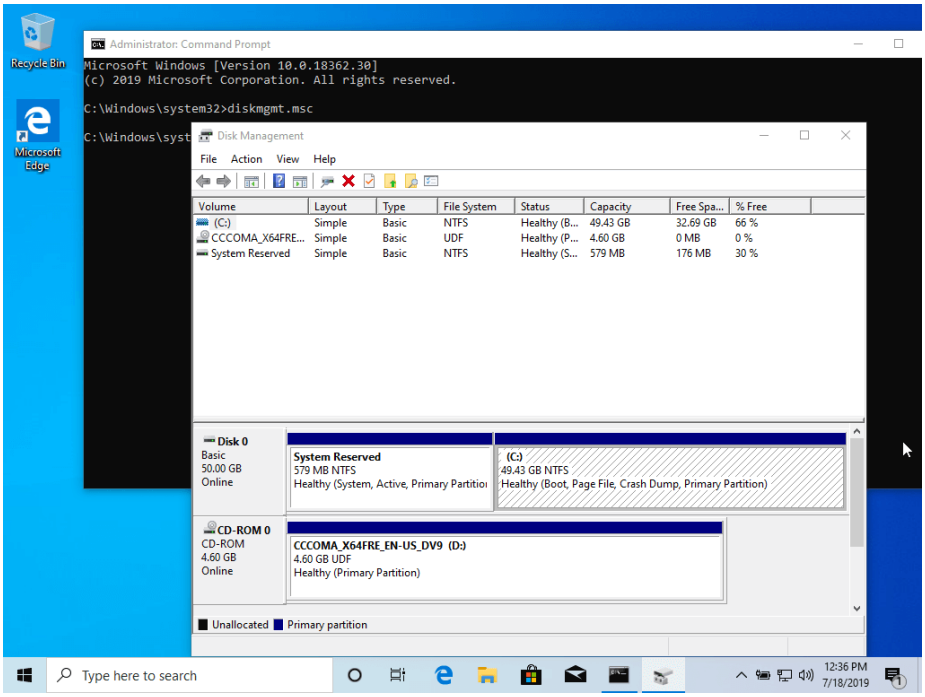


Figura 29 Particiones de Disco

Una vez en el administrador de discos procederemos a darle clic derecho en el disco C: de Windows y a reducir volumen, se nos abrirá una ventana donde seleccionaremos la cantidad a reducir, la cual debe ser mínimo de 20000 MB, en esta ocasión le daremos 100000 MB para nuestra instalación.

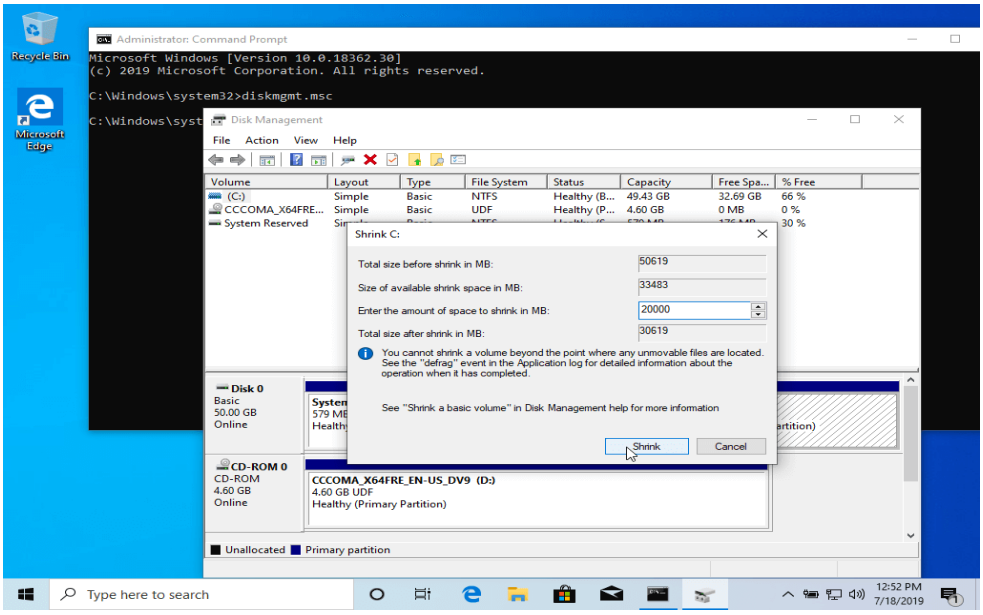


Figura 30 Reducción de Volumen

Una vez creada la partición veremos algo como esto:

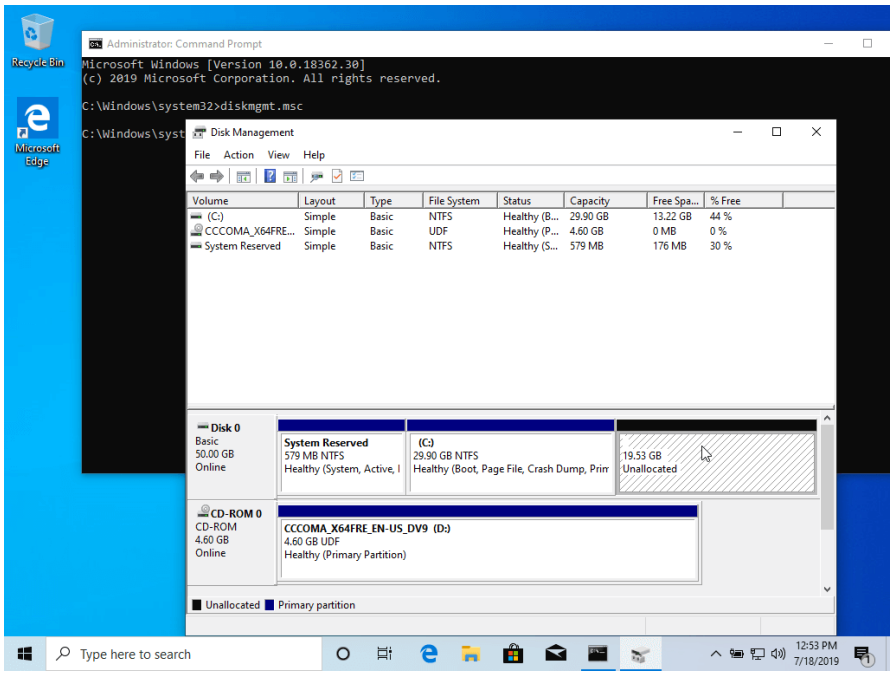


Figura 31 Partición para Linux Ubuntu

Parte 2:

Para esta parte deberíamos tener una USB boteada con Ubuntu 20.04 LTS, o en caso de que no, botear

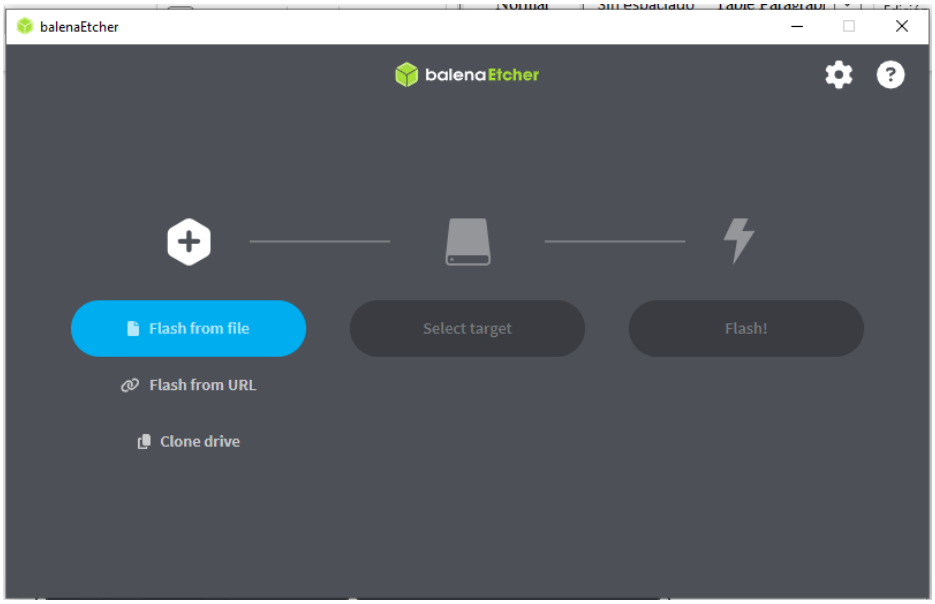


Figura 32 Ballena Etcher

A continuación, reiniciaremos el computador y justo cuando vuelva a encender, rápidamente presionaremos la tecla F9, lo cual nos abrirá el menú de arranque del computador, seleccionaremos el USB boteado en la opción UEFI.

Una vez hecho esto nos aparecerá lo siguiente, elegimos el idioma que hablamos y continuamos con la instalación de Linux Ubuntu

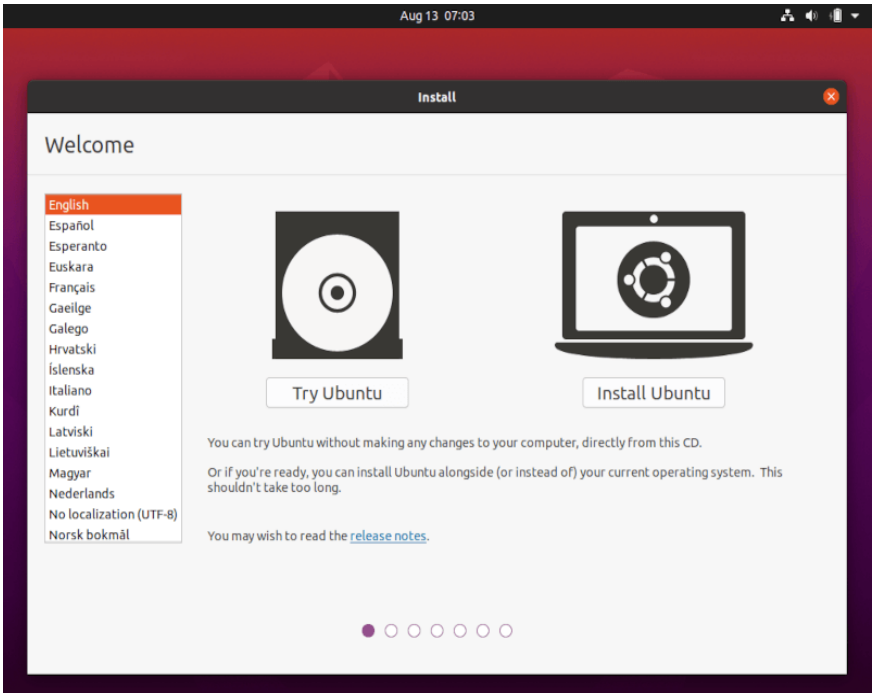


Figura 33 Instalación de Linux

A continuación, colocamos el idioma de nuestro teclado

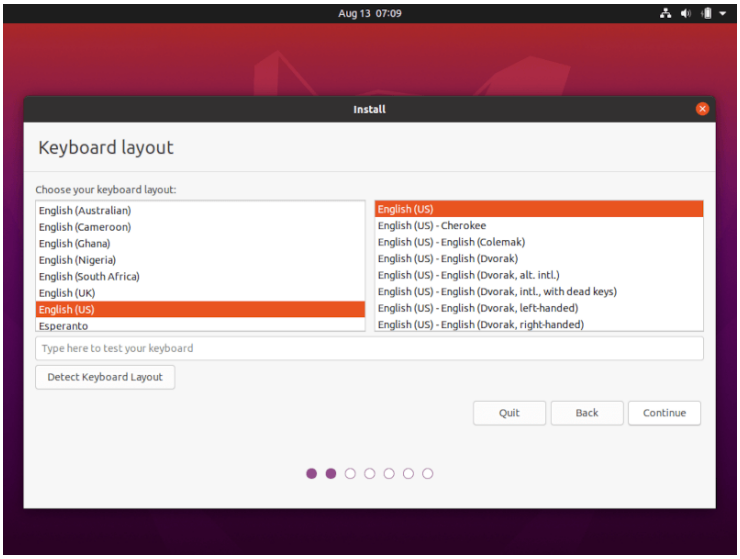


Figura 34 Idioma del teclado

Para continuar la instalación de nuestra distribución, colocamos la opción normal

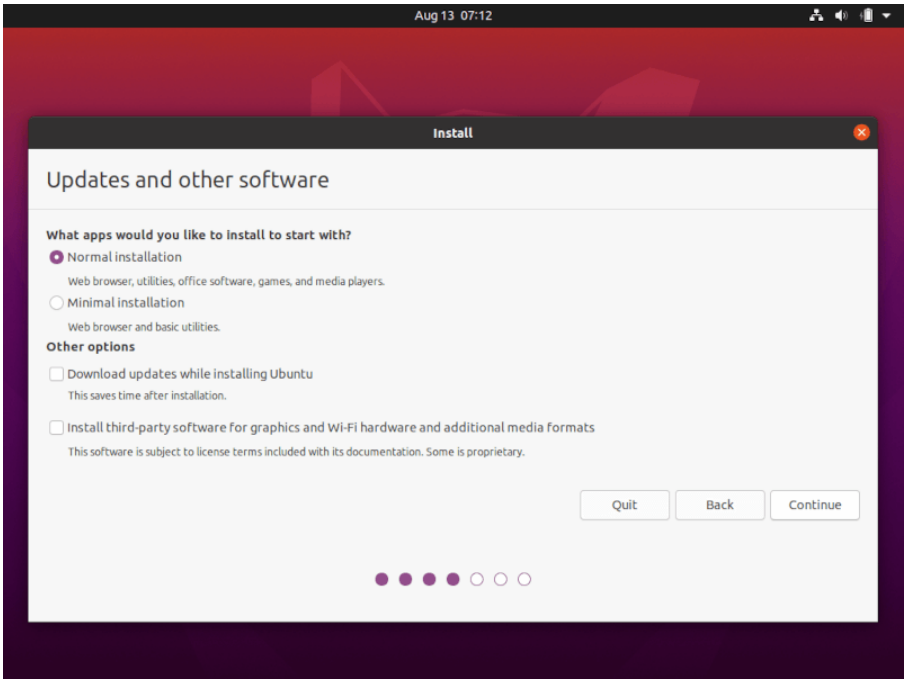


Figura 35 Instalación Normal

En el tipo de instalación elegimos la última opción para realizar una instalación personalizada, aquí podremos crear las particiones que queramos de acuerdo con nuestras necesidades

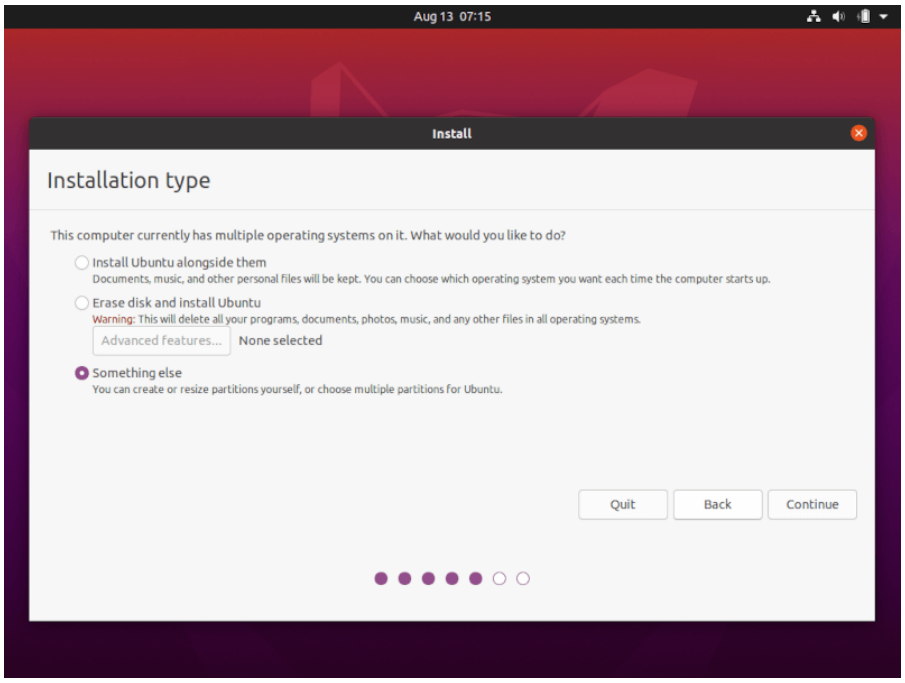


Figura 36 Tipo de Instalación

En este apartado podemos realizar las particiones correspondientes necesarias para el sistema y otras opcionales, según nuestras preferencias

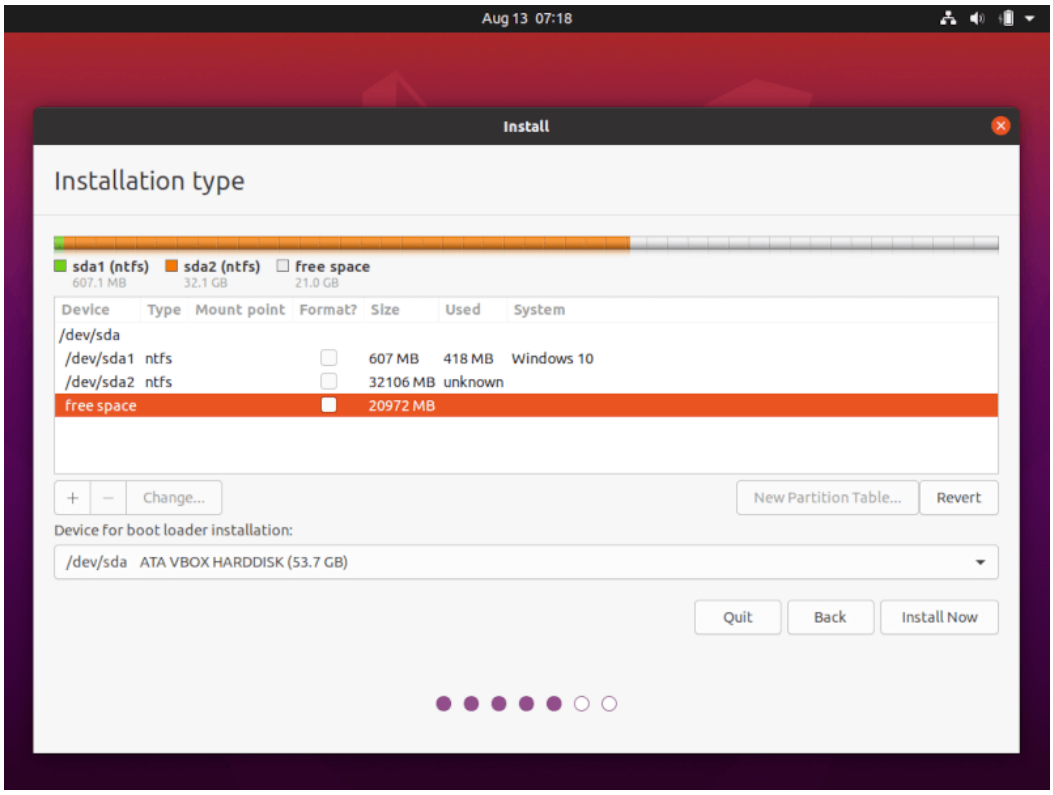


Figura 37 Particiones de Linux

A continuación, elegimos nuestra ubicación, para obtener las actualizaciones según nuestra región.

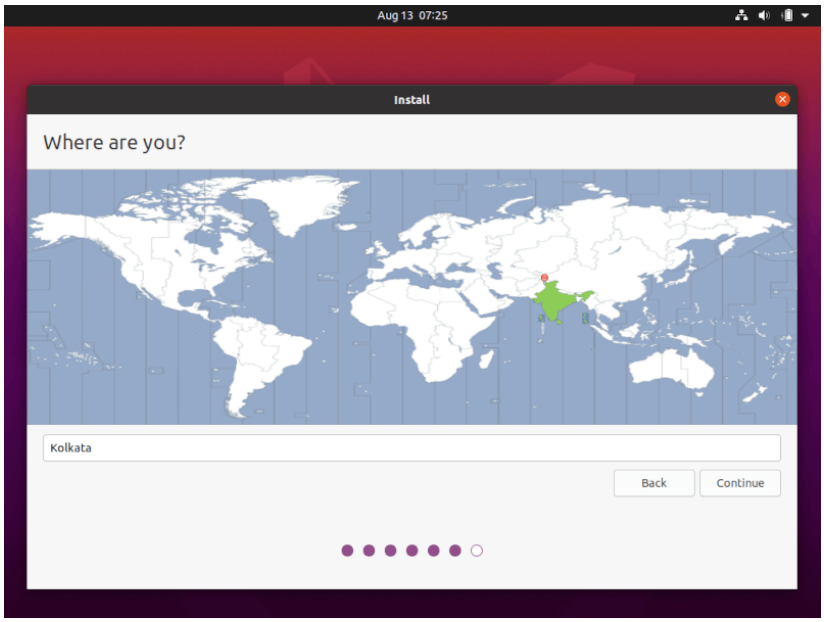


Figura 38 Región o ubicación

Para finalizar registramos un usuario y una contraseña para poder iniciar sesión.

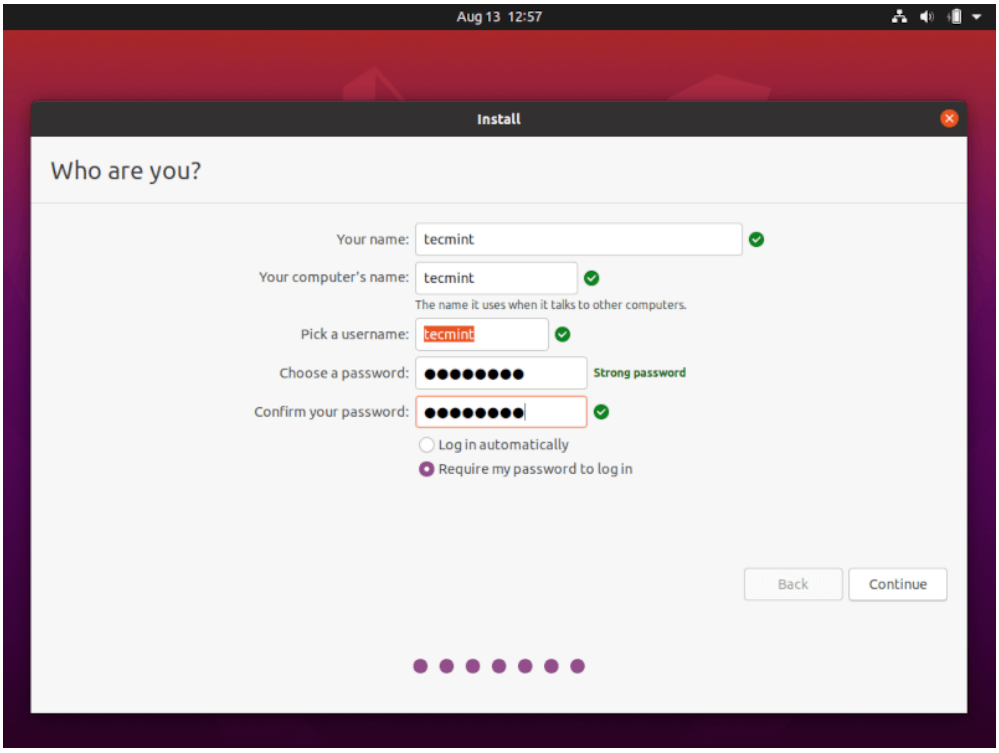


Figura 39 Usuario y contraseña

Una vez terminado la configuración, se procederá a instalar el sistema operativo tal y como se muestra a continuación.

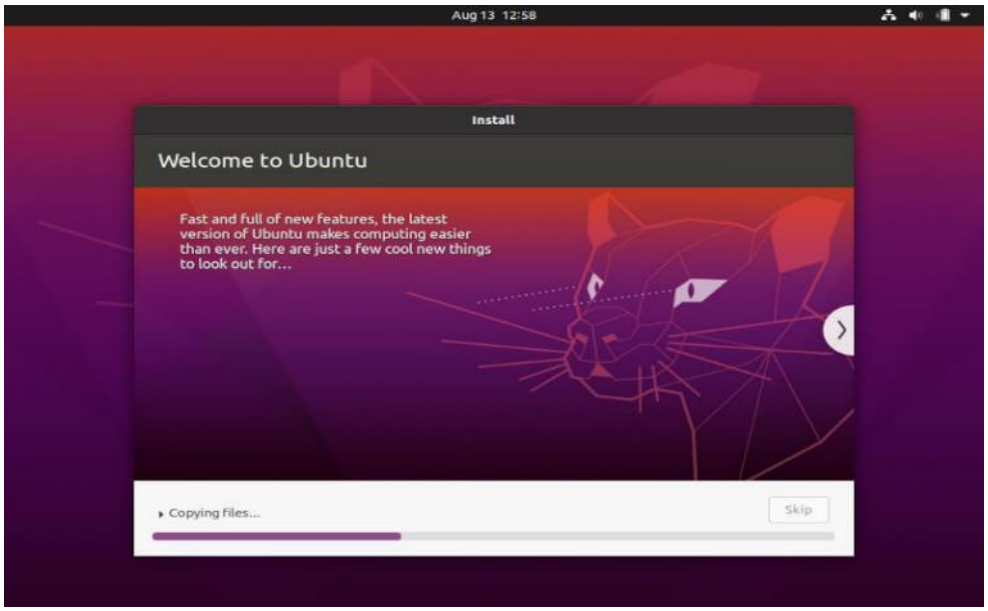


Figura 40 Instalación de Ubuntu

Una vez finalizado nos mandara este mensaje.

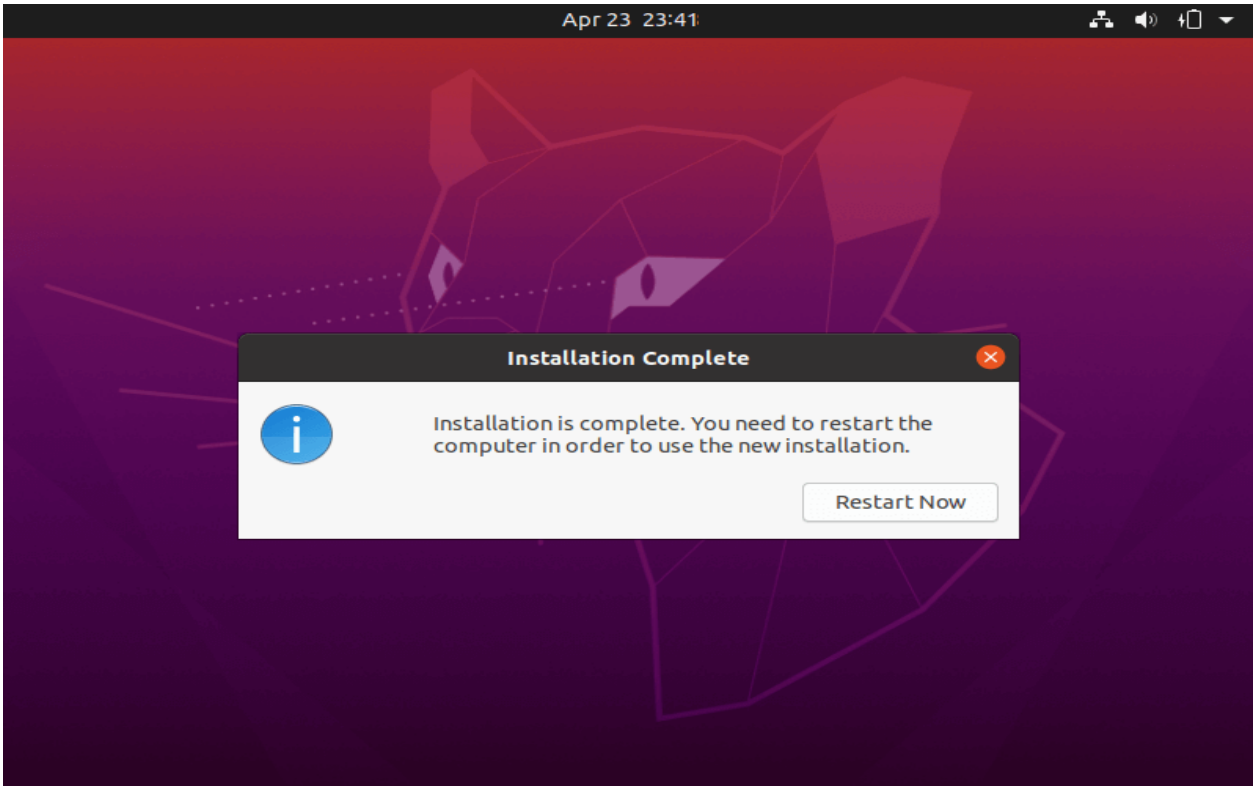


Figura 41 Finalización de Instalación

3.2.2.2.Open5GS

Figura 42 Open5gs

OPEN5GS será el software principal para implementar tanto el núcleo EPC de la red LTE experimental. Este proyecto de código abierto permite la creación de redes móviles reales con equipos de bajo costo, permitiendo pruebas de transmisión, autenticación, gestión de movilidad y calidad de servicio.

El repositorio que se utilizo es el Open5GS, que, aunque es un proyecto que ya está siendo dedicado para redes móviles 5G, aun cuenta con soporte y características propias de LTE, además al ser un repositorio relativamente nuevo, es compatible con la versión de Ubuntu que utilizaremos.

3.2.2.3.Wireshark

Figura 43 WireShark

Wireshark es una herramienta de análisis de protocolos de red que se utilizará para capturar, visualizar y analizar el tráfico de datos en la red LTE. Su empleo permitirá verificar la correcta transmisión de paquetes, identificar posibles problemas de configuración y asegurar la integridad de las comunicaciones entre los distintos elementos de la red.

3.2.2.4. GRSIMWrite

Es un software que ayudará a configurar los parámetros de las tarjetas SIM LTE programables entre ellos se podrá configurar la PLMN, códigos IMSI que servirán para la autenticación.

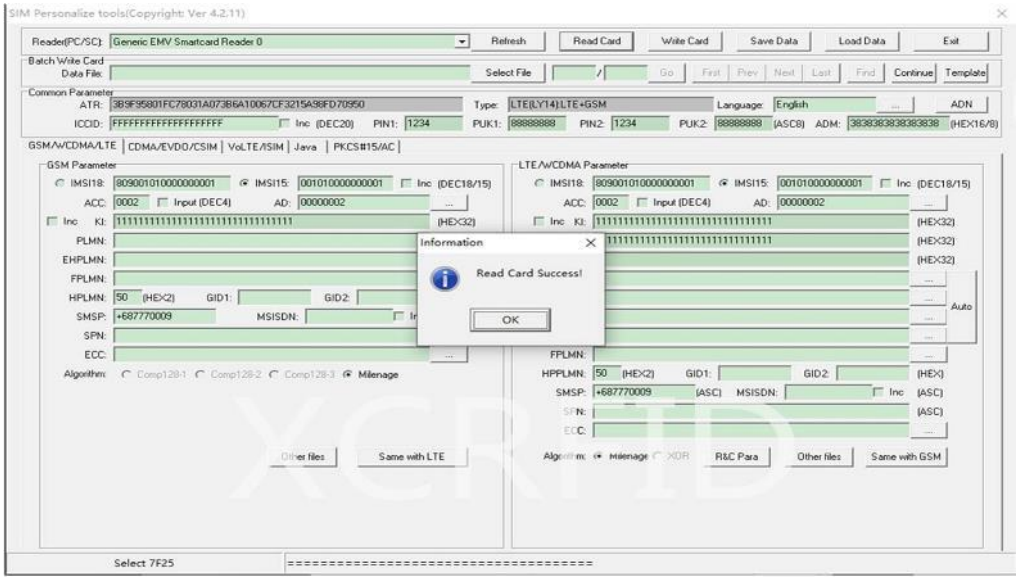


Figura 44 Software GRSIMWrite

3.3. Arquitectura de la red LTE

La arquitectura de la red LTE se conforma de EPC + eNodeB + UE, siguiendo esa estructura se diseñó la red siguiendo los estándares de la 3GPP. Las entidades que se configuraran son la MME, SGW, PGW, HSS y PCRF. Con un equipo SDR simularemos la estación base, y se dará acceso a los equipos de usuario que se conectaran a la red y se verificaran las credenciales mediante el HSS y MME.

3.3.1. Arquitectura Utilizada

La arquitectura de red se basó en Open5GS para conforman el EPC, mientras que eNodeB estará a cargo del software srsRAN, y se conforma como Open5GS EPC + srsRAN eNodeB + UE. En un computador crearemos el EPC, y en otro equipo el eNodeB, donde estará operando la estación base utilizando la tarjeta BladeRF micro.2.0 XA4, y los equipos de usuario serán teléfonos inteligentes que se usarán para conectarse a la red, de esta forma se creara la arquitectura necesaria.

Se ha seleccionado dos computadores del laboratorio de telecomunicaciones para realizar la implementación de la arquitectura, sería posible crearlo en un solo ordenador, pero si fuera un equipo con especificaciones técnicas muy elevadas, en este caso disponemos de una Intel i5 y 8 GB de Memoria RAM, y el otro ordenador es una Ryzen 7 y 8 GB de memoria RAM y se les instalo el sistema operativo Linux en la versión Ubuntu 20.04.6 LTS.

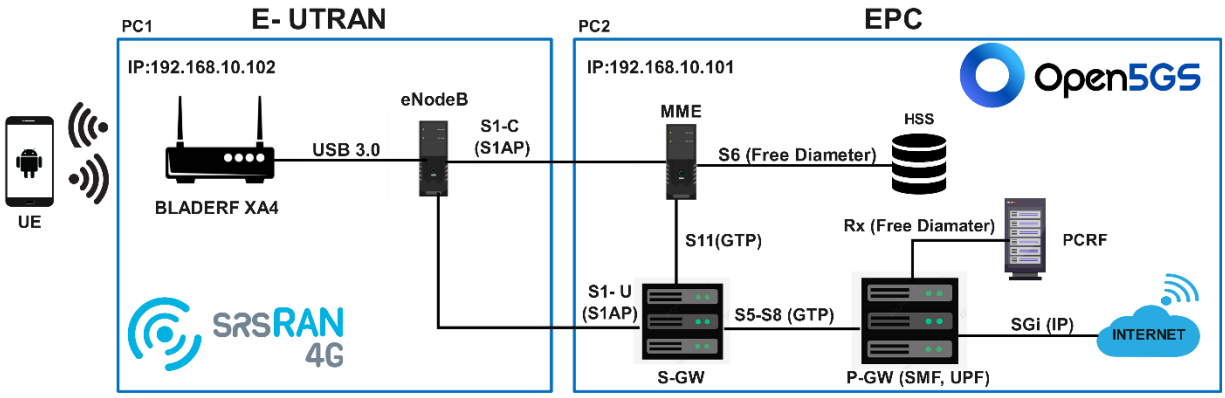


Figura 45 Esquema de la red Implementada

En la figura 45, se usa la norma TS 23.401 se puede observar el esquema de la red que se ha implementado en el laboratorio de Telecomunicaciones. La PC1 que es una Ryzen 7 será el encargará de simular la estación base por medio del software SrsRAN4G estará conectada a una tarjeta BladeRF micro2.0 XA4 el cual será full-dúplex o sea esto quiere decir que transmite y recibe datos de manera

simultánea mediante una interfaz USB 3.0, si se implementa otra clase de USB la conexión sería muy inestable.

La interfaz encargada de realizar esta comunicación es la interfaz S1 que conecta el eNodeB a la red principal EPC con las entidades, la MME por su parte es la encargada de realizar la autenticación de las claves enviadas por los equipos de usuarios. Se tendrá una base de datos con las claves respectivas que servirá como almacenamiento de los usuarios.

La entidad PGW está conectada a internet mediante una interfaz SGi ya que esta es la encargada de conectar a esta entidad con las redes externas, ya que el funcionamiento de esta entidad es actuar como puerta de enlace entre la red LTE interna y las redes externas.

La entidad SGW es el punto de interconexión de la red de acceso por radio (E-UTRAN) y el núcleo de la red (EPC). Esta entidad es la responsable de dirigir el tráfico de datos entre las redes externas como Internet y entre los equipos de usuario (UE).

Una de las entidades más importantes de la arquitectura LTE, es la HSS dicho de otro modo es la base de datos de los usuarios, en esta entidad se podrá encontrar la información de cada usuario, como claves IMSI, velocidad máxima de subida. Trabaja juntamente con la MME ya que le permite calcular las claves de autenticación de manera correcta.

La arquitectura estará configurada a un Router y crearemos una red local con IPS estáticas para realizar su comunicación la PC1 con el nombre de Facsistel estará trabajando con la IP: 192.168.10.102, y la otra computadora llamada Facsistel 2 estará operando con la IP estática: 102.168.10.101, cuando se tenga comunicación entre ambas direcciones IP, la red estará configurada para empezar con la implementación de la red.

3.4. Implementación

3.4.1. Configuración de la red

Una vez que se instale el sistema operativo de Linux en la versión Ubuntu 20.04.6 LTS, se va a configurar de la red para garantizar la comunicación entre ambas computadoras, se creó una red local llamada Tesis_4G y en los puertos Ethernet configuraremos del mismo roster las IPS estáticas.

Lista de Clientes de DHCP

Esta página muestra la información de todos los clientes de DHCP en la red.

ID	Nombre del Cliente	Dirección MAC	IP Asignado	Tiempo de Arrendamiento
1	facsistel2-HP-ProDesk-400-G4-DM	84:A9:3E:13:CC:61	192.168.10.101	Permanente
2	facsistel-desktop	70:85:C2:CB:7E:53	192.168.10.102	Permanente

Figura 46 Configuración de las IPS estáticas

Como se puede apreciar en la figura 46, se conectó las computadoras mediante la interfaz del Router TPLink se les asigno IPS estáticas, y se estableció la conexión entre las dos computadoras, para ello se realizó un ping para comprobar que exista comunicación, mediante el comando

Ping: 192.168.10.102

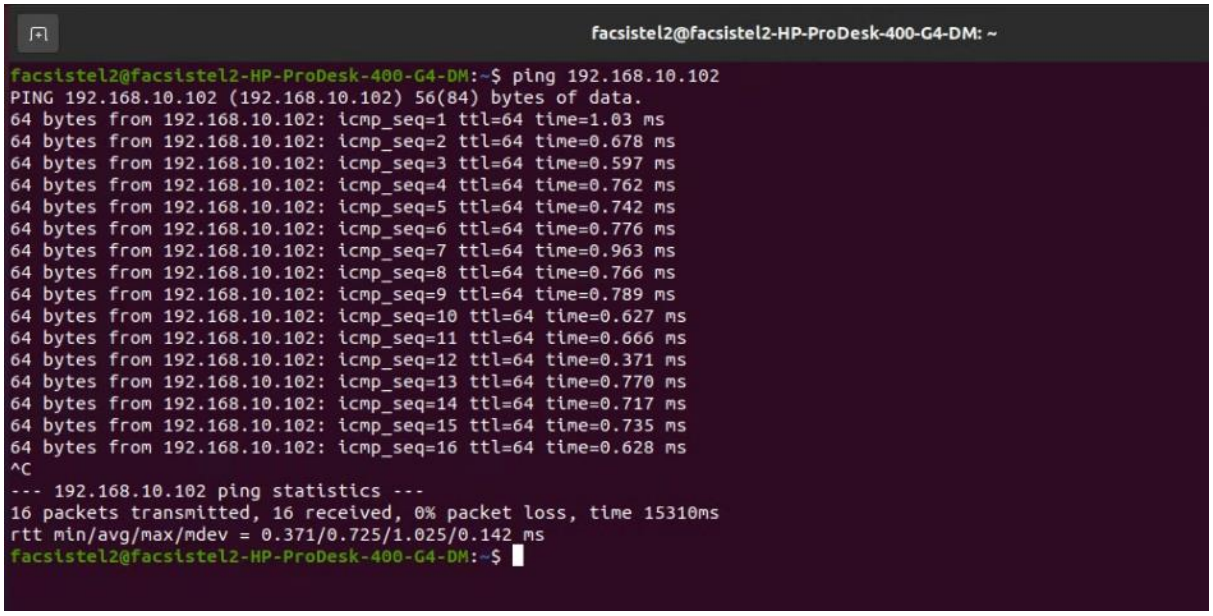


Figura 47 Comunicación entre ambos dispositivos

Como se puede observar en la figura 47, se estableció la comunicación entre los dos computadores del laboratorio de Telecomunicaciones, ya que nos da respuesta y todos los paquetes que enviamos fueron correctamente recibidos.

3.4.2. Preparación del Sistema

Una vez instalado Ubuntu 20.04.6 LTS y las maquinas conectadas en la misma red mediante cables Ethernet CAT 5, tenemos que actualizar las librerías, a veces nos darán errores, al momento de intentar actualizar las librerías nos dio un error, y tuvimos que buscar una variante de los repositorios que teníamos en este caso buscamos en la web y pudimos actualizar las librerías mediante:

```
deb https://archive.ubuntu.com/ubuntu focal main restricted universe multiverse
deb https://archive.ubuntu.com/ubuntu focal-updates main restricted universe multiverse
deb https://archive.ubuntu.com/ubuntu focal-security main restricted universe multiverse
deb https://archive.ubuntu.com/ubuntu focal-backports main restricted universe multiverse
```

Para colocar estas líneas en el script que nos ayudara a actualizar las librerías necesarias para el correcto funcionamiento, se usara el siguiente comando:

```
sudo nano /etc/apt/sources.list
```

Figura 48 Repositorios reemplazados con las nuevas líneas.

Ahora se podrán actualizar las librerías mediante el comando:

```
sudo apt update
sudo apt upgrade -y
```

En la máquina del eNodeB en este caso la computadora Ryzen 7, se instalará un Kernel de baja frecuencia, también llamado núcleo del sistema operativo es el encargado de actuar como puente entre el hardware y las aplicaciones que se ejecutan en ella.

Se instalará un Kernel de baja latencia para reducir el procesamiento en tiempo real, menor jitter, y mejorar el rendimiento, para realizar la instalación de este Kernel primero usaremos los siguientes comandos:

```
dpkg --list | grep linux-lowlatency
uname -r
```

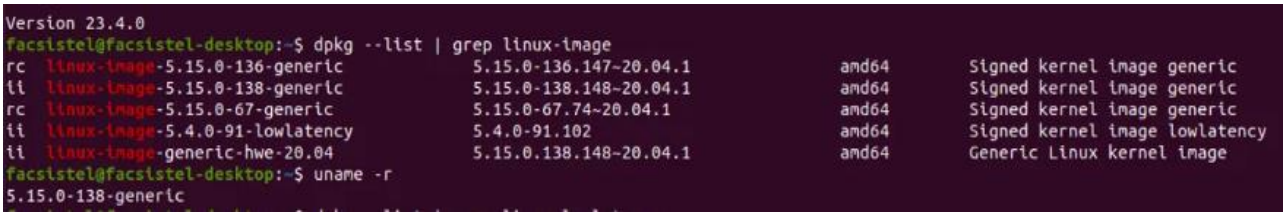


Figura 49 Lista de Kernel activos

Como se puede observar en la figura 49, el primer comando es para ver los Kernel activos y nos dará una lista actualizada, se puede observar que ya se había realizado la instalación de un Kernel de baja frecuencia, pero este no nos servirá ya que será muy desactualizado para el equipo SDR. Con el comando uname -r se utiliza para ver con que Kernel está trabajando el computador actualmente.

Con el siguiente comando se instalará un Kernel más reciente compatible con la versión de Ubuntu 20.04.6 LTS:

```
sudo apt install linux-lowlatency-hwe-20.04 -y
```

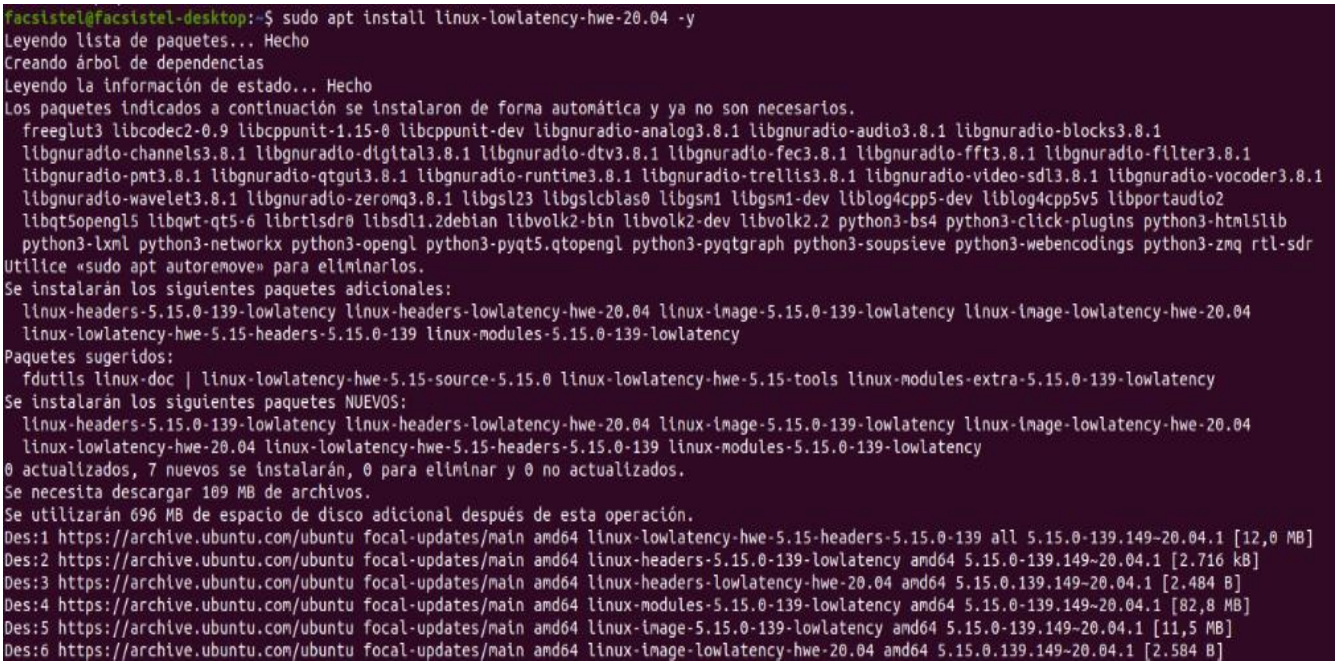


Figura 50 Instalación de Kernel de Baja Frecuencia

En la Figura 50, se está instalando el Kernel de baja frecuencia en este caso es la versión lowlatency-hwe 20.04, una vez que se completó la instalación podemos verificarlo con el comando para ver las lista de los Kernel.

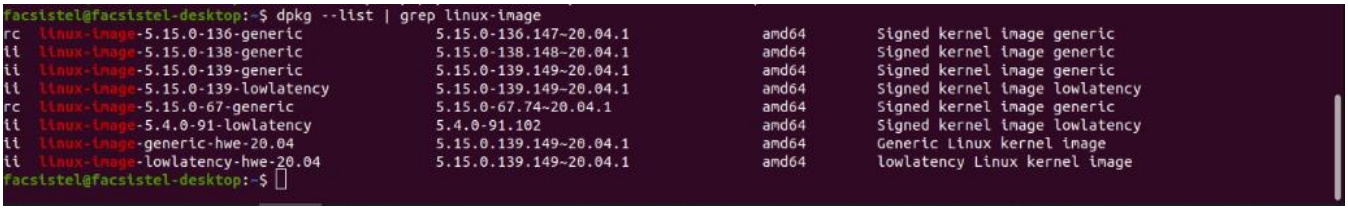


Figura 51 Verificación de Instalación

En la Figura 51, se puede observar que se instaló correctamente el Kernel de baja latencia lowlatency-hwe 20.04, para activar el Kernel y que se inicie correctamente en el sistema se utilizara el siguiente comando:

```
sudo reboot
```

El cual reiniciará el sistema, cuando el sistema se halla reiniciado, aparecerá una ventalla similar a esta:

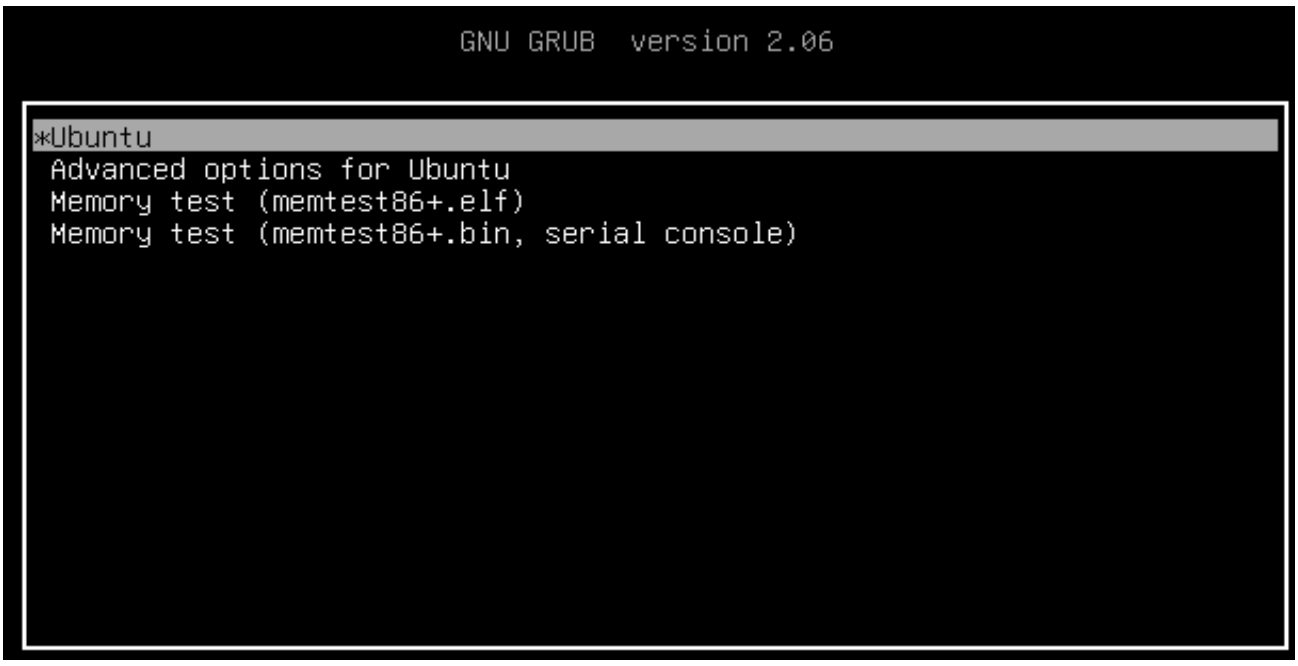


Figura 52 Pantalla de Inicio de Ubuntu

En la figura 52, realizaremos los siguientes pasos cuando arranque el sistema de Ubuntu 20.04:

1. Primero se dará a la tecla shift para acceder al menú Grub

2. Se seleccionará la opción Advanced optión for Ubuntu
3. Y se elegirá el Kernel 5.15.0-139-lowlatency

Luego de reiniciar el sistema mediante el comando uname -r, deberá aparecer el Kernel que se estará usando:

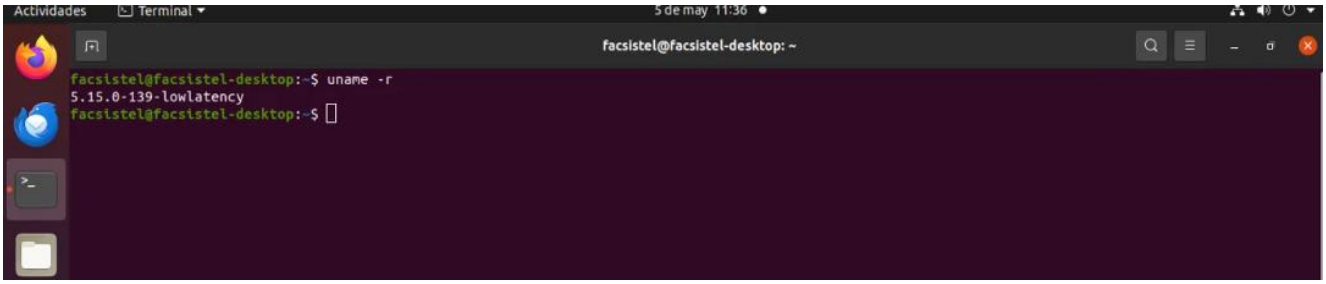


Figura 53 Kernel de baja latencia activo

Como segundo paso para optimizar el sistema se tiene que desactivar el escalonado por frecuencia para obtener el máximo rendimiento al momento de ejecutar la red LTE, por ello se desactivara esta función primero se instalara cpufrequtils, mediante la siguiente línea de comando:

```
sudo apt-get install cpufrequtils -y
```

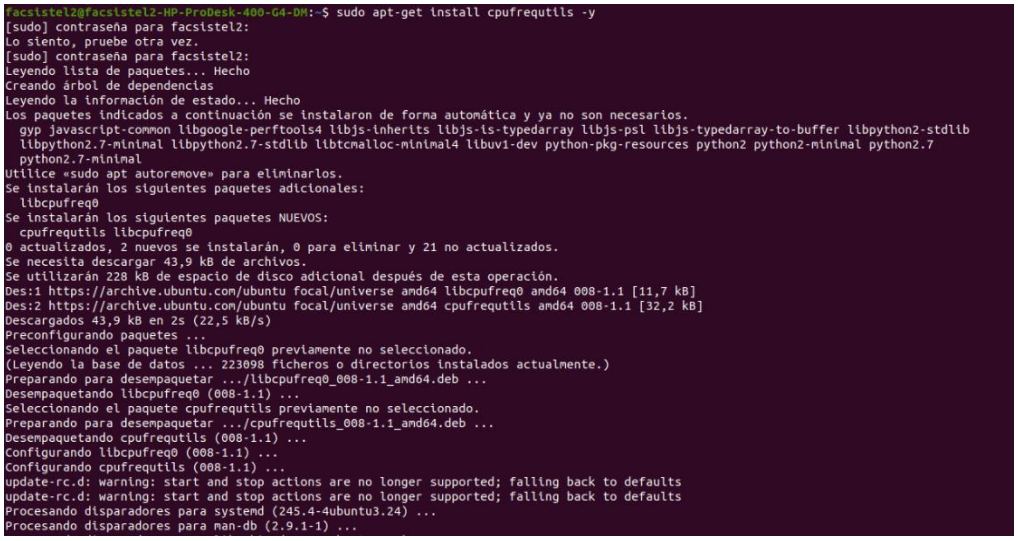


Figura 54 Instalación de cpufrequtils

Para verificar la configuración actual se realiza con los siguientes comandos:

```
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_available_governors
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
```

La primera línea de comandos es para verificar las configuraciones disponibles que se tiene al momento, y la segunda línea de comandos es para ver con cual se está trabajando.

En este caso se puede observar que tenemos dos configuraciones posibles performance y powersave, y verificamos que actualmente se está trabajando con powersave, como se puede observar en la Figura 55.

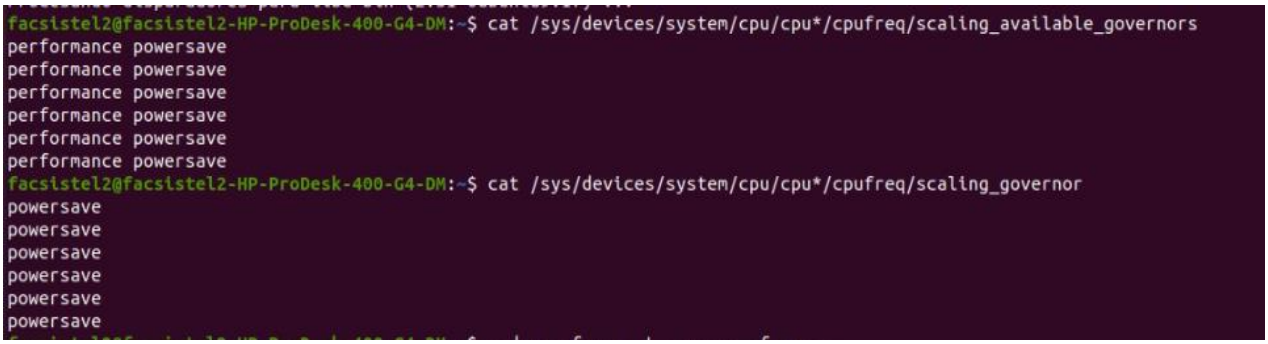


Figura 55 Configuración actual de la frecuencia

Al ejecutar un comando para cambiar a modo performance en la Intel i5 tuvimos un error, ya que solo un núcleo estaba en ese estado, pero queríamos todos en modo performance, pero se pudo solucionar mediante la manera directa mediante los siguientes comandos, que se describirán a continuación:

```
for cpu in /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor; do
    echo performance | sudo tee $cpu
done
```

Al momento de ejecutar esta línea pueda que se tenga que modificar algunas líneas que tiene que ver con la gestión de energía del computador para este caso se puede verificar con las siguientes líneas de código:

```
sudo systemctl stop thermald 2>/dev/null
sudo systemctl disable thermald 2>/dev/null
sudo systemctl stop power-profiles-daemon 2>/dev/null
sudo systemctl disable power-profiles-daemon 2>/dev/null
```

Para asegurar que la configuración se guarde correctamente se usara la siguiente línea para modificar el directorio:

```
sudo nano /etc/systemd/system/cpufreq.service
```

Se añadirán las siguientes líneas:

```
[Unit]
Description=CPU frequency scaling governor setup
After=network.target
[Service]
Type=oneshot
ExecStart=/bin/bash -c 'for cpu in /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor; do echo performance > $cpu; done'
RemainAfterExit=yes
[Install]
WantedBy=multi-user. target
```

Y luego verificamos que se haya cambiado la configuración correcta, mediante el comando:

```
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
```

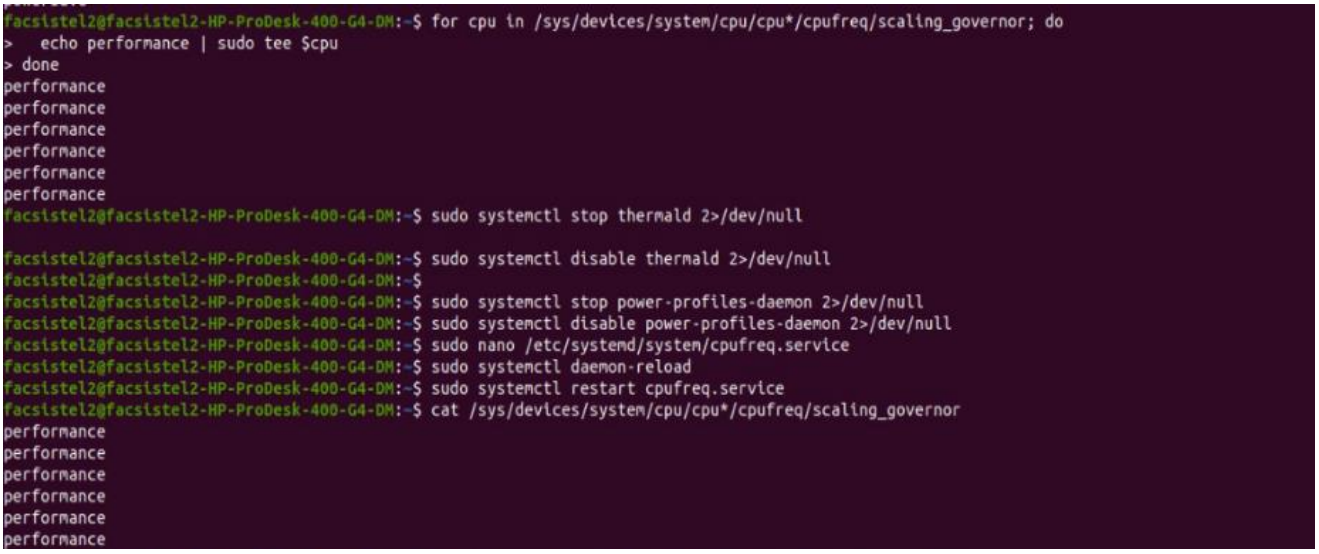


Figura 56 Configuración correctamente realizada

Como se puede observar en la figura 56, la configuración se realizó de manera correcta, y se aplicaran pasos similares para el otro computador, aunque necesariamente se prioriza más esta configuración en el computador que simule la eNodeB, para el otro computador los pasos que se siguieron fueron los siguientes:

```
sudo apt-get install cpufrequtils -y
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_available_governors
sudo cpufreq-set -r -g performance
echo 'GOVERNOR="performance"' | sudo tee /etc/default/cpufrequtils
sudo nano /etc/systemd/system/cpufreq.service
```

En la última línea de código se nos abrirá un script y copiaremos los siguiente:

```
[Unit]
Description=CPU frequency scaling governor setup
After=network.target
[Service]
Type=oneshot
ExecStart=/bin/bash -c 'for cpu in /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor; do echo performance > $cpu; done'
RemainAfterExit=yes
[Install]
WantedBy=multi-user. target
```

Habilitaremos los servicios, y verificamos que todo funcione correctamente

```
sudo chmod +x /etc/systemd/system/cpufreq.service
sudo systemctl daemon-reload
sudo systemctl enable cpufreq. service
sudo systemctl start cpufreq.service
# Deshabilitar el gestor de energía tlp si está instalado
sudo systemctl stop tlp
sudo systemctl disable tlp
# Configuración adicional para srsRAN
echo 'kernel.sched_rt_runtime_us = -1' | sudo tee -a /etc/sysctl.d/99-srsran.conf
sudo sysctl -p /etc/sysctl.d/99-srsran.conf
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
```

:

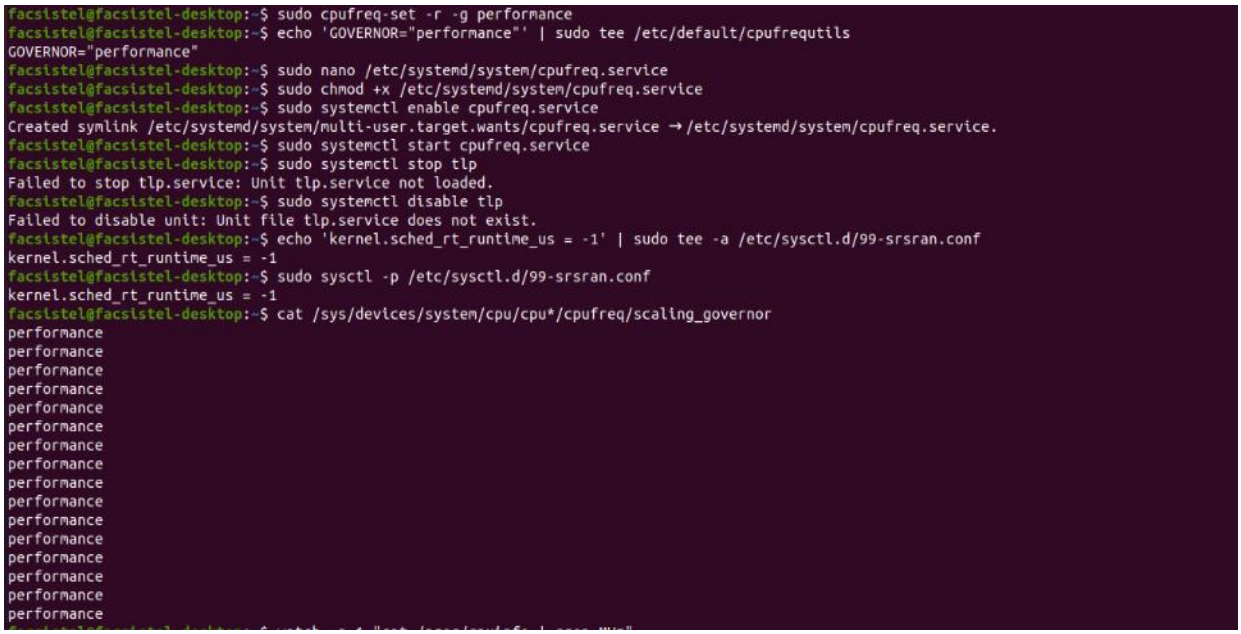


Figura 57 Modo Performance activado

Con la última línea de comando podemos verificar si está en modo performance

Figura 58 Frecuencia máxima y mínima de cada núcleo

Como último paso una vez activo el modo performance se verifica que el sistema este trabajando a su máxima capacidad, y podemos observar este comportamiento mediante las líneas de comando:

```
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_min_freq
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_max_freq
```

Como se puede observar en la figura 58, esta correctamente configurado y se pueden instalar los componentes para simular la red LTE.

3.4.3. Instalación del núcleo de red EPC

Para empezar con la instalación del Núcleo EPC, la primera recomendación es actualizar las librerías y actualizar las dependencias necesarias, esto se realizará con el siguiente comando:

```
sudo apt update
```

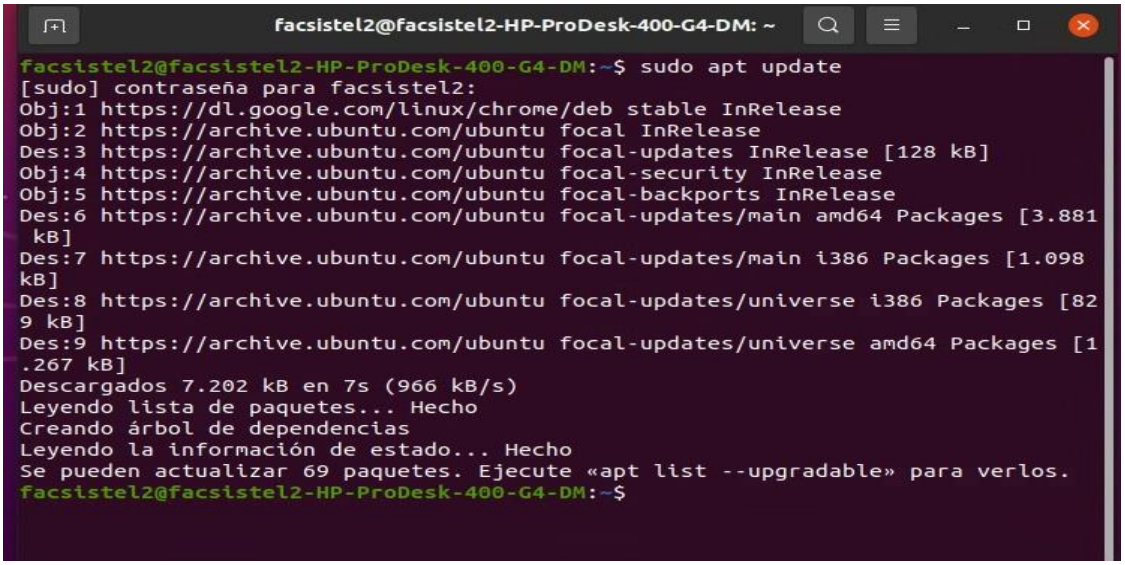


Figura 59 Repositorios listo para ser actualizados

Como se puede observar en la figura 59, se pueden actualizar 69 paquetes los cuales serán necesarios para el correcto funcionamiento, mediante el comando:

```
sudo list --upgradable
```

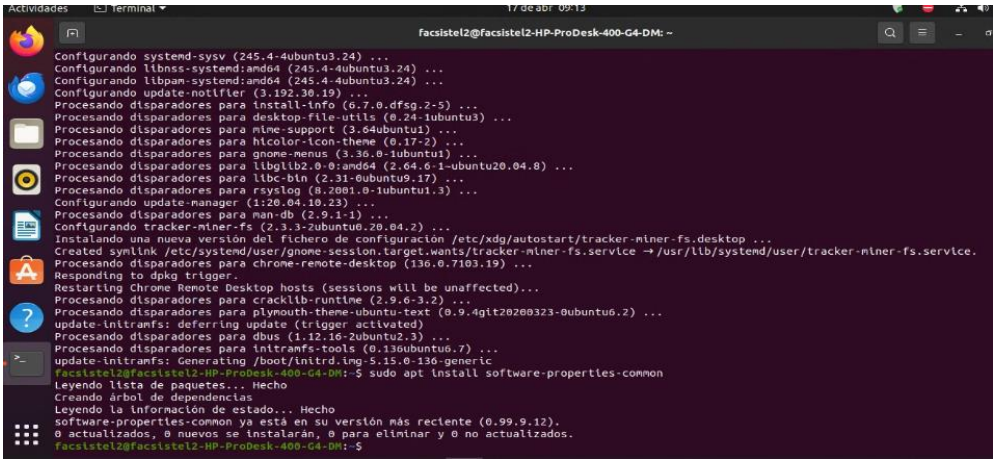


Figura 60 Repositorios actualizados y listos

Se ejecuta de esta manera porque necesitamos los permisos sudo ya que este comando dirá que somos administradores y ya no nos pedirá claves cada vez que intentemos ejecutar un comando, este comando es para realizar una actualización de los repositorios que pueden ser actualizados.

Una vez que se actualizan los paquetes, adicionalmente también ejecutamos el comando:

```
sudo apt install software-properties-common
```

Con el fin de gestionar paquetes que no están disponibles en repositorios oficiales, y nos ayudara instalaciones de programas.

Una vez hecho eso nuestro sistema estará listo para funcionar, pero debido a que vamos a instalar un software especializado, necesitaremos muchas dependencias nuevas y las instalaremos con los siguientes comandos:

```
sudo apt update

sudo apt install python3-pip python3-setuptools python3-wheel ninja-build build-essential flex bison git cmake libsctp-dev libgnutls28-dev libgcrypt-dev libssl-dev libidn11-dev libmongoc-dev libbson-dev libyaml-dev libnghttp2-dev libmicrohttpd-dev libcurl4-gnutls-dev libnghttp2-dev libtins-dev meson
```

Esta actualización de librerías puede demorar minutos dependiendo del procesador y velocidad de conexión a internet que se tenga.

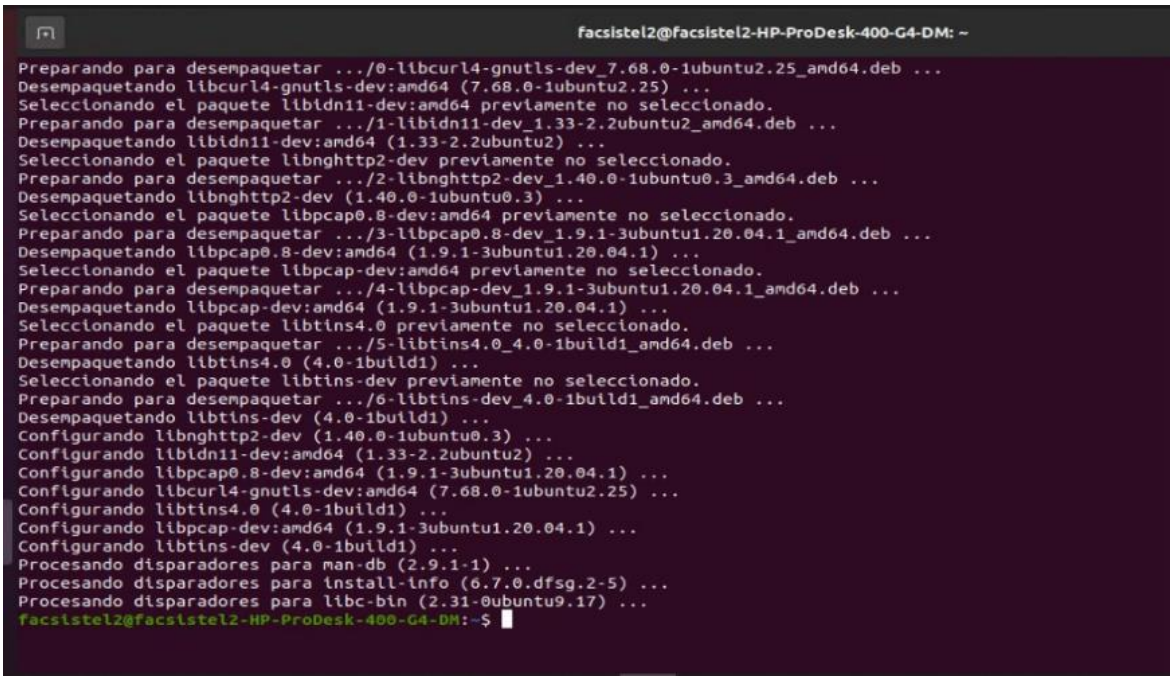


Figura 61 Dependencias instaladas

Como podemos observar en la figura 61, las dependencias se instalaron sin ningún error y ahora estamos listo para instalar diferentes complementos antes de instalar el Software Open5GS

3.4.3.1.Instalación MongoDB (Base de datos)

Primero se instalará la base de datos que es esencial para la arquitectura en ellos almacenaremos los datos de los usuarios, utilizaremos los siguientes comandos:

```
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add –
apt-key list
```

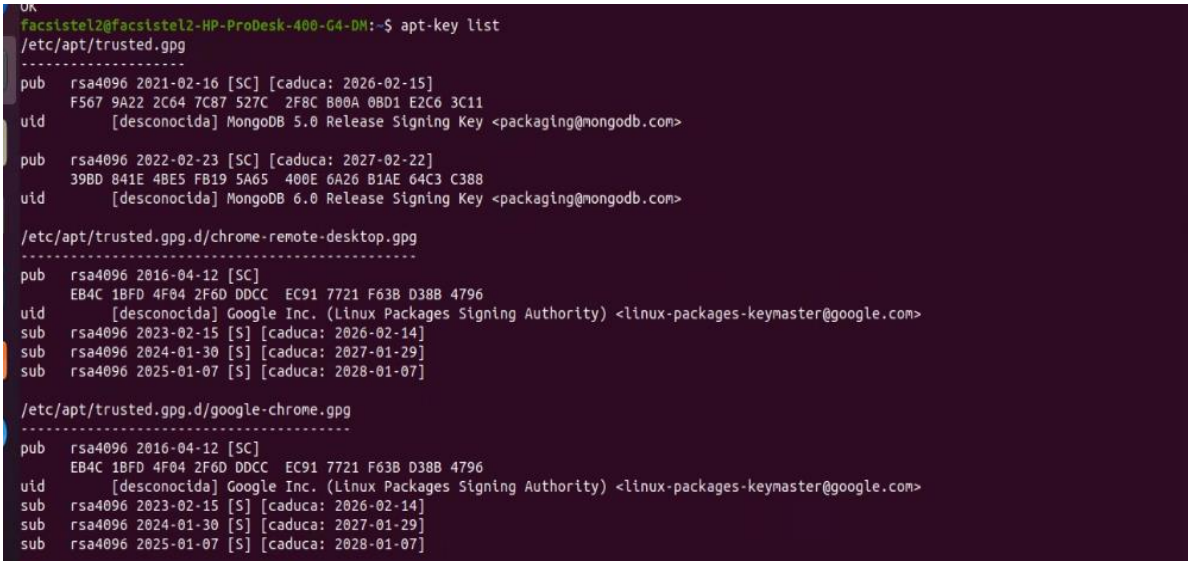


Figura 62 Salida de los comandos para instalar MongoDB

Como se puede observar en la Figura 62, con el primer comando nos devolverá como salida Ok, y con segundo comando nos dará una lista y proporcionará una clave, la cual la tendremos que añadir a los repositorios del sistema, y luego actualizamos los repositorios con el siguiente comando:

```
$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
$ sudo apt update
```

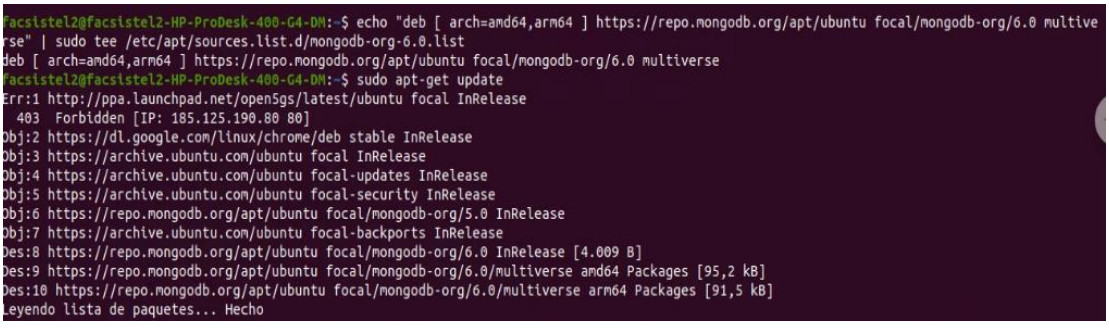


Figura 63 Base de datos activa

Cuando tengamos descargados las dependencias necesarias vamos a ejecutar el comando para instalar Mongo DB 6.0 ya que es una versión estable para el Open5GS.

```
sudo apt-get install -y mongodb-org
```

Figura 64 Instalación de Mongo DB

Una vez realizada la instalación tendremos la base de datos Mongo DB en su versión 6.0 esto lo hacemos porque a veces los repositorios están desactualizados y nos puede dar problemas, vamos a verificar que esté funcionando correctamente con los comandos:

```
sudo systemctl start mongod.service

sudo systemctl status mongod
```

Figura 65 Instalación correcta de la base de datos

Se observa que en la Figura 65, dice que la base de datos esta activa lo cual quiere decir que se instaló correctamente y será la principal que gestionará el almacenamiento de los usuarios.

3.4.3.2.Configuración base de datos

Para conectarnos a la base de datos necesitamos usuarios y estos se configuran almacenándolos en la base de datos de MongoDB y para verificar suscriptores se utilizan los siguientes comandos:

```
mongosh --eval 'db = db.getSiblingDB("open5gs"); db.subscribers.find().pretty()'
mongosh --eval 'db = db.getSiblingDB("open5gs"); const subscribers = db.subscribers.find().toArray();
if(subscribers.length === 0) { print("No hay suscriptores en la base de datos"); } else {
print(JSON.stringify(subscribers, null, 2)); }'
```

Este comando nos garantizara que hallan usuarios aparecerán en la base y nos aparecerá un mensaje "No hay Suscriptores en la base de datos".

```
mongosh --eval 'db = db.getSiblingDB("open5gs"); db.subscribers.deleteOne({imsi: "001010000000001"})'
```

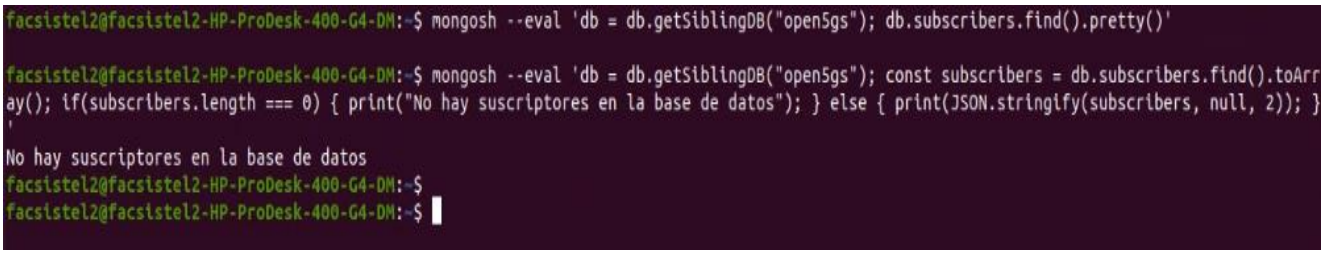


Figura 66 Verificar suscriptores en la base de datos

Para eliminar usuarios cuando ya no los necesitemos o ya no formen parte de la red LTE, se eliminan con el siguiente comando:

Figura 67 Eliminar un usuario

Como se puede observar en la figura 67, para ellos necesitamos el código IMSI para poder eliminarlo, de lo contrario no se podrá eliminar ni modificar.

3.4.4. Instalación de Open5GS

Para ello primero descargaremos la versión que se encuentra en los repositorios de GitHub, y luego accederemos a la carpeta descargada, mediante los comandos:

```
git clone https://github.com/open5gs/open5gs
cd open5gs
```

Figura 68 Instalación de repositorio GitHub

Ahora para descargar una de las versiones recientes usaremos el comando:

```
git pull
```

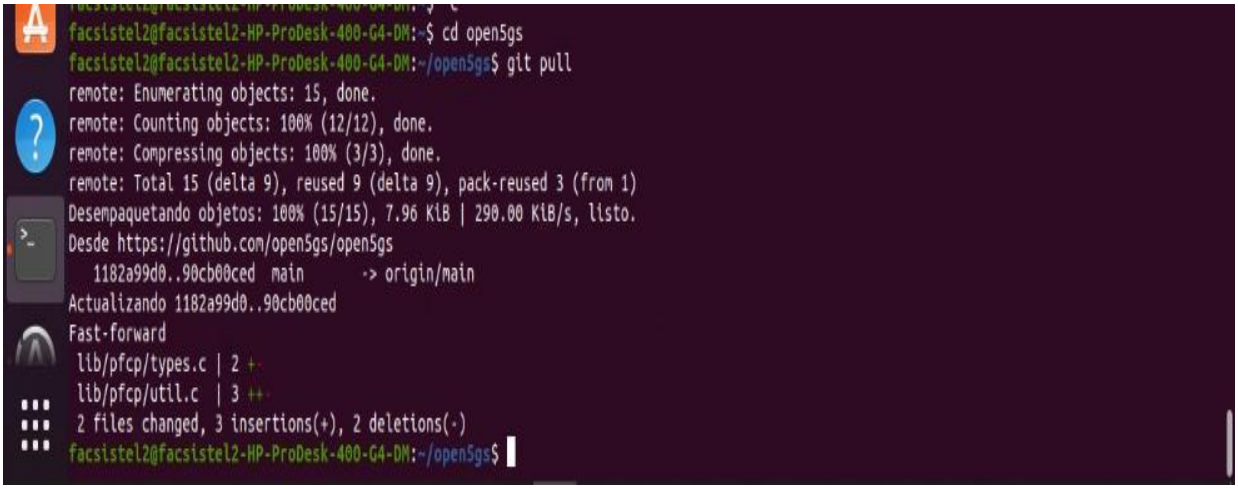


Figura 69 Salida del comando Git Pull

Usaremos el siguiente comando para construir nuestro proyecto se creará una carpeta llamada instaló donde se ubicará todo las configuraciones y entidades que se creen, se ejecuta mediante instrucción:

```
meson build --prefix=`pwd`/install
```

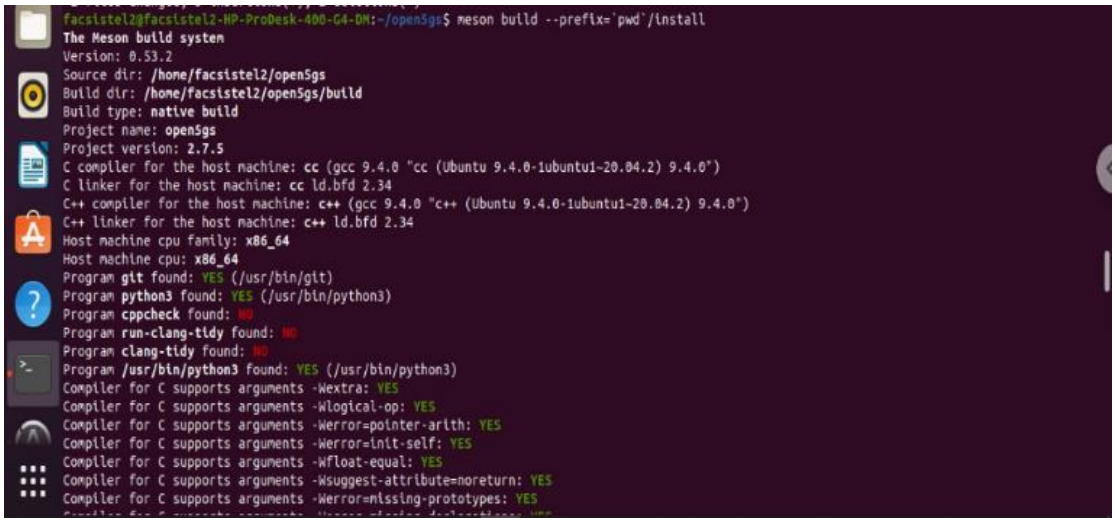


Figura 70 Construcción de la carpeta install

Por último, para finalizar la instalación de las carpetas usaremos los siguientes comandos:

```
ninja -C build
sudo ninja -C build install
```

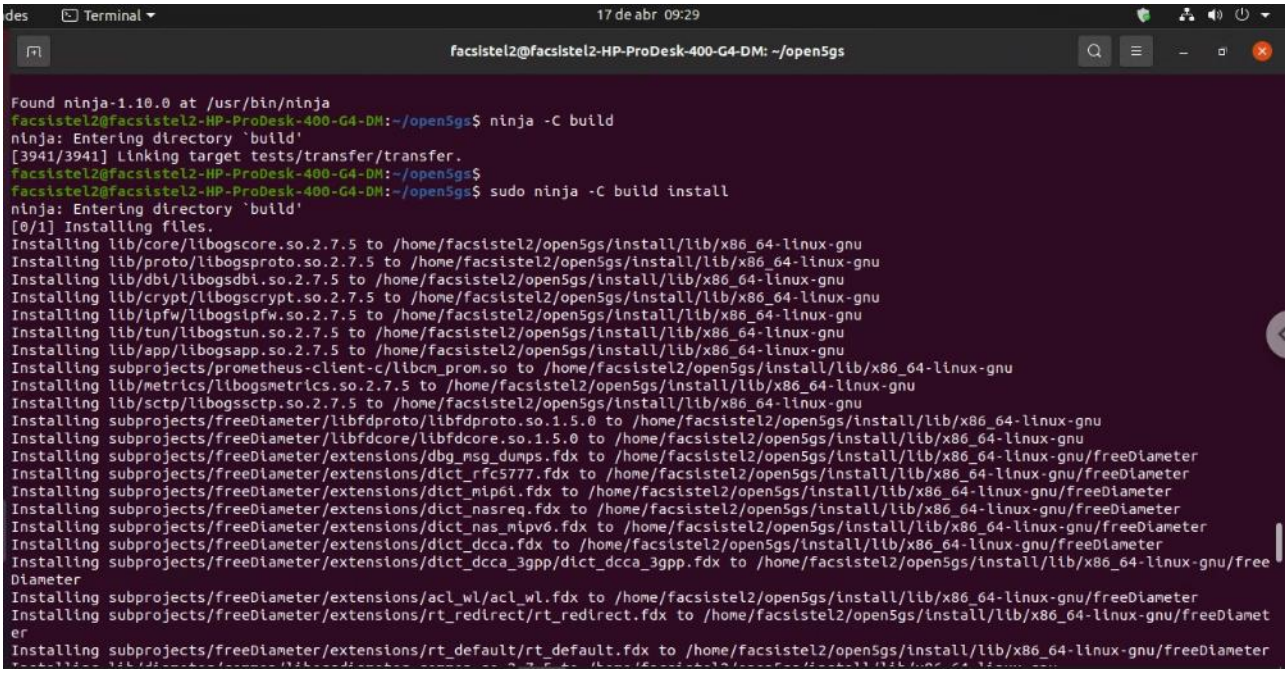


Figura 71 Salida del comando build

Como se puede observar en la Figura 71, primero se van a transferir archivos y luego se construirá el proyector en la carpeta build estos nos darán los archivos necesarios para la configuración de las entidades y su funcionamiento.

Y finalmente ejecutaremos el comando:

```
sudo ldconfig
```

Figura 72 Comando idconfig

Como se instaló Open5GS mediante el código fuente con mesón, los archivos de configuración de las entidades estarán en otra ruta precisamente en la carpeta Install buscaremos nuestros archivos de configuración con los siguientes comandos:

```
cd open5gs/install/etc/open5gs/
ls
```

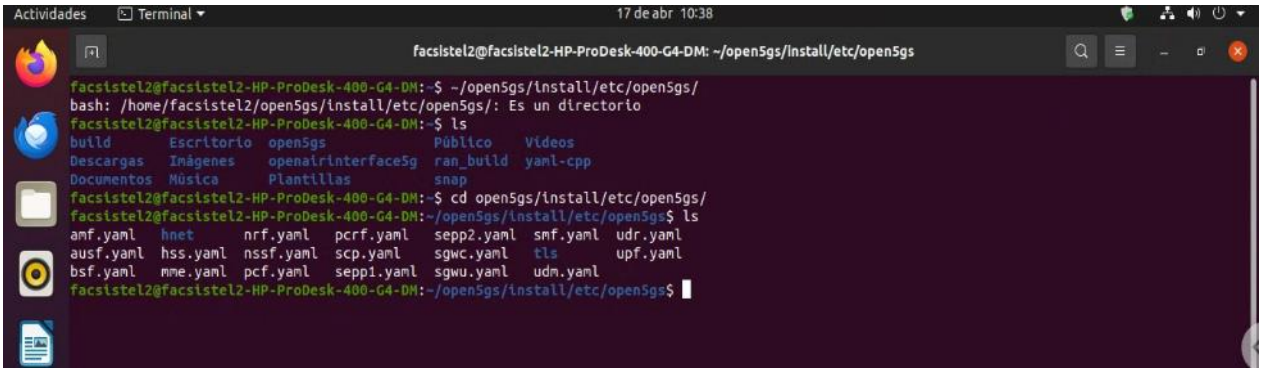


Figura 73 Ruta correcta de los archivos de configuración

Como se puede observar en la figura 73, los archivos de configuración de las entidades aparecen en otra carpeta, pero podemos moverlos para evitar errores, con las siguientes instrucciones:

```
sudo mkdir -p /etc/open5gs
sudo cp /home/facsistel2/open5gs/build/configs/systemd/*.service /etc/systemd/system/
find -/openg5gs -name “*.service”
```

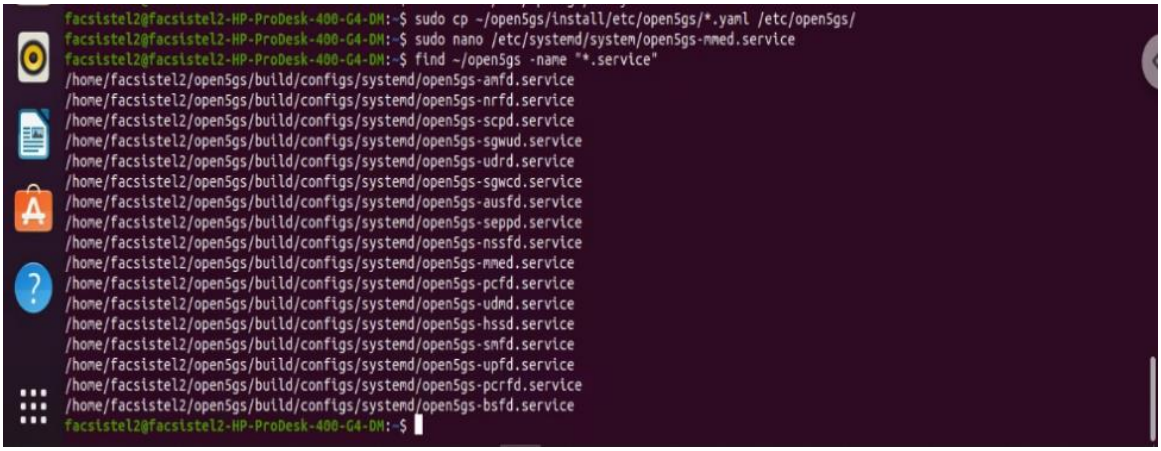


Figura 74 Rutas correctas establecidas

En la figura 74, se puede ver los archivos de configuración en las rutas correctas, los buscamos mediante el comando find y aparecieron donde las configuramos.Como cambiamos las rutas de las entidades tenemos que apuntar a la dirección correcta, esto se hace mediante el comando:

```
sudo nano /etc/systemd/system/open5gs-mmed.service
```

Y modificaremos la línea que dice ExecStart, y colocaremos la ruta correcta, y colocaremos un usuario para evitar quedando de esta manera:

```
sudo adduser --system --group open5gs
ExectStart:/home/facsistel2/open5gs/install/bin/open5gs-mmed
```

Figura 75 Archivo de configuración de la MME

Haremos esta configuración con cada entidad que vallamos a ejecutar, pero podemos hacerlo de manera automática con un comando

```
sudo sed -i 's|ExecStart=.*|ExecStart=/home/facsistel2/open5gs/install/bin/open5gs-mmed|g' /etc/systemd/system/open5gs-mmed.service

sudo sed -i 's|ExecStart=.*|ExecStart=/home/facsistel2/open5gs/install/bin/open5gs-hssd|g' /etc/systemd/system/open5gs-hssd.service

sudo sed -i 's|ExecStart=.*|ExecStart=/home/facsistel2/open5gs/install/bin/open5gs-sgwcd|g' /etc/systemd/system/open5gs-sgwcd.service

sudo sed -i 's|ExecStart=.*|ExecStart=/home/facsistel2/open5gs/install/bin/open5gs-sgwud|g' /etc/systemd/system/open5gs-sgwud.service

sudo sed -i 's|ExecStart=.*|ExecStart=/home/facsistel2/open5gs/install/bin/open5gs-smfd|g' /etc/systemd/system/open5gs-smfd.service

sudo sed -i 's|ExecStart=.*|ExecStart=/home/facsistel2/open5gs/install/bin/open5gs-upfd|g' /etc/systemd/system/open5gs-upfd.service

sudo sed -i 's|ExecStart=.*|ExecStart=/home/facsistel2/open5gs/install/bin/open5gs-pcrfd|g' /etc/systemd/system/open5gs-pcrfd.service
```

Una vez que hallamos editarlo archivo de servicio recargamos mediante el comando:

```
sudo systemctl daemon-reload
```

Ahora habilitamos y arrancamos los servicios necesarios para configurar la red LTE, mediante las siguientes líneas:

```
sudo systemctl enable open5gs-mmed open5gs-hssd open5gs-sgwcd open5gs-sgwud open5gs-smfd open5gs-upfd open5gs-pcrfd

sudo systemctl start open5gs-mmed open5gs-hssd open5gs-sgwcd open5gs-sgwud open5gs-smfd open5gs-upfd open5gs-pcrfd
```

3.4.5. Configuración de la MME

Una vez configurado de manera correcta nuestro entorno, ahora si vamos a configurar las diferentes entidades, la primera entidad que se va a configurar será la MME, podremos realizar la configuración mediante el comando:

```
sudo nano ~/open5gs/install/etc/open5gs/mme.yaml
```

En este archivo de configuración nos enfocaremos en la dirección IP y en la PLMN que son vitales para la comunicación con el eNodeB.

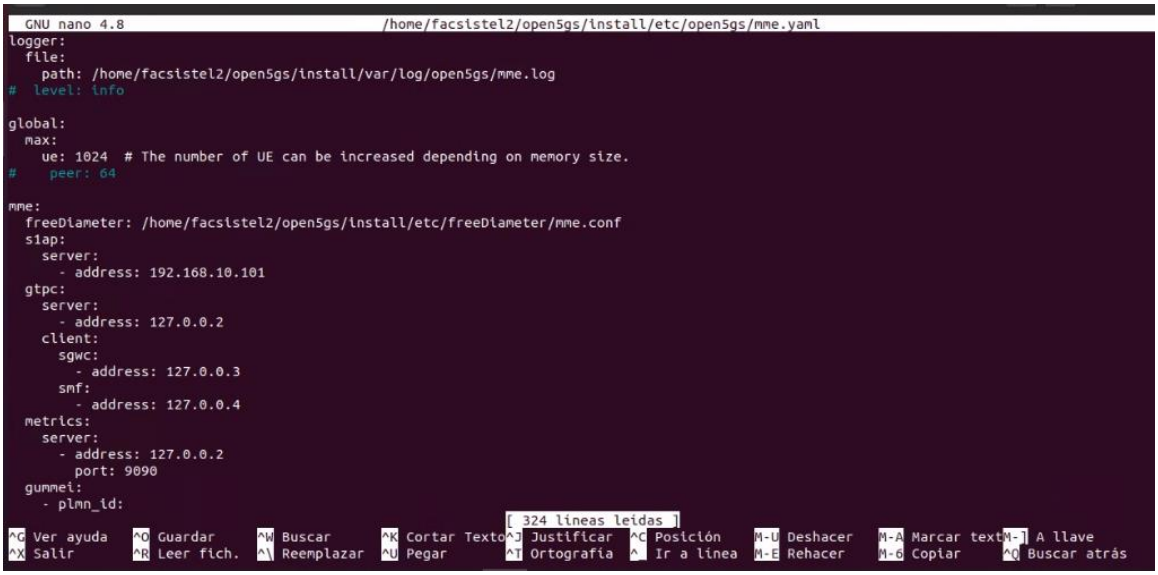


Figura 76 Configuración de la entidad MME

Como podemos observar en la figura 76, vamos a colocar la dirección IP del computador en este caso la 192.168.10.101, esta dirección será importante ya que con esta tendremos que configurar el eNodeB para establecer la correcta comunicación. Ahora vamos a ver las configuraciones de la PLMN:

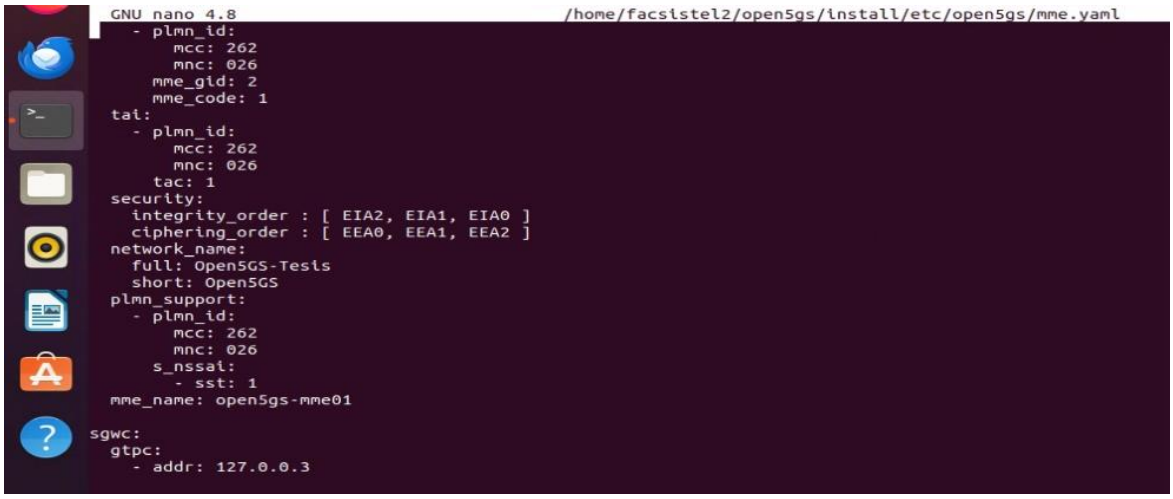


Figura 77 Configuración de la PLMN

En la figura 77, las PLMN tiene un MCC y MNC estos son códigos de usuarios genéricos generados aleatoriamente como prueba, una vez que finalice la configuración de la MME se digitaran los siguientes comandos:

Con estos comandos daremos permisos para poder editar sin ser un super usuario, son necesarios al tener que ejecutar comandos con sudo.

```
sudo mkdir -p /home/facsistel2/open5gs/install/var/log/open5gs/
sudo chown -R $USER:$USER /home/facsistel2/open5gs/install/var/log
```

Y ahora si iniciaremos la entidad MME con el siguiente comando:

```
cd ~/open5gs/install/bin/
sudo ./open5gs-mmed -c ~/open5gs/install/etc/open5gs/mme.yaml
```

Figura 78 Arranque de la MME

Como se puede observar en la figura 78, la MME inicio sin problemas y la finalizamos sin problemas, luego se optimizará la entidad para el despliegue completo.

3.4.6. Configuración HSS

Para la entidad HSS replicaremos pasos similares con el comando:

```
sudo nano ~/open5gs/install/etc/open5gs/hss.yaml
```

Realizaremos configuraciones respectivas y guardaremos las configuraciones:

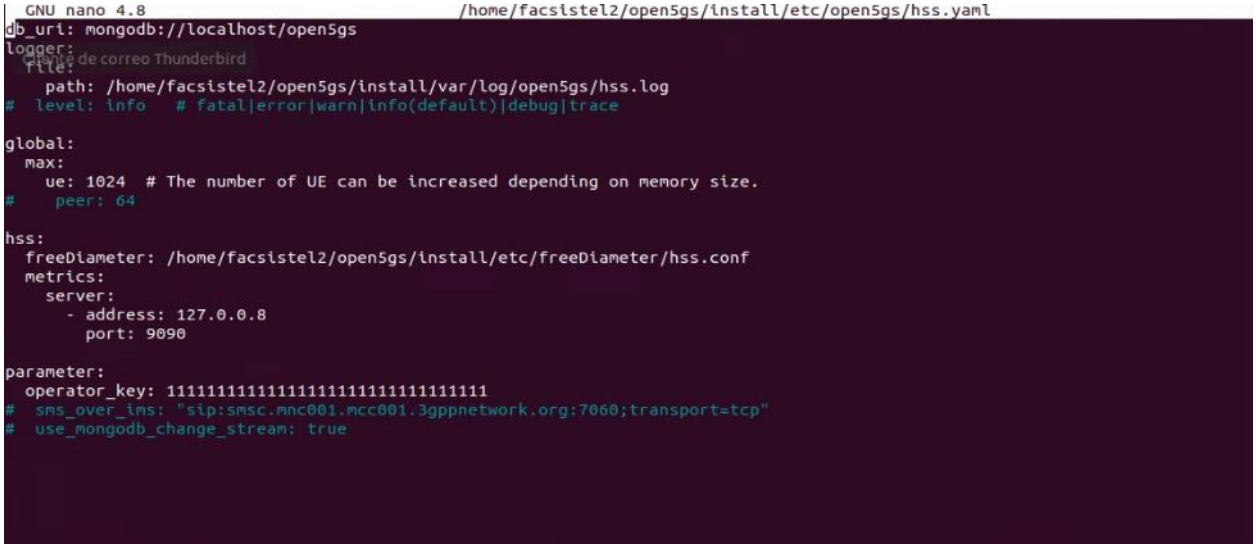


Figura 79 Configuración de la HSS

Al guardar la configuración, guardaremos y ejecutaremos la entidad para ver su correcto funcionamiento:

```
sudo ./open5gs-hssd -c ~/open5gs/install/etc/open5gs/hss.yaml
```

Figura 80 Funcionamiento de la entidad HSS

Se puede observar en la figura 80 que conecta correctamente con la base de datos Mongo DB, y nos da mensajes de error lo cual quiere decir que esta correctamente configurada.

3.4.7. Configuración SGW

Para configurar esta entidad se divide en dos partes uno es la SGW-U, que funciona para el plano de usuario y el otro SGW- C que funciona para el plano de control

3.4.7.1. Configuración del SGW-U

Como primer punto para configurar esta entidad configuraremos la que este encargado del plano de usuario para ello usaremos el siguiente comando:

```
sudo nano ~/open5gs/install/etc/open5gs/sgwu.yaml
```

No realizaremos modificaciones ya que es un entorno de pruebas y se conectaran de manera instantánea, la configuración por defecto queda de la siguiente manera:

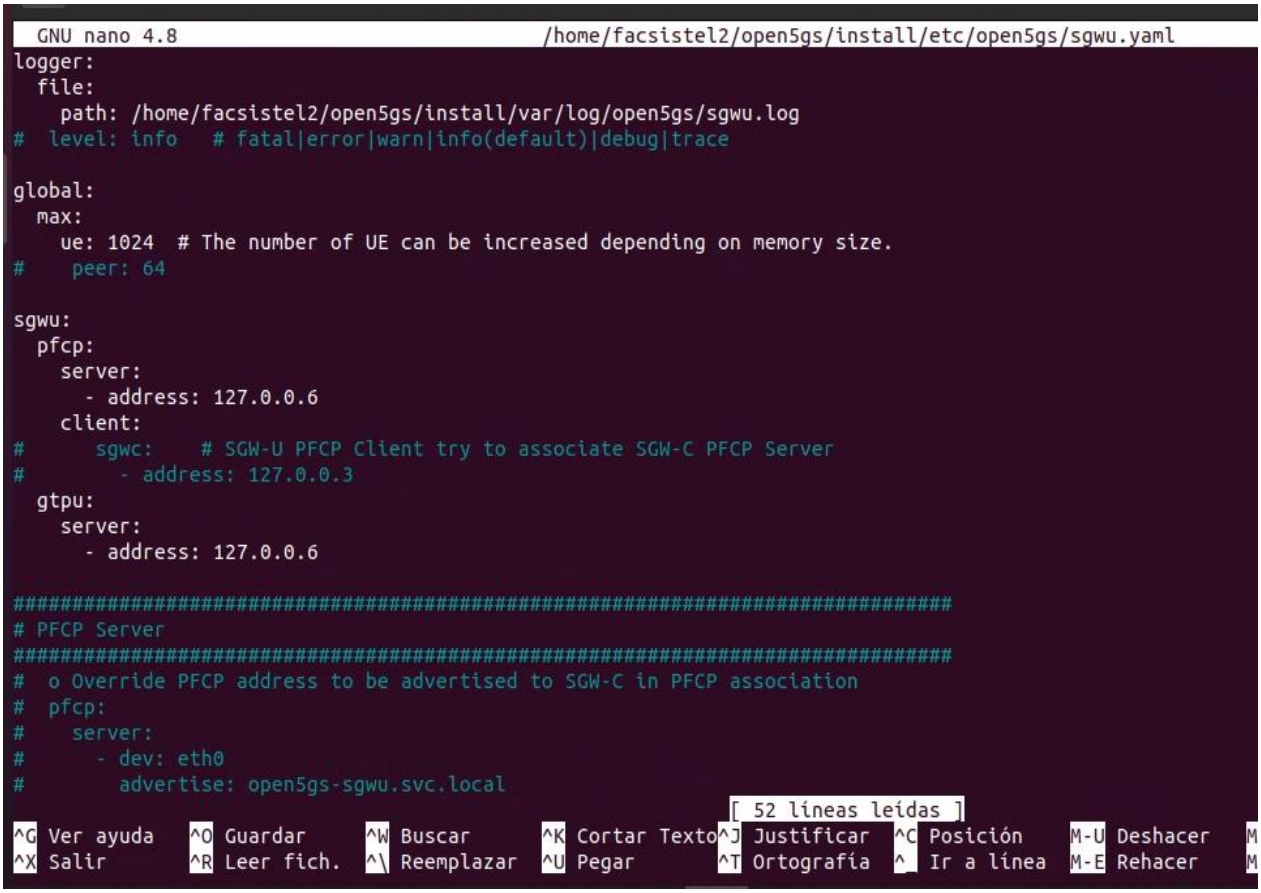


Figura 81 Configuración de la SGWU

3.4.7.2.Configuración de la SGWC

Para la configuración de la SGWC que conforma el plano de control con el comando:

```
sudo nano ~/open5gs/install/etc/open5gs/sgwc.yaml
```

Dejaremos por defecto el script y guardaremos la configuración:

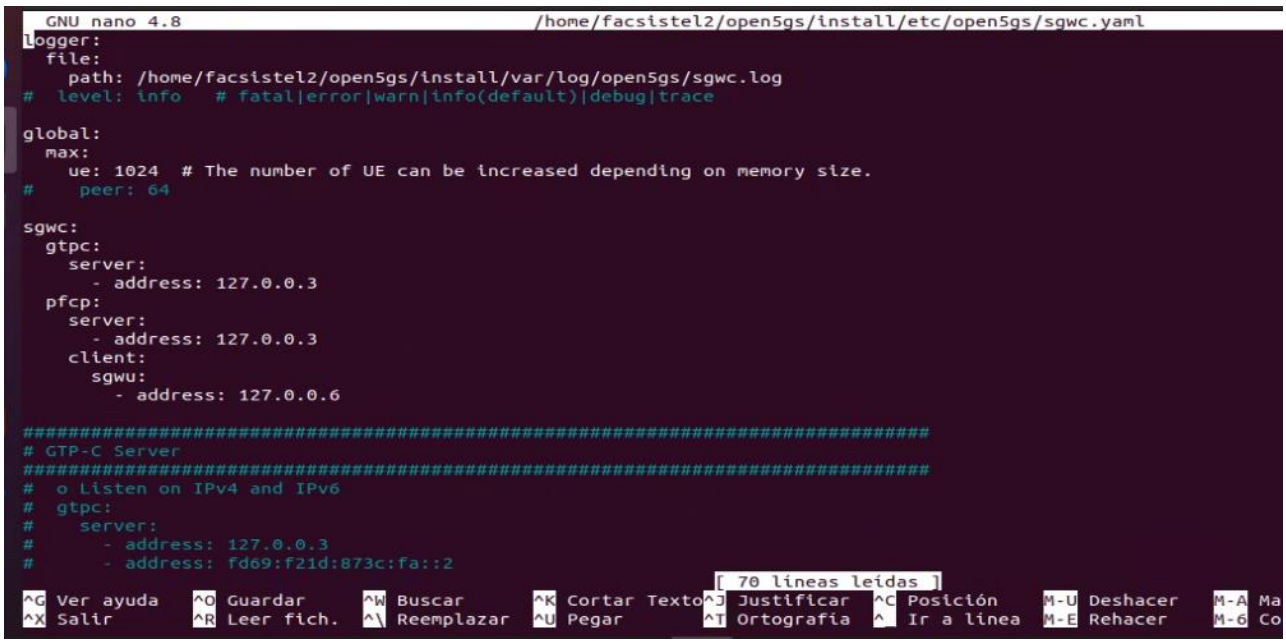


Figura 82 Configuración de la SGWC

Ahora vamos a ejecutar las dos entidades a la vez, ya que si eso no se hace fallarán y nos darán errores, porque una depende de la otra entidad para funcionar, iniciaremos el servicio con los siguientes comandos:

```
sudo ./open5gs-sgwcd -c ~/open5gs/install/etc/open5gs/sgwc.yaml
sudo ./open5gs-sgwud -c ~/open5gs/install/etc/open5gs/sgwu.yaml
```

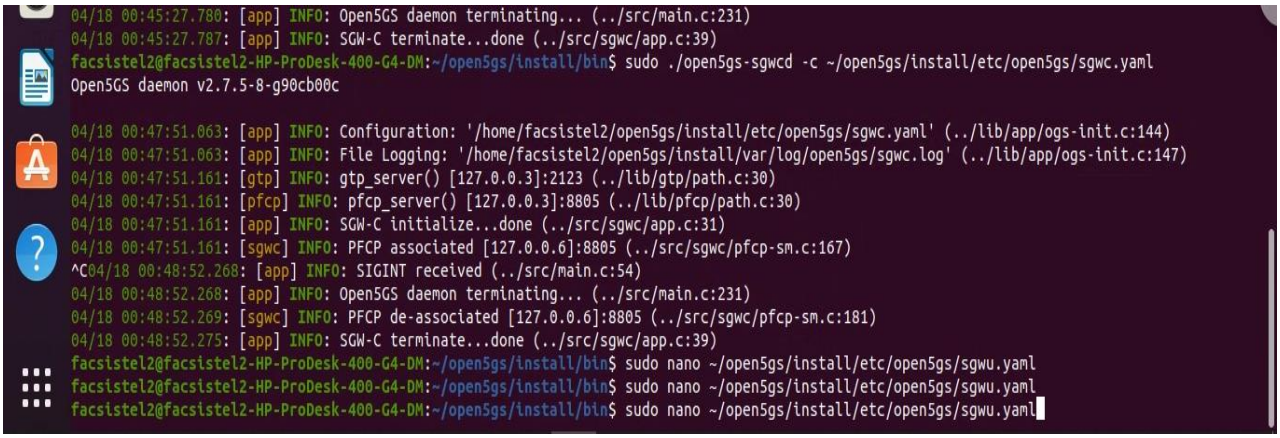


Figura 83 Ejecutando SGWU y SGWC

Como se observa en la Figura 83, ambos servicios dependen del otro si uno tiene errores el otro no podrá comunicarse se realizó la comunicación mediante PFCP y ambos servicios se escuchan entre sí y para confirmar estos muestran los siguientes mensajes:

"SGW-U inicializado...hecho"

"SGW-C inicializado...hecho"

3.4.8. Configuración de la PGW

Para configurar la PGW se configura de manera similar que la SGW, pero las entidades para el plano de control se llaman SMF (Sessión Management Functión), y la entidad encargada del plano de usuario es UPF (User Plane Functión). Esto sucede porque usamos un repositorio basado en 5G, pero también compatible Para formar la EPC.

3.4.8.1.Configuración de la SMF

Para la configuración de la SMF usaremos el siguiente comando:

```
sudo nano ~/open5gs/install/etc/open5gs/smf.yaml
```

Guardaremos la configuración por defecto y esperamos para levantar la red si la levantamos en el momento nos dará un error porque también depende de UPF.

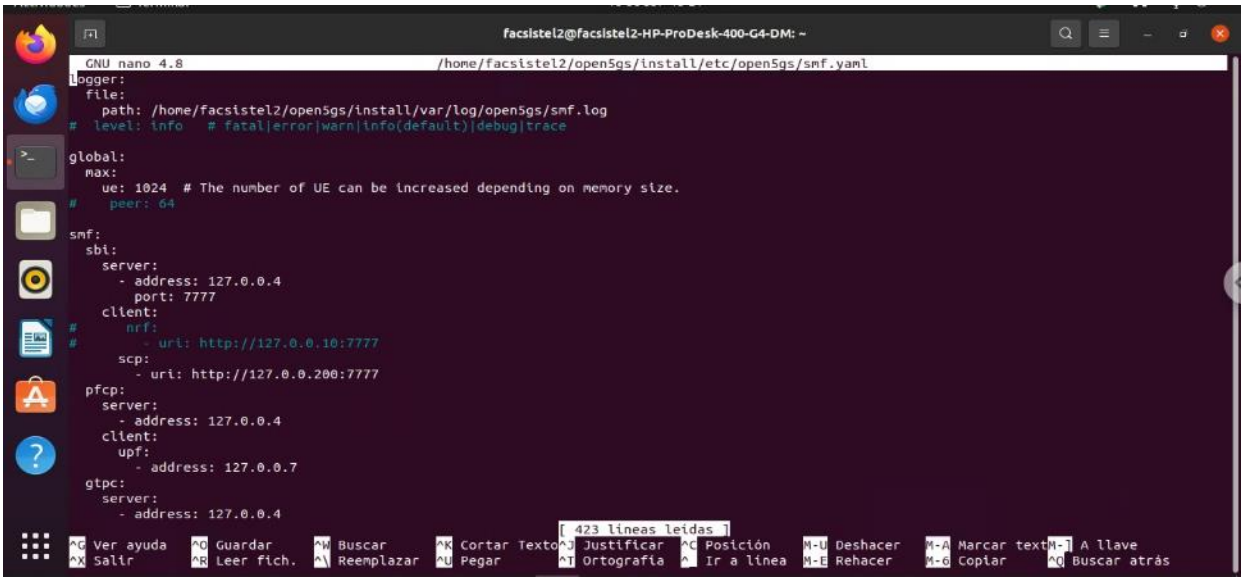


Figura 84 Configuración del SMF

3.4.8.2. Configuración de la UPF

Ahora configuraremos la entidad UPF que es para el plano de usuario, mediante el siguiente comando:

```
sudo nano ~/open5gs/install/etc/open5gs/upf.yaml
```

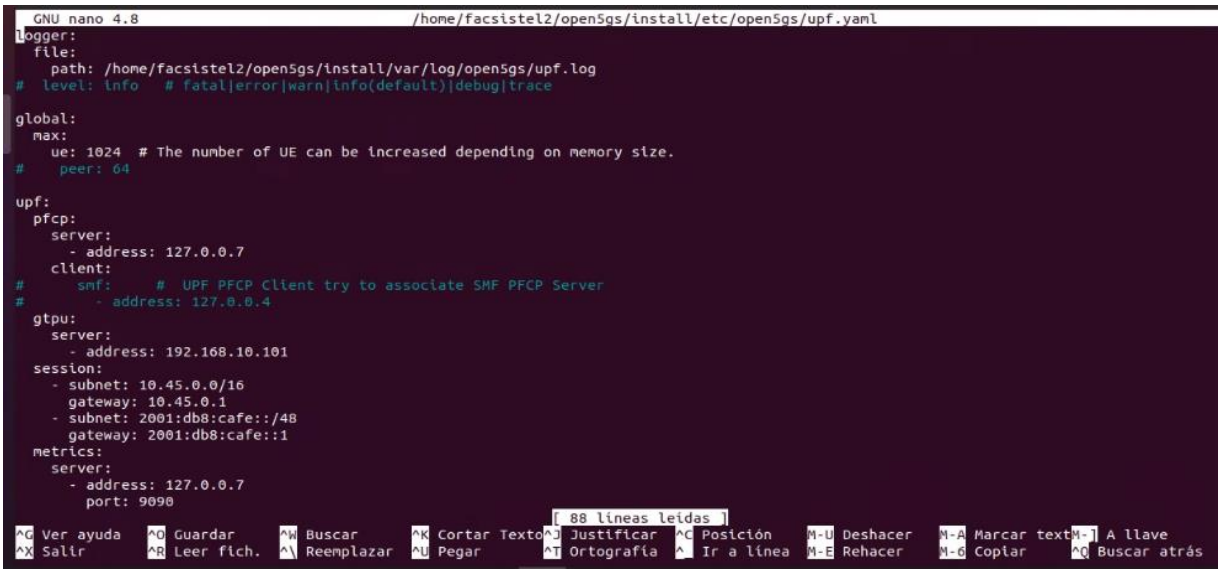


Figura 85 Configuración de la UPF

Dejaremos la configuración por defecto, así como se puede observar en la Figura 3-63 para ejecutar estas dos entidades a la vez se hará uso de los siguientes comandos:

```
sudo ~/open5gs/install/bin/open5gs-upfd -c ~/open5gs/install/etc/open5gs/upf.yaml
sudo ~/open5gs/install/bin/open5gs-smfd -c ~/open5gs/install/etc/open5gs/smf.yaml
```

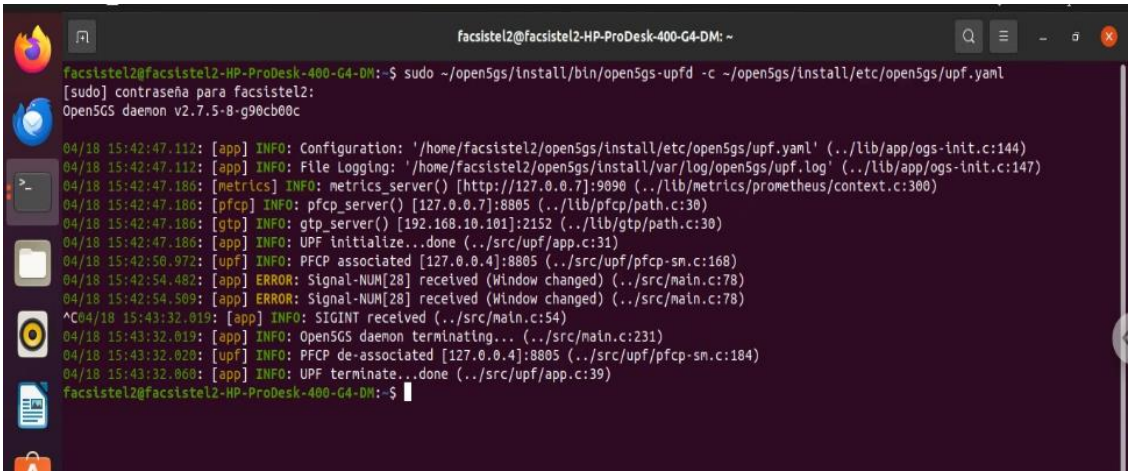


Figura 86 Arranque de la UPF

Como se puede en la figura 86, una vez ejecutado saldrá errores como de la pantalla grande o sin mayor importancia la línea que queremos ver es la siguiente:

```
INFO: PFCP associated [127.0.0.4]:8805 (../src/upf/pfcp-sm.c:168)
```

Esta línea nos indica, que se pudo establecer con éxito la conexión de asociación PFCP entre el SMF y el UPF.

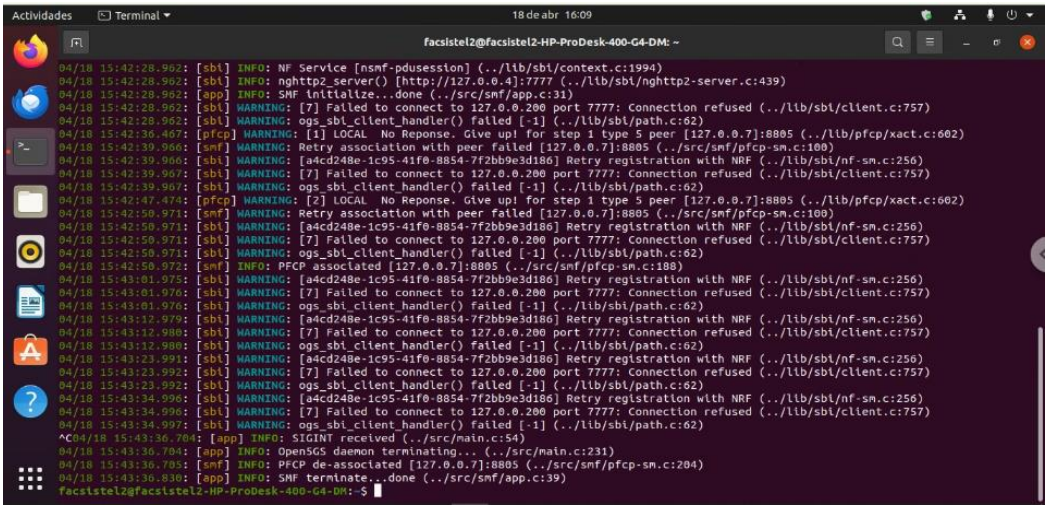


Figura 87 Ejecución de SMF

Como se puede observar en la figura 87, en la configuración de la entidad SMF se puede observar warnings que no se pudo establecer conexión con la NRF porque es un componente de 5G, y no nos afecta en la funcionalidad de nuestro EPC.

3.4.9. Configuración del PCRF

Y por último configuraremos la PCRF (Policy and Charging Rules Función), mediante el comando:

```
sudo nano ~/open5gs/install/etc/open5gs/pcrf.yaml
```

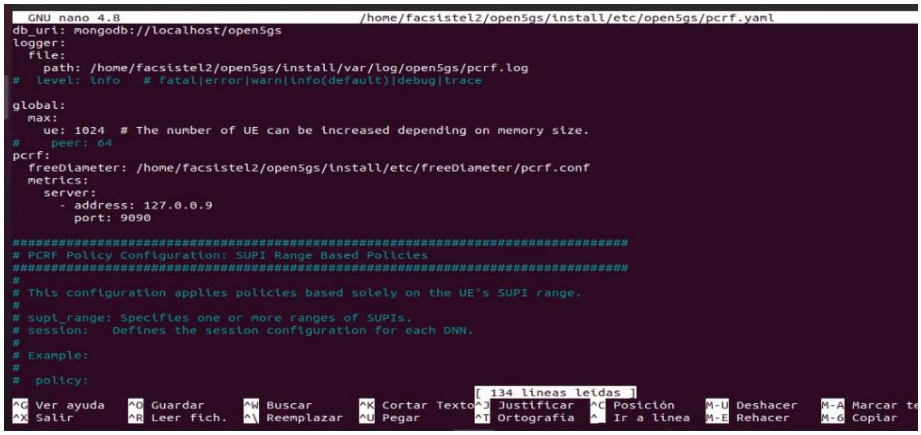


Figura 88 Código de la entidad PCRF

Dejamos por defecto la configuración e iniciamos la entidad.

```
sudo ~/open5gs/install/bin/open5gs-pcrfd -c ~/open5gs/install/etc/open5gs/pcrf.yaml
```

Figura 89 Ejecución de la entidad PCRF

Se puede observar en la Figura 89 La configuración muestra que se cargó correctamente, se conecta a la base de datos polling FreeDiameter está configurado. Una vez que configuramos todas las entidades correctamente podemos arrancar la EPC y lo tenemos listo para funcionar con el eNodeB.

3.4.10. Configuración del EPC

Como último punto en vez de andar ejecutando nuestro EPC de manera individual componente por componente, se creará un script capaz de iniciar todos los componentes a la vez de la siguiente manera, por medio del comando se creará un script nuevo:

```
nano ~/start-epc.sh
```

Y escribiremos lo siguiente:

Figura 90 Script para ejecutar todos los servicios a la vez

Como observamos en la figura 90, se realiza un script para iniciar todo a través de un solo comando y nos enviara resultados a la terminal de que todos los componentes del EPC iniciados, y podremos ver procesos, verificar puertos. Haremos este script ejecutable mediante el comando:

```
chmod +x ~/start-epc.sh
sudo ~/start-epc.sh
```

Adicionalmente se creará un script para detener la EPC con el siguiente comando:

```
nano ~/stop-epc.sh
chmod +x ~/stop-epc.sh
sudo ~/stop-epc.sh
```

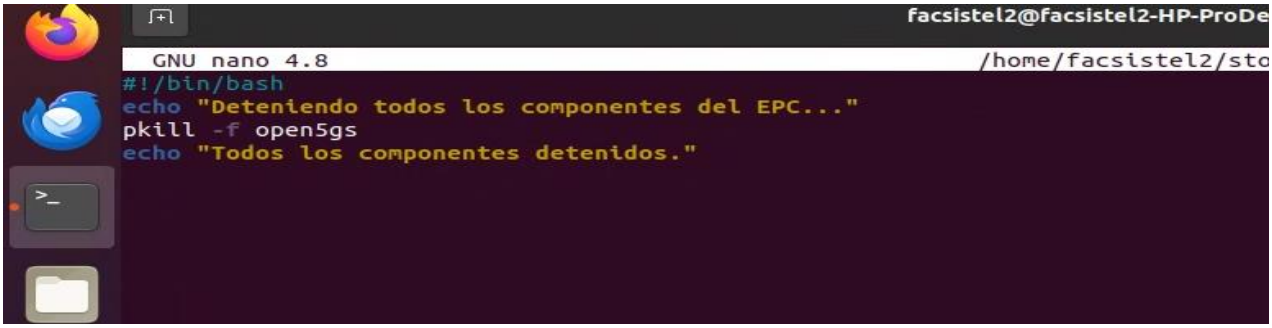


Figura 91 Script para detener la EPC

Vamos a ejecutar el script para iniciar el EPC mediante:

```
sudo ~/start-epc.sh
```

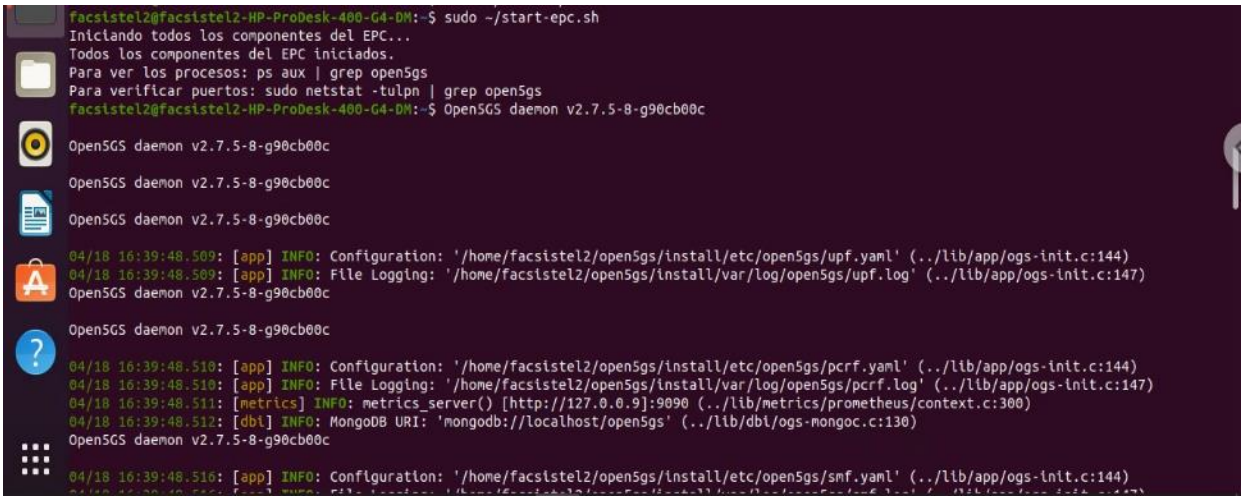


Figura 92 EPC iniciada

Para detener la EPC se ejecuta el comando:

```
sudo ~/stop-epc.sh
```

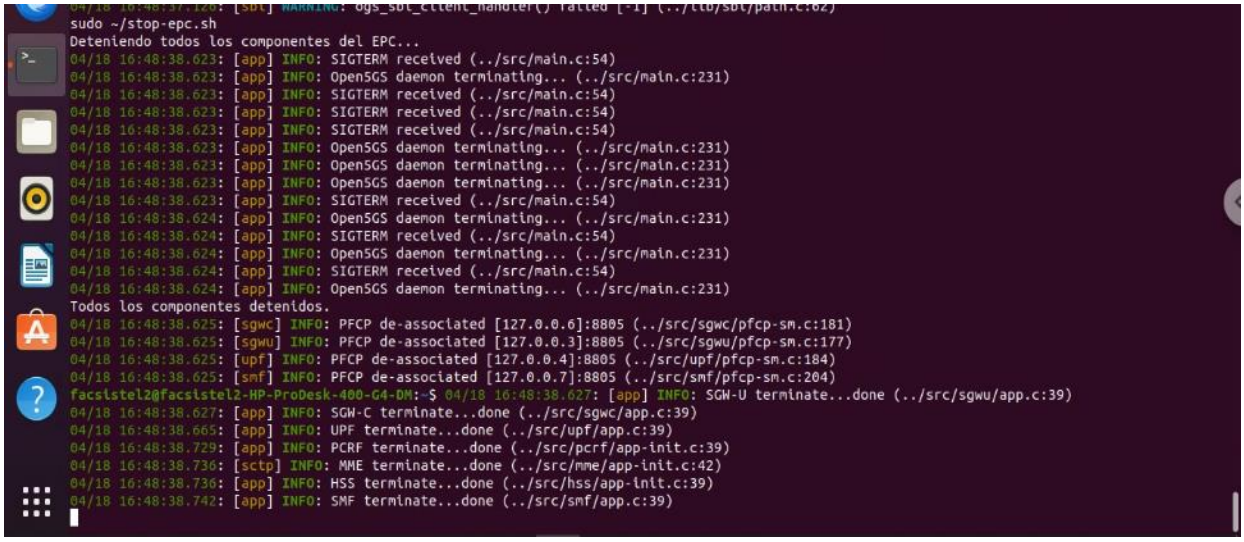


Figura 93 EPC detenida

Como se observa en la figura 93, vemos que la EPC se detenido correctamente, y está lista para conectar con el eNodeB, para ver procesos relacionados con la EPC, esto quiere decir las entidades que están funcionando en el momento que la EPC se inicia y ver que funciona correctamente ejecutaremos el comando:

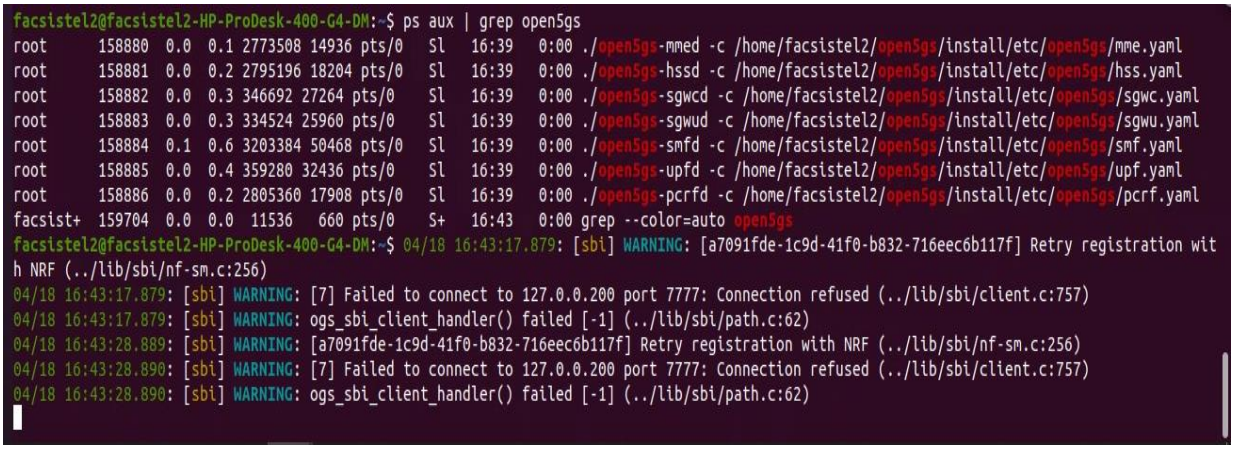


Figura 94 Procesos del EPC activos

Y por último con el comando:

```
grep -A 5 siap ~/open5gs/install/etc/open5gs/mme.yaml
```

Se podrán observar los procesos relacionados con la interfaz S1iap que es la encargada de conectar nuestra red con el eNodeB.

Figura 95 Información de S1ap

3.4.11. Instalación de las librerías BladeRF

Para instalar el equipo BladeRF micro 2.0 XA4, lo primero que se debe realizar es verificar si se dispone de un puerto USB 3.0 se utilizan los siguientes comandos:

```
sudo add-apt-repository ppa:bladerf/bladerf
sudo apt update
sudo apt install -y bladerf
```

Y este comando tendría que detectar el equipo BladeRF, y debería conectarnos directamente

```
bladeRF-cli i
```

Figura 96 Detección del equipo

A continuación, instalaremos la imagen FPGA del equipo BladeRF xA4, que la encontramos en la página del fabricante Nuand: https://www.nuand.com/fpga_images/

En BladeRF CLI usaremos los siguientes comandos para instalar la imagen FPGA y comprobaremos su versión

```
bladeRF> load fpga /home/facsistel/Descargas/hostedxA4-latest.rbf
bladeRF> version
bladeRF> info
```

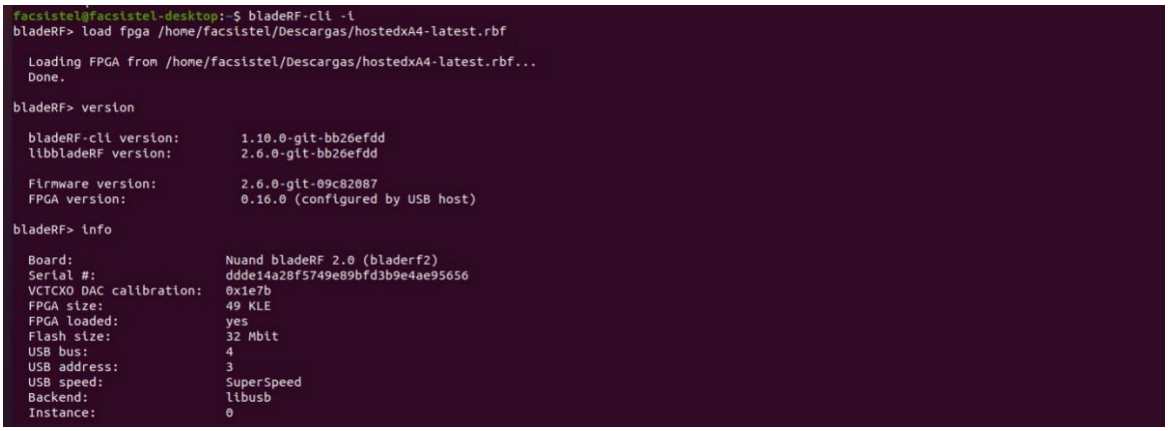


Figura 97 Información del dispositivo

También se utilizan otros comandos para observar la ganancia, frecuencia, frecuencia de muestreo y ancho de banda:

print gain print frequency print bandwitch print samplerate

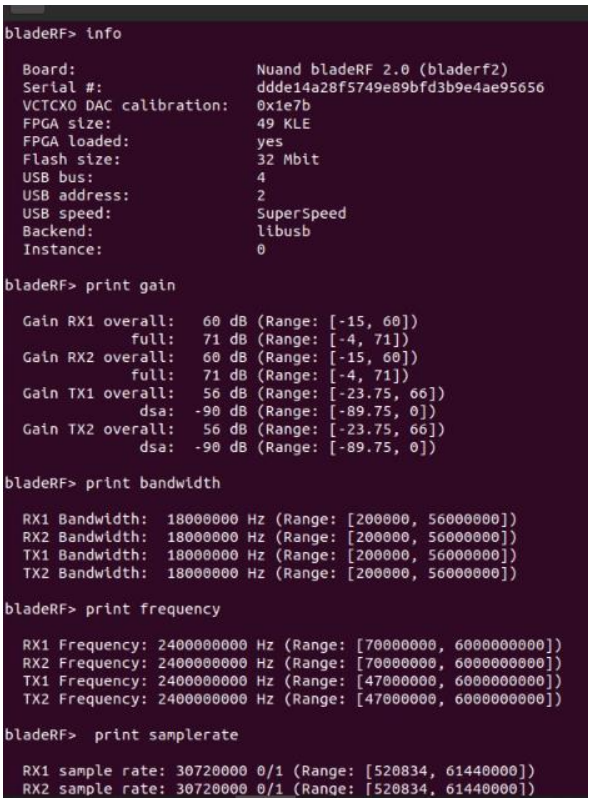


Figura 98 Parámetros del equipo

3.4.12. Construir eNodeB

Para empezar a construir la eNodeB en la Pc Ryzen, procederemos a actualizar los paquetes con el comando:

```
sudo apt update
```

Este comando buscara actualizaciones para la gestión de paquetes ubicados en los repositorios configurados.

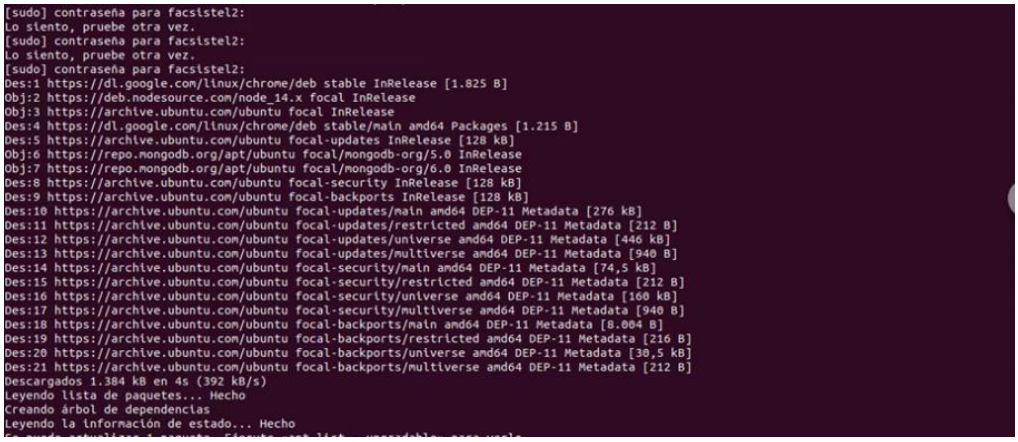


Figura 99 Actualización de paquetes

A continuación, instalaremos las librerías disponibles para actualizar con el siguiente comando:

```
sudo apt upgrade -y
```

Figura 100 Actualización de librerías disponibles

Una vez encontradas todas las actualizaciones disponibles, se procede a instalar los paquetes obtenidos de los repositorios, para almacenarlos de manera local.

Una vez instaladas todas las librerías y actualizaciones de paquetes, procederemos usar el siguiente comando:

```
sudo apt get install -y
```

Este comando nos servirá para la descarga e instalación de dependencias necesarias que requiere el software de código abierto srsRAN.

Figura 101 Instalación y actualización de dependencias de srsRAN

Ahora procederemos a la instalación de srsRAN_4G, que también es una interfaz de código abierto, esto porque la interfaz Open5gs solo es la implementación de una EPC, y no posee una interfaz para el eNodeB.

Estas dos interfaces son compatibles entre sí por lo que no habrá problemas al juntar la una con la otra.

```
cd ~
git clone https://github.com/srsRAN/srsRAN_4G.git
cd srsRAN_4G
```

El primer comando nos Redireccionará a la carpeta principal, se usa en caso de estar en un directorio interno.

El segundo comando git, nos hará una copia de los paquetes necesarios para el uso de srsRAN, ubicados en los repositorios de GitHub para el proyecto de código abierto srsRAN.

Figura 102 Clonar repositorio de GitHub

A continuación, crearemos la carpeta donde irán los archivos que clonamos de nuestro repositorio para cada una de las entidades, será una especie de repositorio local.

```
mkdir build
```

Figura 103 Creación de repositorio

Vamos a poner el comando cmake ../, este comando ayudara a generar el sistema de compilación de manera eficiente para compilar un proyecto de software como lo es SrsRAN.

```
cmake ../
```

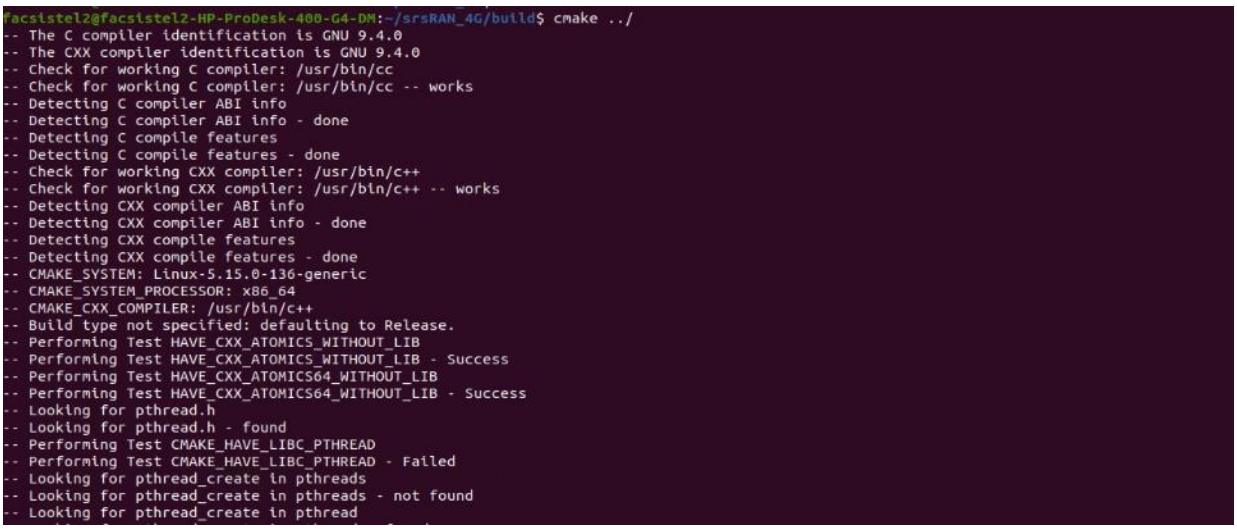


Figura 104 Sistema de compilación de srsRAN

Vamos a insertar el comando make, el cual ayudara a compilar el proyecto srsRAN_4G va a generar bibliotecas, archivos y diferentes componentes del proyecto.

```
make
```

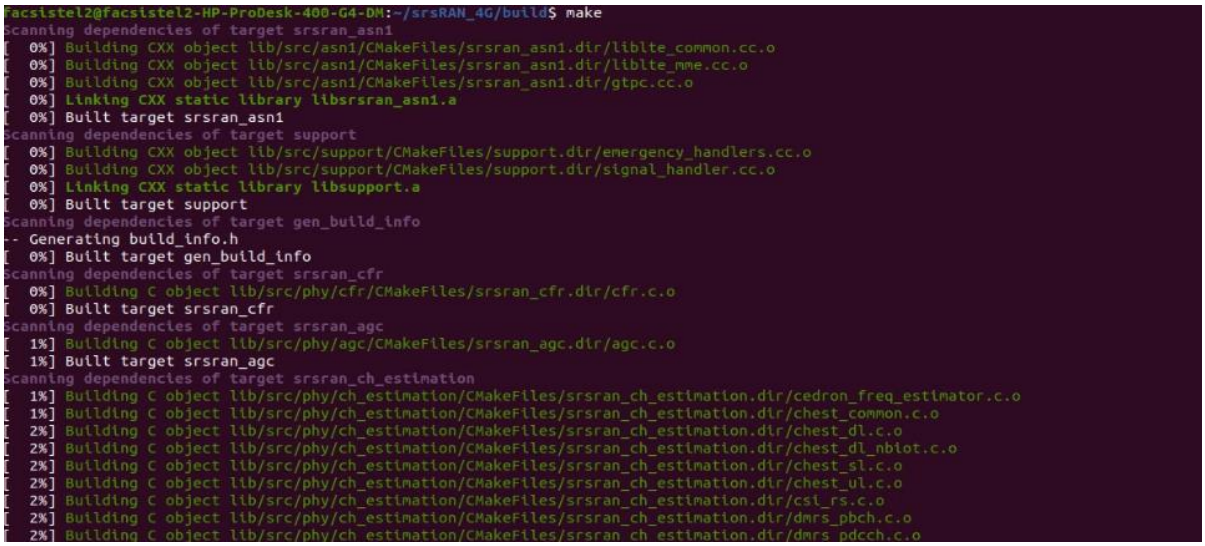


Figura 105 Compilación de srsRAN

Una vez que finalice llegara al 100% y terminara la construcción del proyecto.

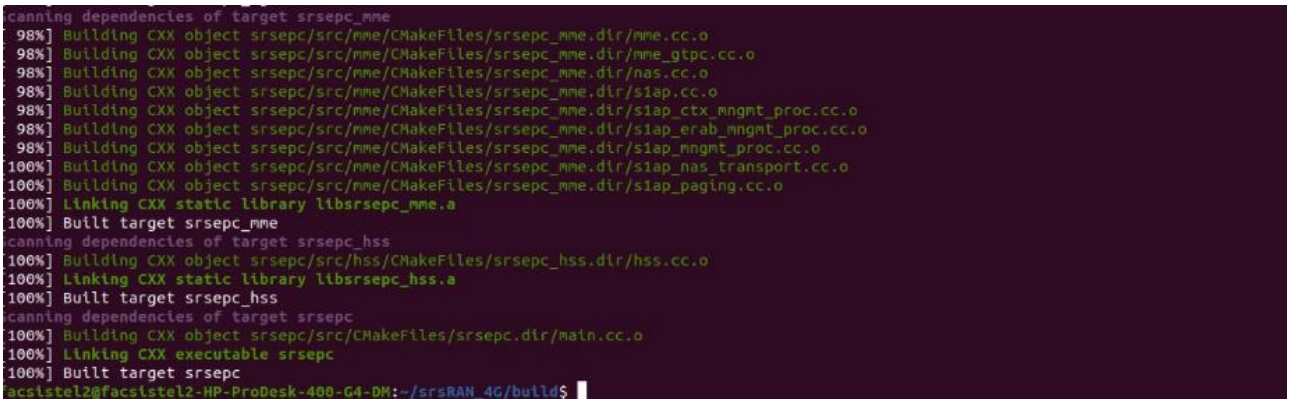


Figura 106 Compilación Finalizada

Ahora usaremos el comando. El comando instalara todos los componentes que se compilaron antes con make.

```
sudo make install
```

Ahora se instalarán, esto podría tardar varios minutos, dependiendo de la conexión a internet disponible.

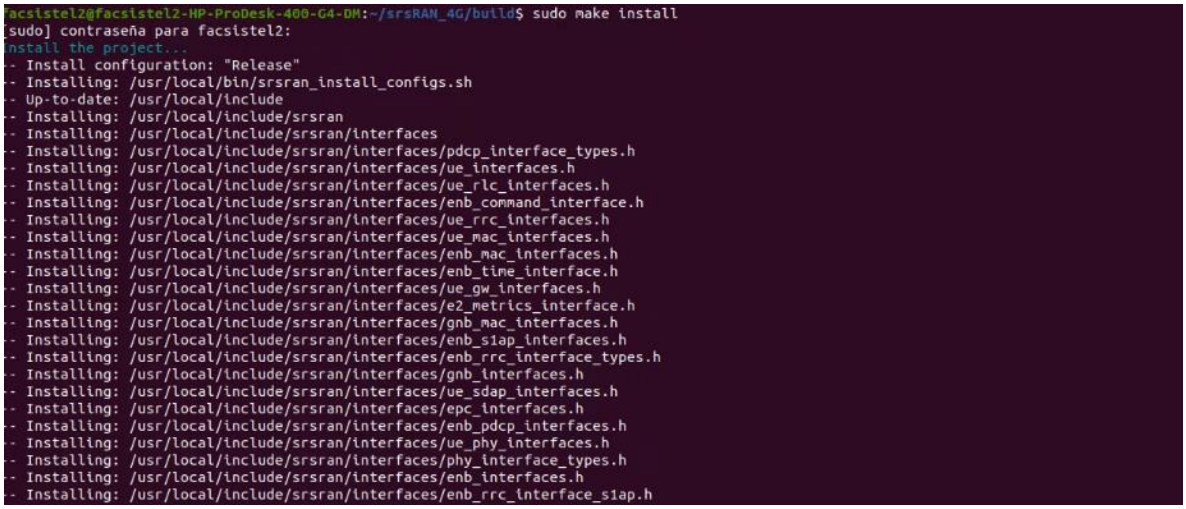


Figura 107 Instalación de Componentes compilados

Ahora vamos a crear una carpeta llamada srsran_configs, y aquí copiaremos los archivos de ejemplo que nos da el propio srsRAN_4G para la configuración del eNodeB.

```
mkdir ~/srsRAN_4G/srsran_configs
#Comando para copiar los archivos de ejemplo
cp ~/srsRAN_4G/srsenb/enb.conf.example ~/srsRAN_4G/srsran_configs/enb.conf
cp ~/srsRAN_4G/srsenb/sib.conf.example ~/srsRAN_4G/srsran_configs/sib.conf
cp ~/srsRAN_4G/srsenb/rr.conf.example ~/srsRAN_4G/srsran_configs/rr.conf
cp ~/srsRAN_4G/srsenb/rb.conf.example ~/srsRAN_4G/srsran_configs/rb.conf
```

Lo que hacen estos comandos, es copiar los paquetes descargados de los repositorios de GitHub para srsRAN, hacia nuestro proyecto en la carpeta srsran_configs, donde modificaremos el proyecto de acuerdo con nuestras necesidades.

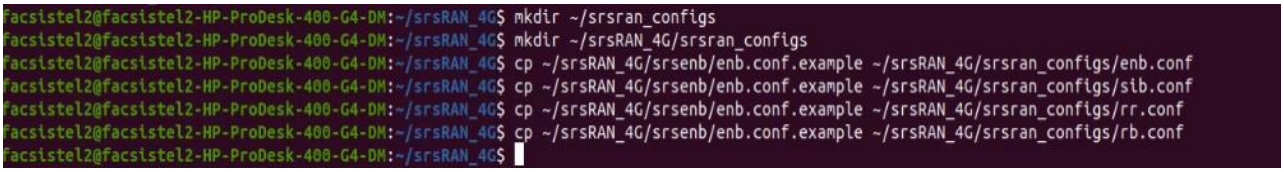


Figura 108 Copia de Archivos de ejemplo de los repositorios de srsRAN

Se puede apreciar los archivos fueron copiados correctamente y estamos listos para iniciar con la configuración del eNodeB.

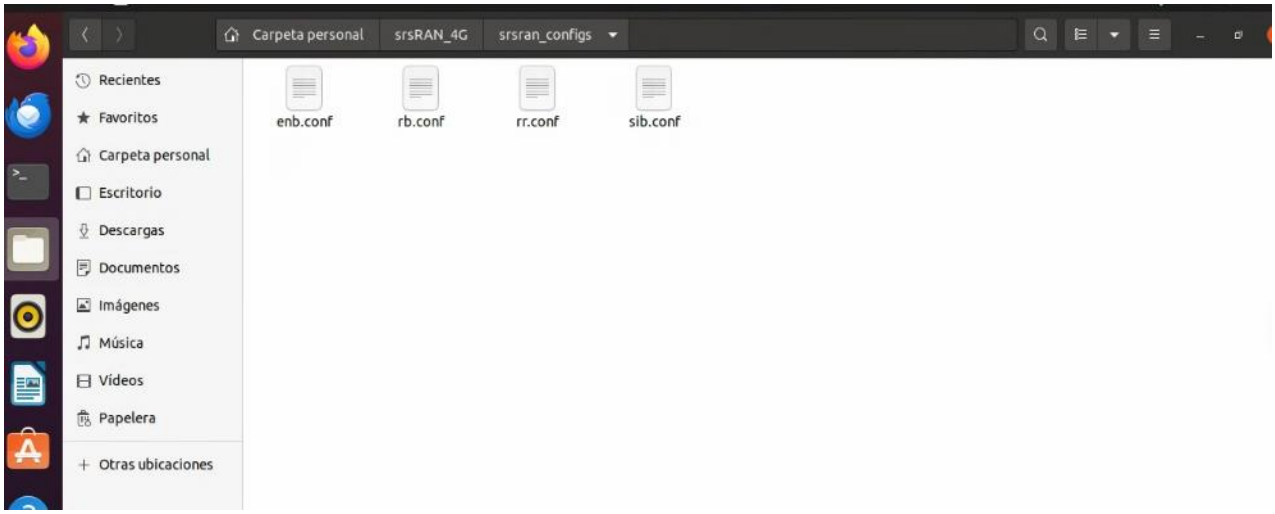


Figura 109 Directorio de srsran_config

A continuación, habilitaremos los servicios del grupo dialout y agregaremos nuestra computadora para obtener permisos especiales.

```
sudo adduser $USER dialout
```

```
facsistel2@facsistel2-HP-ProDesk-400-G4-DM:~/srsRAN_4G$ sudo adduser $USER dialout
Añadiendo al usuario `facsistel2' al grupo `dialout' ...
Añadiendo al usuario facsistel2 al grupo dialout
Hecho.
facsistel2@facsistel2-HP-ProDesk-400-G4-DM:~/srsRAN_4G$
```

Figura 110 Grupo Dialout

Una vez instalado todos los repositorios de srsRAN, mas las copias de los ejemplos del repositorio al directorio que creamos, se procederá a verificar la instalación y la versión de este con el siguiente comando:

```
~/srsRAN_4G/build/srsenb/src/srsenb --version
```

```
facsistel@facsistel-desktop:~$ ~/srsRAN_4G/build/srsenb/src/srsenb --version
Active RF plugins: libsrsran_rf_blade.so libsrsran_rf_zmq.so
Inactive RF plugins:
---  Software Radio Systems LTE eNodeB  ---

Version 23.4.0
```

Figura 111 Verificación de versión de srsRAN

O mediante el comando:

```
cd ~/srsRAN_4G
nano ~/srsRAN_4G/srsran_configs/enb.conf
```

```
facsistel2@facsistel2-HP-ProDesk-400-G4-DM:~/srsRAN_4G$ git describe --tags
release_23_11-6-gec29b0c1f
facsistel2@facsistel2-HP-ProDesk-400-G4-DM:~/srsRAN_4G$
```

Figura 112 Versión de srsRAN_4G mediante TAGS

3.4.13. Configurar eNodeB

Ahora una vez que terminamos la instalación del srsRAN_4G, vamos a configurar el enb.conf, primero tenemos que editar este archivo mediante el siguiente comando:

EL paquete enb.conf, contiene las configuraciones necesarias para poder simular nuestra eNodeB, pudiendo controlar parámetros como frecuencia, potencia, además de agregar la MCC y MNC, que nos ayudaran a identificar la red en los celulares por medio de las USIM que se configuraran en la MME en la EPC.

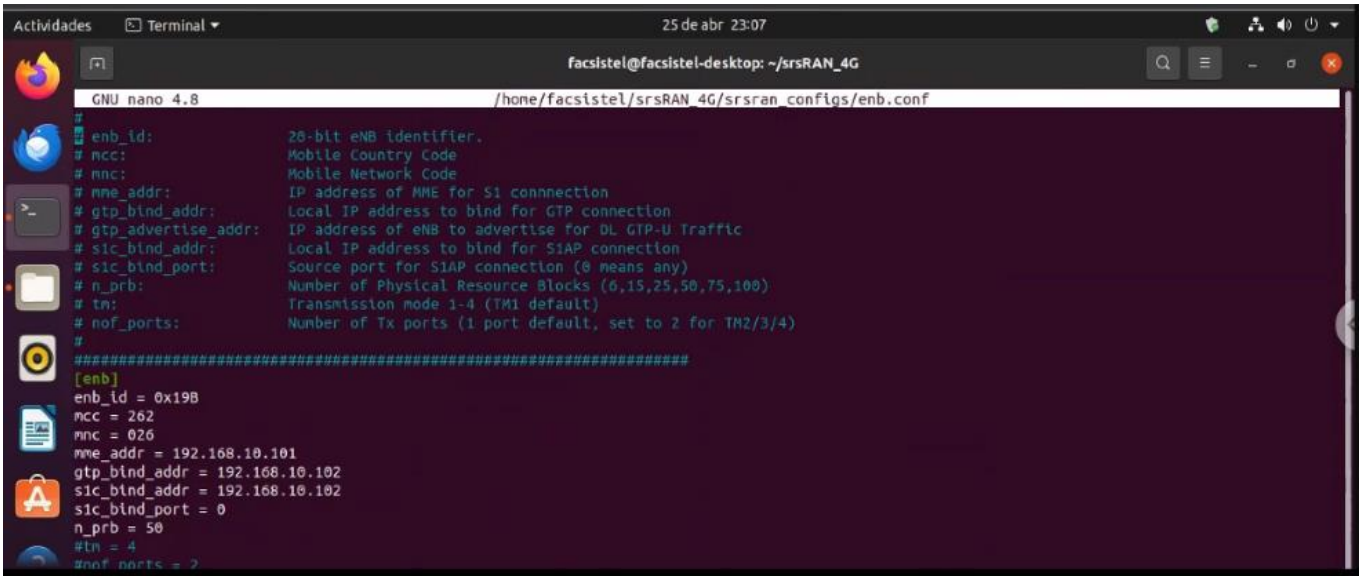


Figura 113 Script configurable del eNodeB de SrsRAN_4G

En la figura 113, se muestra la configuración del script de ejemplo que se copió, primero se configuran bloques el primero bloque aparece es [enb] el parámetro mme_addr se colocara a la misma

dirección IP que el de la otra PC que es la 192.168.10.102 y los parámetros GTP y s1c la dirección IP de este computador que es la 192.168.10.102.

Ahora [enb_files] se pondrá las direcciones de los archivos sib.conf, rr.conf, rb.conf que se configuran más adelante.

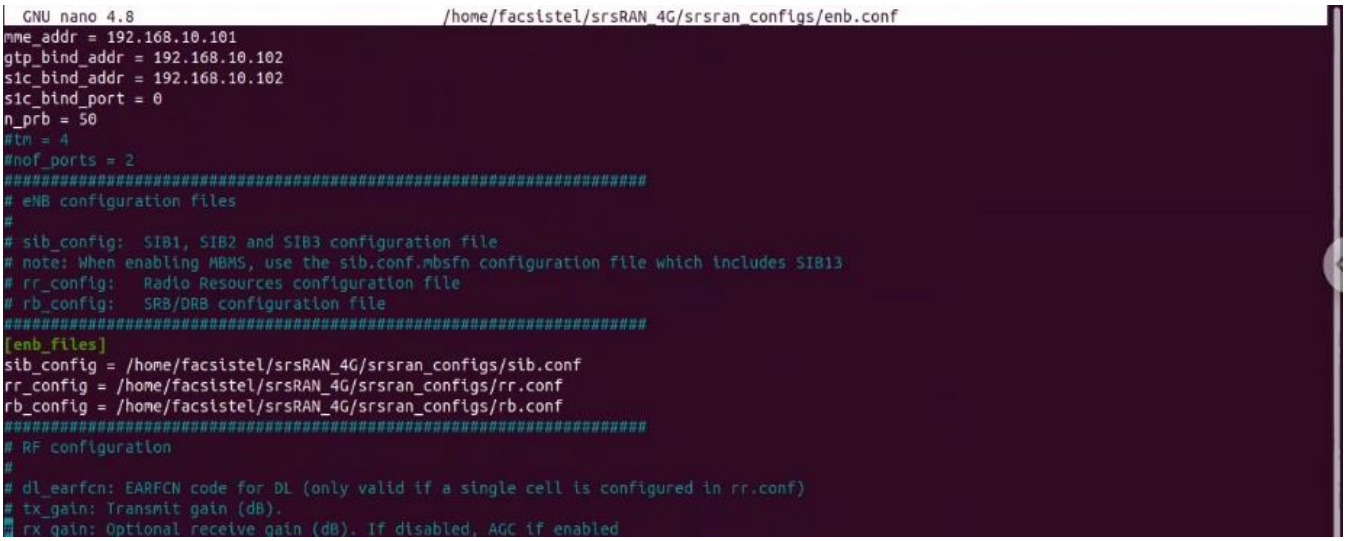


Figura 114 Direcciones de directorios sib.conf, rr.conf , rb.conf.

Como estamos una prueba de conexión con la EPC colocamos una eNodeB virtual se colocarán los siguientes parámetros en el aparto [rf] para que simule esta conexión:

```
device_name = zmq

device_args =
tx_port=tcp://*:2000,rx_port=tcp://localhost:2001,id=enb,base_srate=23.04e6

tx_gain = 50

rx_gain = 30
```

Aquí colocaremos los puertos y la potencia para nuestra transmisión, en esta ocasión se coloca zmq, debido a que usaremos una red simulada completa, simulando la conexión de un SDR, pero sin una transmisión real, esto para comprobar que todo se conecte correctamente.

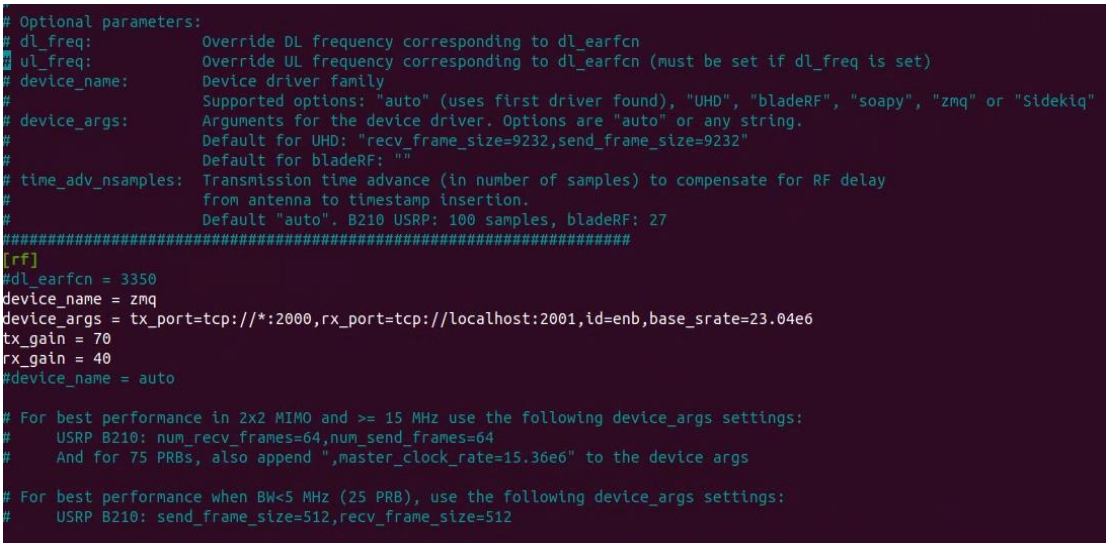


Figura 115 Configuración para simular el eNodeB

En caso de que sucedan errores se configura la sección de log, esta sección es la encargada de presentarnos en pantalla todo lo que ocurre al simular nuestra eNodeB, mostrándonos cambios en las celdas, además de potencias u otros datos, pero debido al alto tráfico de red, puede que la memoria termine llena, causando errores, por tal motivo se configura con warnings, es decir advertencias simples para liberar espacio en caso de ser necesario.

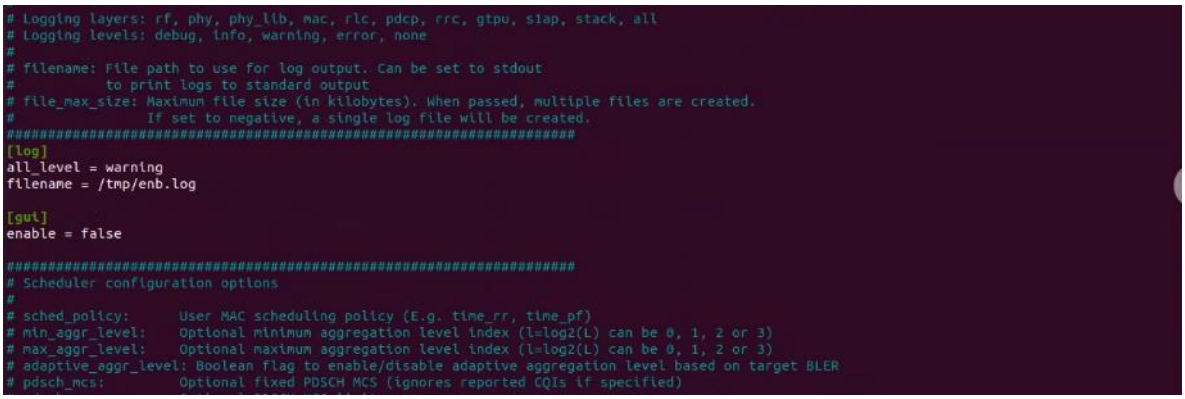


Figura 116 Configuración de Logs

Aquí en configuraciones adicionales, en la sección scheaduler, configuramos las políticas necesarias, para a través de time_pf, visualizar las frecuencias y picos en tiempos específicos, notando cambios en la red.

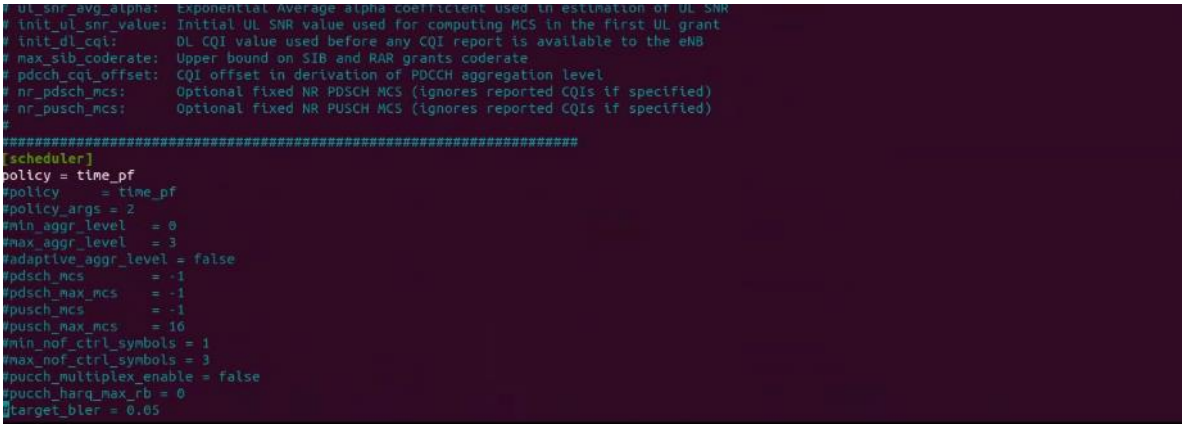


Figura 117 Configuración de los apartados del eNodeB

Con el siguiente comando, abriremos archivos como que debemos configurar, en este caso el sib.conf, donde revisaremos parámetros de la celda, frecuencias de UPLINK y DOWNLINK, permitiéndonos manejar la red.

```
nano ~/srsRAN_4G/srsran_configs/sib.conf
```

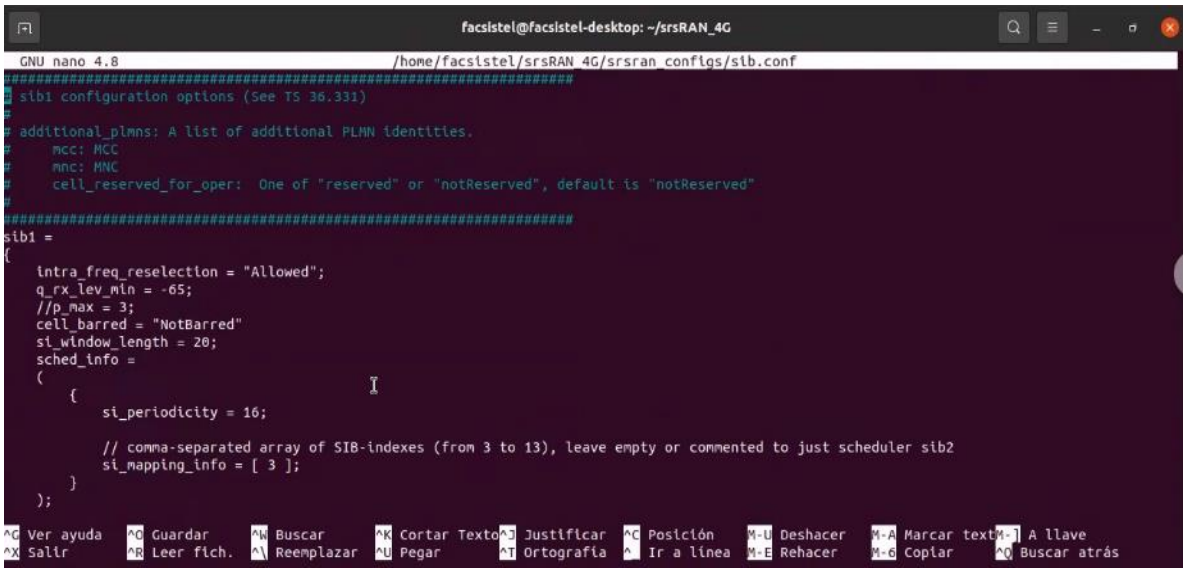


Figura 118 Configuración del script sib.conf

Los siguientes scripts son importantes ya que son indispensables para la ejecución correcta del eNodeB, primero accederemos a este script por defecto mediante la línea de código:

Y en este caso lo por defecto, se guarda la configuración con el comando ctrl + o y ctrl + x para cerrar. Repetiremos el mismo paso con el siguiente script:

```
nano ~/srsRAN_4G/srsran_configs/rr.conf
```

En esta configuración se mostraba un error en el tac =1, ya que esta línea se configuraba en el enb.conf, pero en versiones posteriores se configura en el rr.conf, esta línea se colocará en el apartado cell_list y el resto de las líneas de código se guardará por defecto.

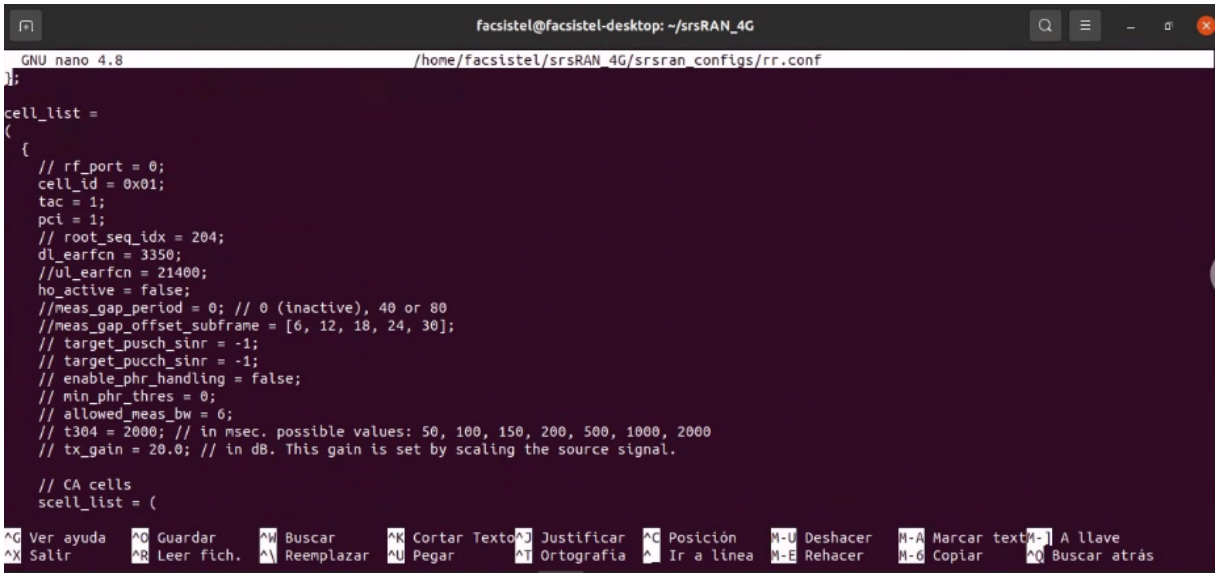


Figura 119 Configuración de Script rr.conf

Y por último vamos al último script, se guardará la configuración por defecto:

```
nano ~/srsRAN_4G/srsran_configs/rb.conf
```

Figura 120 Configuración del script rb.conf

Una vez configurados todas las entidades y scripts que se usaran para simular la red LTE, se realizara una prueba de conectividad para observar si se conectan correctamente, primero se editara en el archivo enb.conf, los siguientes parámetros:

mcc = 263 mnc = 026

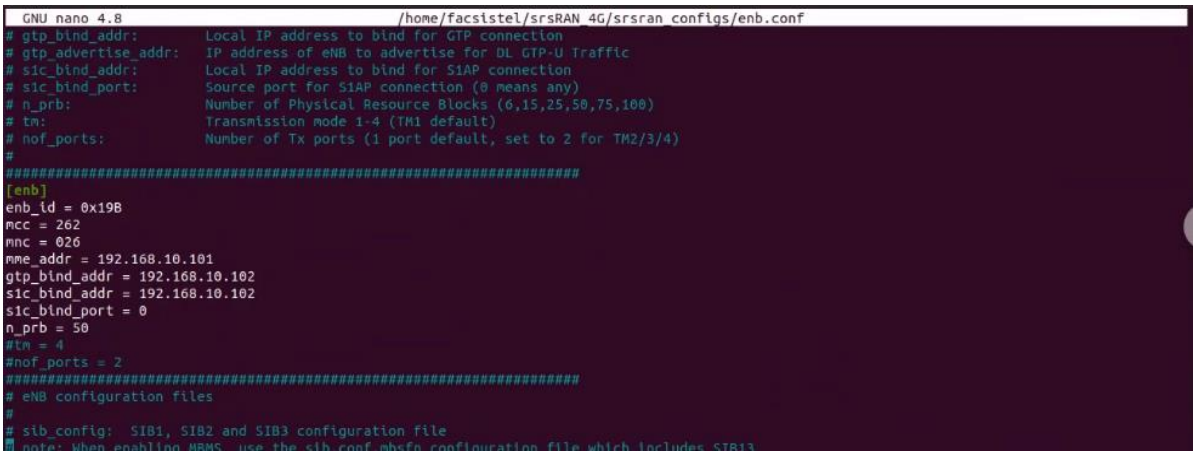


Figura 121 Configuración de MCC y MNC

También se editarán estos parámetros en el script de la entidad mme.yaml, los cuales deben ser iguales, al no ser iguales nos dará una inconsistencia y las interfaces S1-mme no se conectarán.

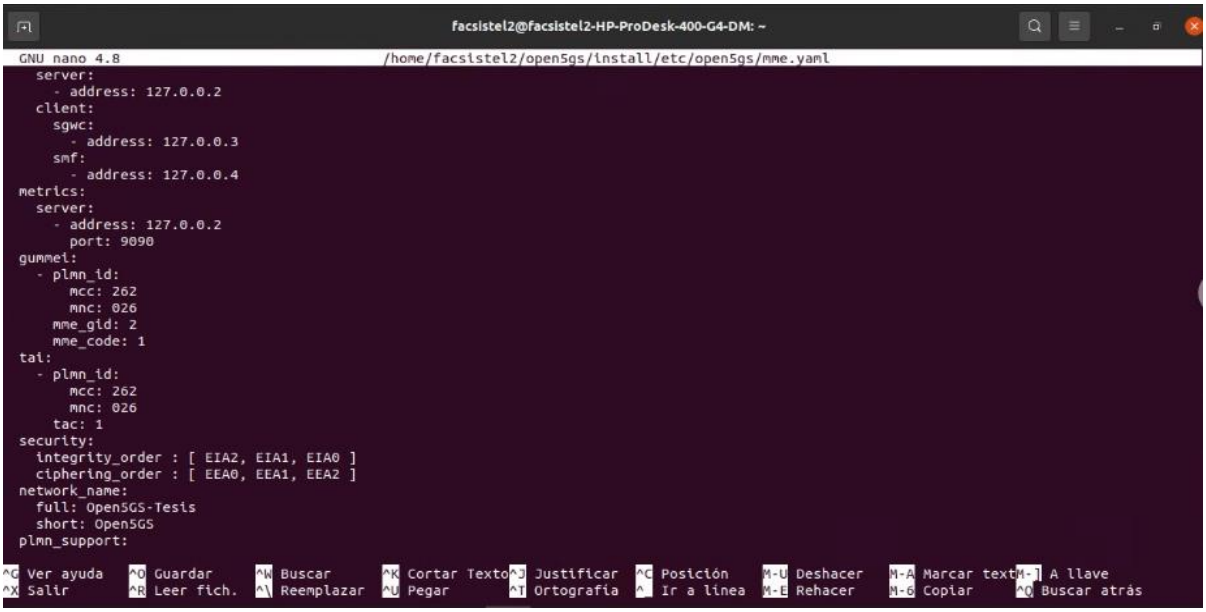


Figura 122 Configuración de la mme.yaml

3.4.14. Arrancar eNodeB

Primero se levantará la EPC ya configurada anteriormente se iniciará con el comando:

Mediante este comando vamos a iniciar eNodeB:

```
sudo ~/start-epc.sh
```

Este comando es esencial ya que abrirá los puertos necesarios como S1AP, GTP para establecer la conexión, cuando eNodeB se inicie nos dirá algo como:

```
sudo ~/srsRAN_4G/build/srsenb/src/srsenb ~/srsRAN_4G/srsran_configs/enb.conf
```

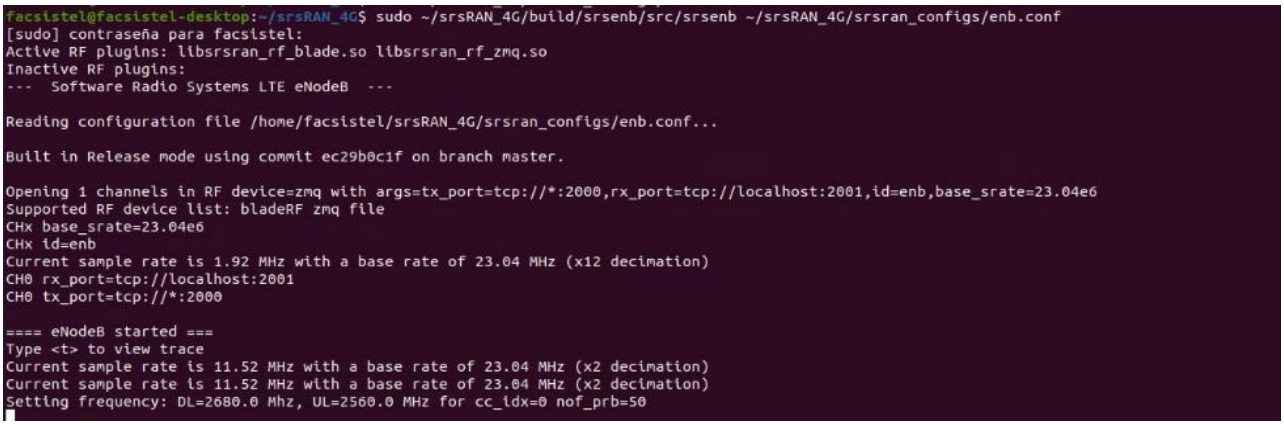


Figura 123 eNodeB iniciado correctamente

Cuando eNodeB se inicie en la EPC, aparecerá de confirmación de la conexión entre los dos componentes de la red:

```
Enb-s1 accepted
Number of eNBS is now 1
```

Figura 124 eNodeB conectada correctamente a la EPC

En la figura 124, la EPC y la eNodeB están completamente conectadas y configuradas entre sí.

Cuando se detiene un servicio aparecerá un mensaje:

```
Connection refused
Failed to initlate S1 connection
```

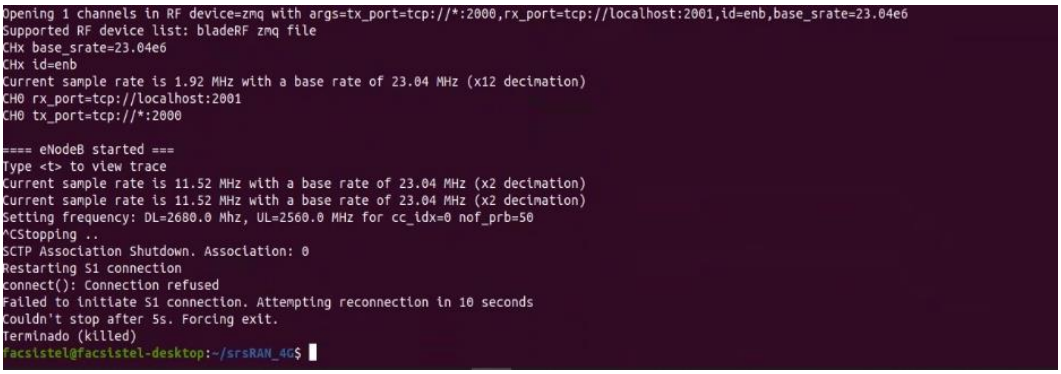


Figura 125 eNodeB sin conexión

En la figura 125, detuvimos el EPC, y por eso se perdió la conexión del eNodeB, cuando el EPC vuelva a estar activo se volverá a habilitar el eNodeB.

3.4.15. Programación de USIM

Para realizar la configuración de las USIM, se realizará mediante la base de datos Mongo DB directamente, se podría realizar mediante la interfaz gráfica que no proporciona Open5GS webIu, pero al momento de querer instalar esta extensión nos dio un error y no se pudo corregir como alternativa se hace directo en la terminal con la base de datos que ya ha sido instalada correctamente.

Primero se verificará el estado de Mongo DB y luego se entrará a la base de datos:

```
Sudo systemctl status mongod
Mongosh
Use open5gs
```

Si Mongo DB esta activo, luego insertaremos el comando Mongosh para acceder a la base de datos, luego se coloca use open5gs para cambiar a la base de datos de open5gs.

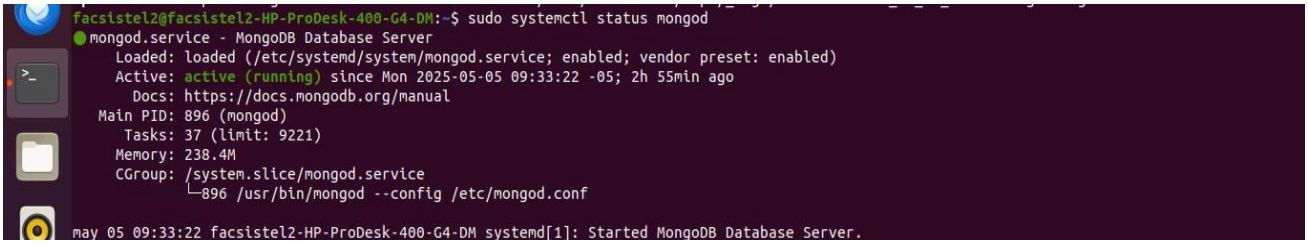


Figura 126 Mongo DB activo

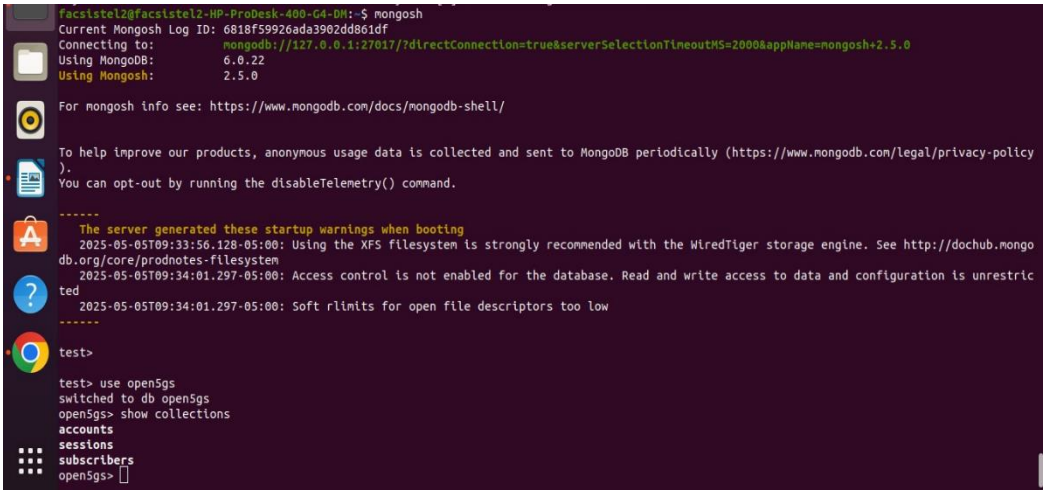


Figura 127 Acceso a la base de datos de Mongo DB

Como se puede observar en la figura 126, Mongo DB este activo entonces se procederá a insertar los comandos de la figura 127. Para la configuración de las USIM LTE se utilizó la base de datos MongoDB, y se simulo dos USIM.

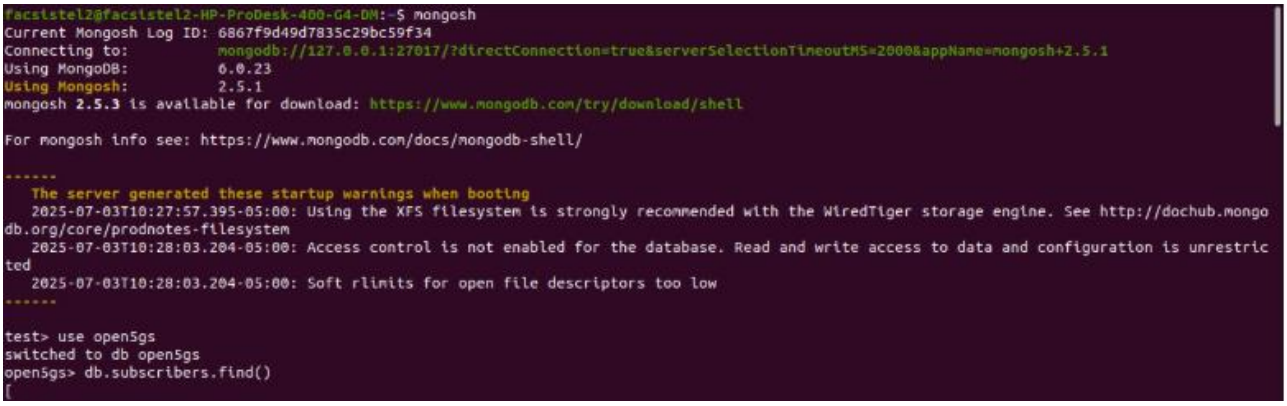


Figura 128 Ingreso a base de datos MongoDB

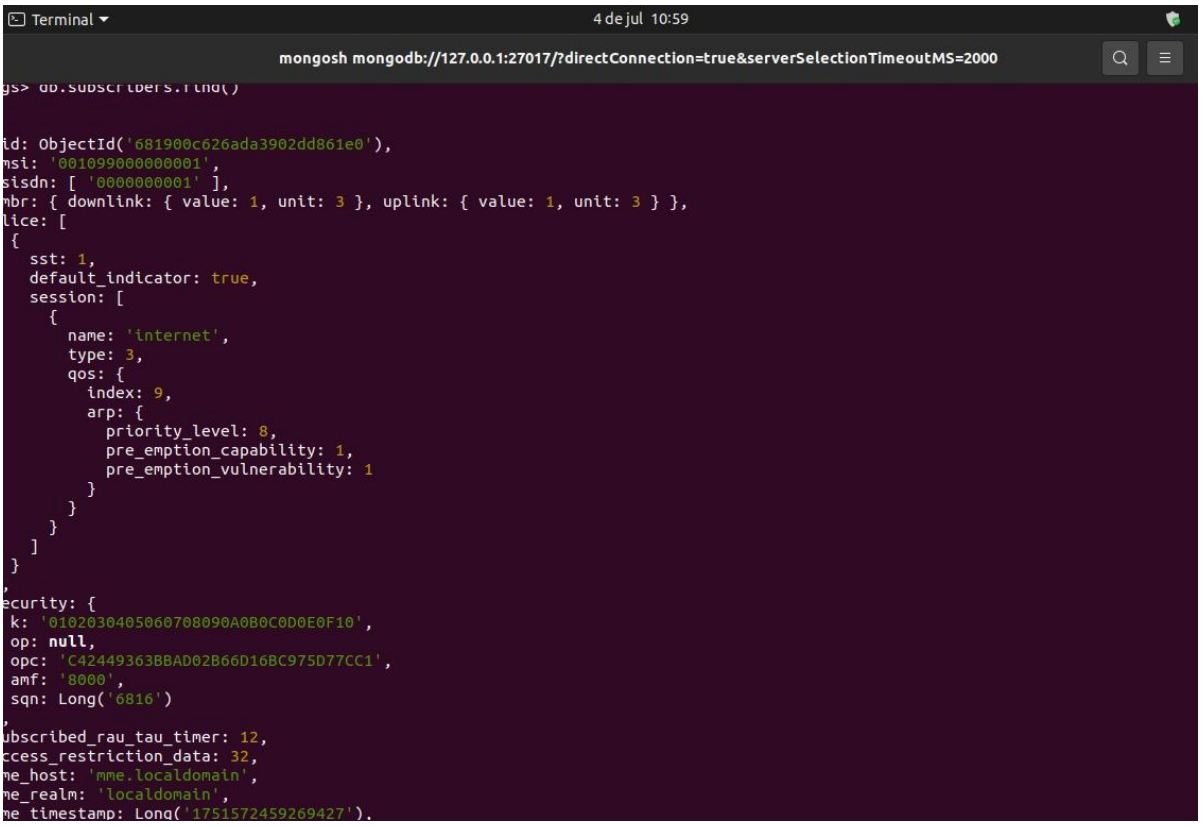


Figura 129 Configuración de la USim 1

Y para la configuración en LTE se configuran los siguientes parámetros:

IMSI15: 001099000000001
KI: 0102030405060708090A0B0C0D0E0F10
OPc: C42449363BBAD02B66D16BC975D77CC1

IMSI15: Identificador único del suscriptor móvil, de 15 dígitos (International Mobile Subscriber Indentity).

KI: Clave secreta única de la SIM usada para autenticación en la red móvil.

OPc: Clave derivada usada en la autenticación mutua entre la red y la SIM en redes 3G/4G.

Para la segunda USIM LTE se van a configurar con parámetros similares, pero variando ya se los últimos dígitos, pero no pueden ser iguales, entonces esta configuración es la siguiente:

ICCID: 8901099000000000002

IMSI15: 001099000000002

KI: 0102030405060708090A0B0C0D0E0F11

OPc: C42449363BBAD02B66D16BC975D77CC2

ACC: 0004

#Parametros iguales

PIN1/PIN2: 2318

ADM: 42370400000000

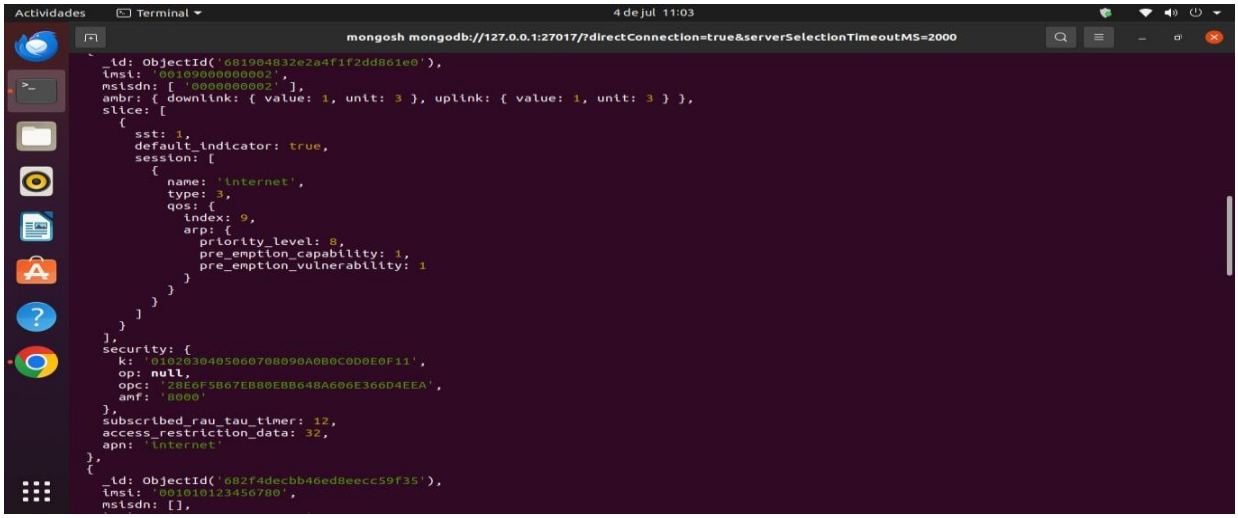


Figura 130 Configuración Tarjeta SIM 2

3.5. Costos de la propuesta

La propuesta se basó en datos de los equipos utilizados para realizar estos trabajos de investigación, que se mostraran a continuación:

Cantidad	Descripción	P. Unitario	P. Total
1	BladeRF micro2.0 XA4	$540	$540
2	Antena Tribanda	$25	$50
1	Otros	$170	$170
TOTAL			$760

Tabla 12 Costes de proyecto

Capítulo IV

4.1. Pruebas

4.1.1. Arquitectura Física de la red Simulada

Se implemento la arquitectura LTE simulada en base a la 3GPP y el equipo SDR como estación base

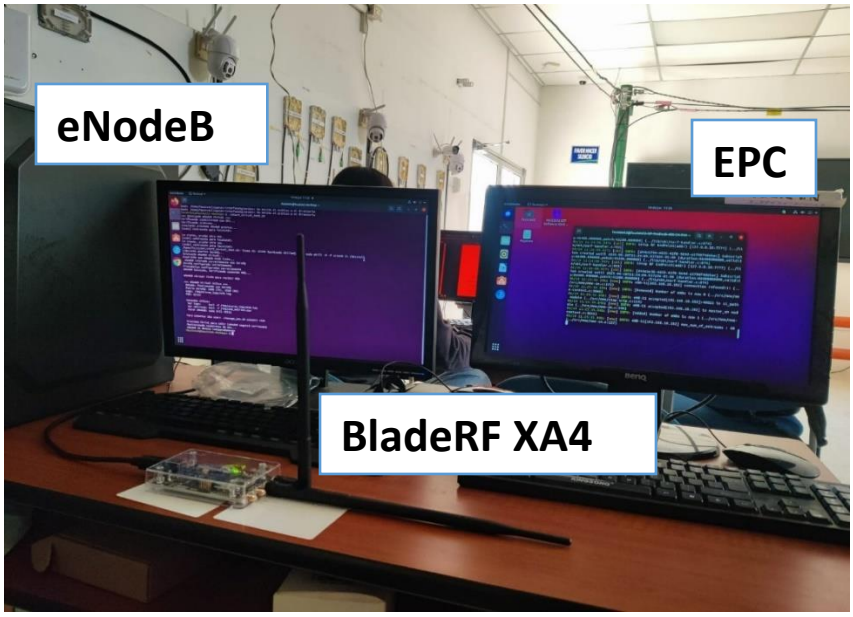


Figura 131 Arquitectura Física

4.1.2. Prueba de conexión del BladeRF como estación base

Se configuro el BladeRF xA4 como estación base para conectarse a la EPC virtual mediante el comando:

```
sudo ~/srsRAN_4G/build/srsenb/src/srsenb ~/srsRAN_4G/srsran_configs/enb.conf
```

```
facsistel@facsistel-desktop:~$ sudo ~/srsRAN_4G/build/srsenb/src/srsenb ~/srsRAN_4G/srsran_configs/enb.conf
Active RF plugins: libsrsran_rf_uhd.so libsrsran_rf_soapy.so libsrsran_rf_blade.so libsrsran_rf_zmq.so
Inactive RF plugins:
---  Software Radio Systems LTE eNodeB  ---

Reading configuration file /home/facsistel/srsRAN_4G/srsran_configs/enb.conf...

Built in Release mode using commit fa56836b1 on branch HEAD.

/home/facsistel/srsRAN_4G/srsenb/src/enb_cfg_parser.cc:1881: Force DL EARFCN for cell PCI=1 to 9410
Opening 1 channels in RF device=bladeRF with args=default
Supported RF device list: UHD soapy bladeRF zmq file
Opening bladeRF...
```

Figura 132 Prueba de conexión del BladeRF y EPC

Se puede apreciar que una vez que conectamos el BladeRF XA4 como estación base a la eNodeB, se va a conectar con la EPC y podremos verificar las frecuencias asignadas en las bandas correspondientes, en los apartados DL= 778.0 MHz que sería la banda 28 APT de CNT en 700 MHz

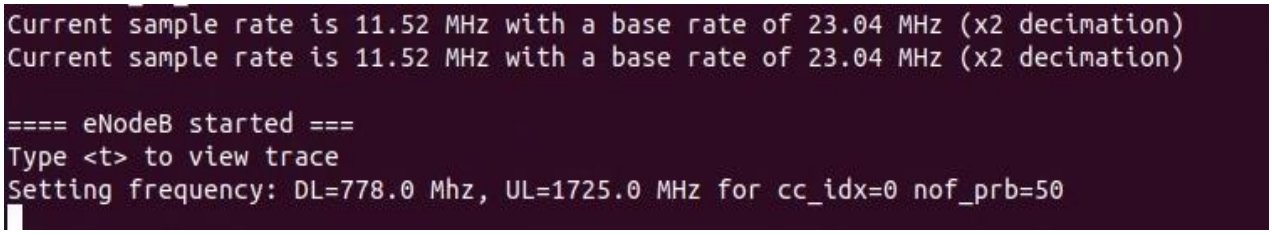


Figura 133 Prueba de BladeRF banda 28

Una vez iniciada el eNodeB con el BladeRF xA4 podemos verificar en los logs de la EPC que ha sido detectado y conectado con éxito.

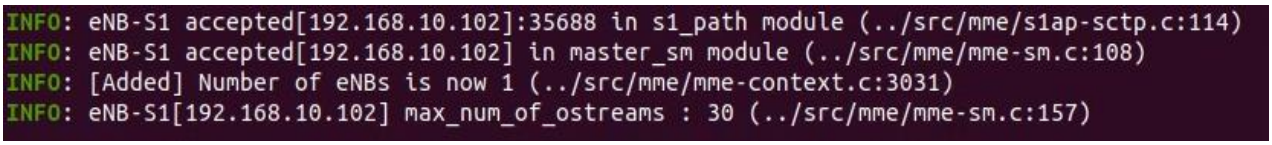


Figura 134 Conexión del BladeRF con EPC

Se puede verificar el funcionamiento del equipo SDR observando que los leds comienzan a parpadear continuamente dándonos a entender que el BladeRF está transmitiendo.

La señal de radio que transmitió el BladeRF xa4, y que se observó por el equipo Narda SRM-3600 de la Universidad en la banda 28 de CNT en la frecuencia de 778 MHz que es lo que se configuro en el BladeRF y es mostrado en los logs de la eNodeB.

Nos indica que la señal del campo eléctrico es igual a 313.7 mV/m, que equivale a -5.83 dBm, esto quiere decir que la potencia de la antena Tri Band no está generando la suficiente ganancia para ser detectado por un equipo de usuario comercial.

Esto se comprobó por medio del equipo VNA en una prueba realizada anteriormente la cual podemos observar en el [Anexo 4] y se verifico que las antenas que vinieron de fabrica con el equipo SDR no poseen la suficiente potencia al solo ser antenas con ganancia con 5 dBi, esta también fue la razón por la que se decidió medir los parámetros de radiofrecuencia con el equipo Narda SRM-3600.

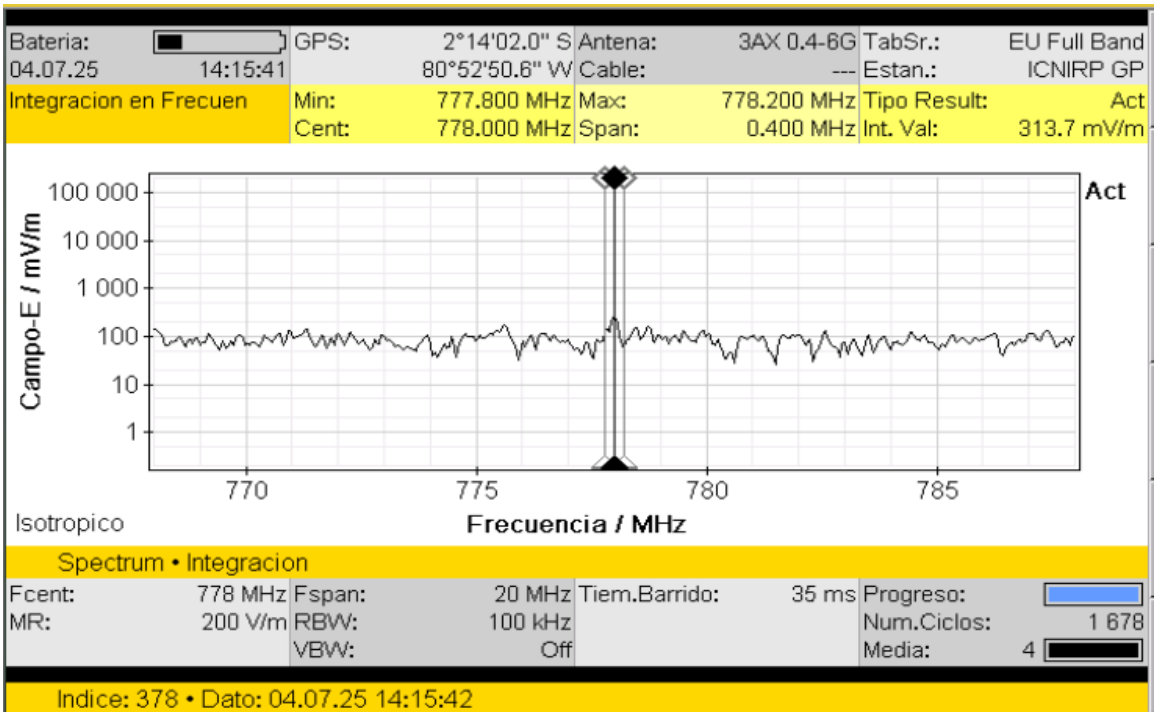


Figura 135 Medición de la frecuencia de 778 MHz

De la misma manera se inició el BladeRF en la banda 4 AWS que es la que utiliza Movistar y Claro.

```
==== eNodeB started ===
Type <t> to view trace
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Setting frequency: DL=2125.0 Mhz, UL=1725.0 MHz for cc_idx=0 nof_prb=50
```

Figura 136 Prueba BladeRF en AWS

Al revisar esta banda de frecuencia en le NARDA, se observó una ligera mejora en la señal mostrándonos un campo eléctrico 223.3 mV/m esto equivale a -8.3 dBm, lo cual es lo máximo que se pudo llegar con el equipo. Pero de todas maneras no fue suficiente para ser detectado por un equipo de usuario real.

Como es una banda licenciada a veces se puede captar por error la frecuencia de la banda comercial de Claro, es por eso por lo que se debe tener cuidado al momento de configurar el equipo NARDA para no equivocarnos en las mediciones.

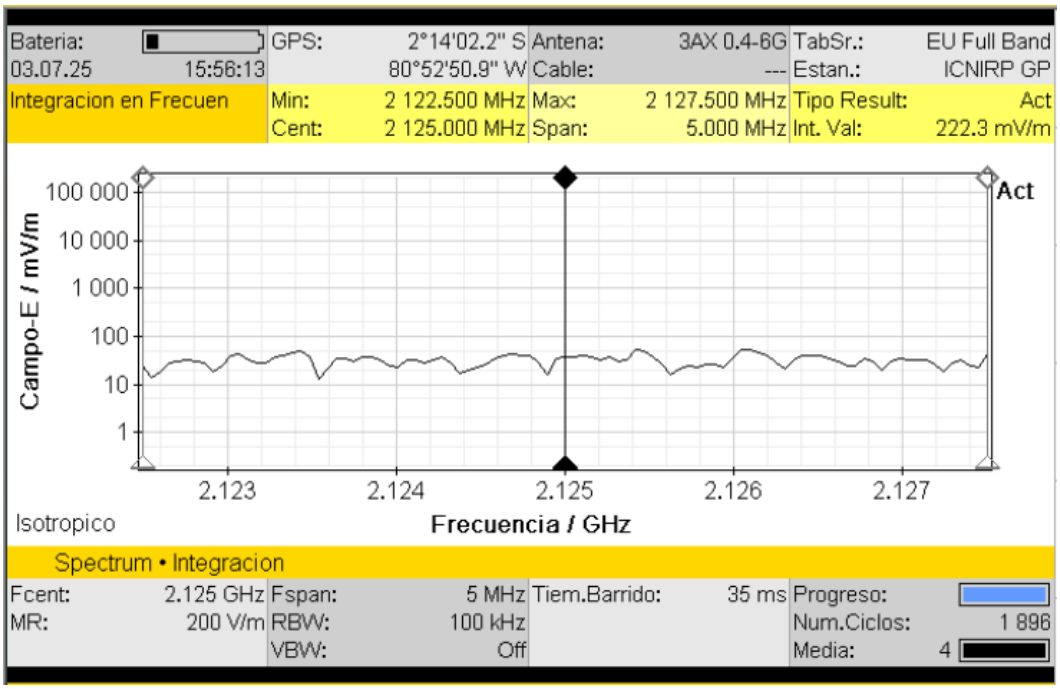


Figura 137 Medición en la banda AWS

En la siguiente tabla se aprecia los valores de la ganancia obtenida por medio del NARDA en parámetros de dBm para las diferentes bandas probadas

Banda de frecuencia	Ganancia (dBm)
CNT 700 MHz	-5.83 dBm
Claro / Movistar 1700/2100 MHz	-8.83 dBm

Tabla 13 Resultados de las mediciones del Narda

La conexión del UE con la eNodeB y EPC, se realiza de manera virtual con ayuda de un SDR virtual a través de las librerías zmq y un UE virtual simulado por srsUE que forma parte del software de código abierto srsRAN_4G.

4.1.3. Captura de trafico de la conexión (Wireshark)

Se utilizo el software Wireshark para observar la conexión de la eNodeB y EPC mediante la interfaz S1, que es la encargada de comunicar estos dos componentes de la arquitectura LTE.

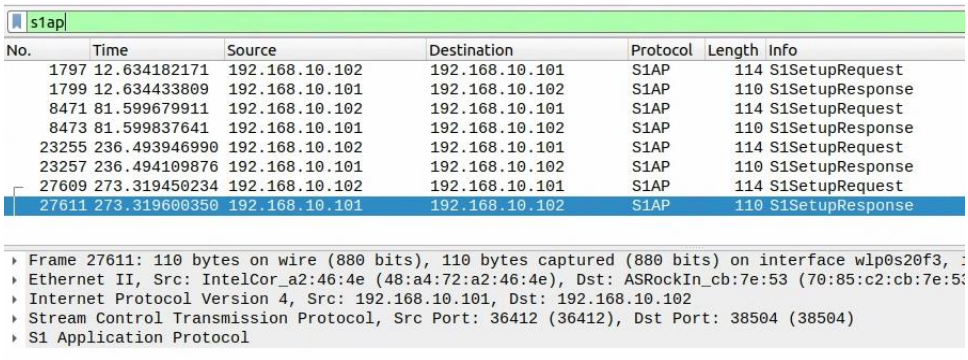


Figura 138 Comunicación entre EPC y la eNodeB

En la Figura 138, se puede observar que se intercambian mensaje S1SetupRequest que se describe como el mensaje inicial que el eNodeB envía a la entidad MME para establecer la conexión mediante la interfaz S1 y a continuación se devuelve un mensaje S1SetupResponse que es la respuesta del MME al eNodeB para completar la configuración.

4.1.4. Equipo de usuario virtual

Para la siguiente prueba se realizó la configuración de un equipo de usuario virtual para conectarnos a la red LTE Simulada, es por ello por lo que primero se creó un script en la carpeta de configuración donde instalamos nuestro eNodeB que en este caso sería la computadora Ryzen 7 y la carpeta de configuración srsran_configs y llamaremos a este archivo ue_virtual.conf. Y con el primer usuario que creamos anteriormente se va a configurar:

Los códigos de IMSI, k, OPC, IMEI tienen que ser idénticos a los que se tenía registrados en la base de datos para poder conectarse a la EPC. El archivo de configuración queda de la siguiente manera:

```
[rf]
device_name = zmq
device_args = tx_port=tcp://*:2001,rx_port=tcp://localhost:2000,id=ue,base_srate=23.04e6
[rat.eutra]
dl_earfcn = 3400
[usim]
mode = soft
algo = milenage
opc = C42449363BBAD02B66D16BC975D77CC1
k = 0102030405060708090A0B0C0D0E0F10
imsi = 001099000000001
imei = 353490069873319
[nas]
apn = internet
apn_protocol = ipv4
[log]
all_level = info
filename = /tmp/ue_virtual.log
```

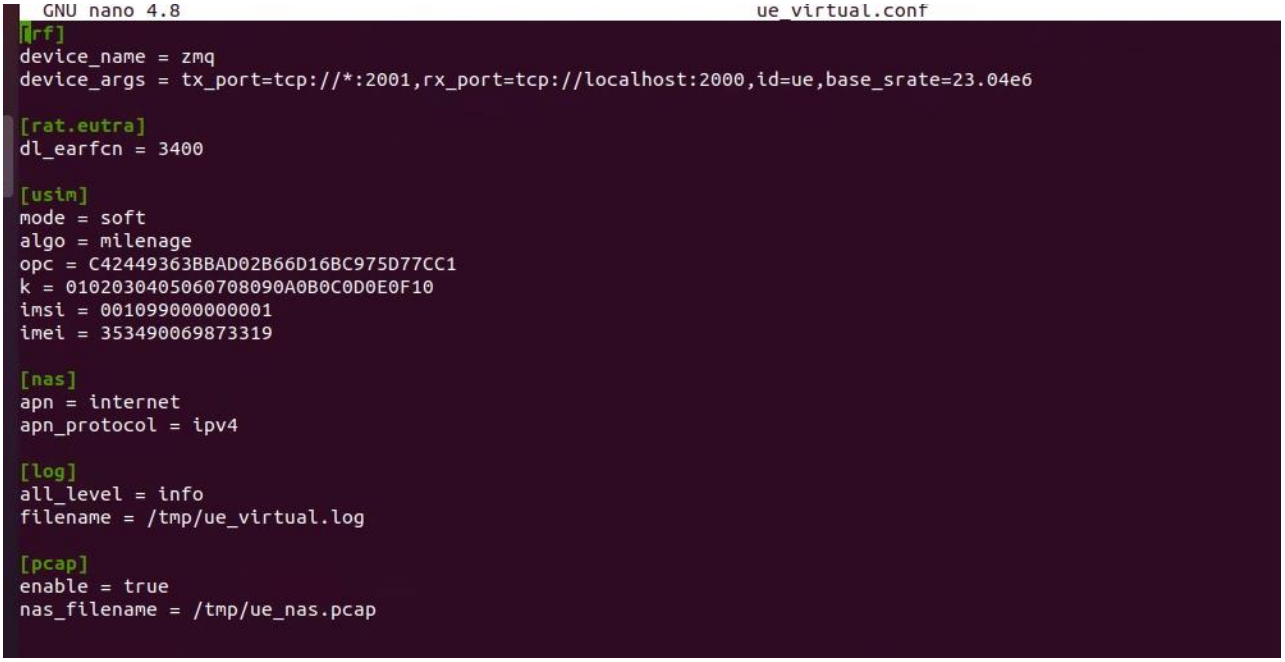


Figura 139 Interfaz Tun

En la máquina del EPC se configura la Interfaz TUN (ogstun) que en términos simples se trata de ser el puente de establecer la comunicación entre la red LTE virtual y el internet, si no se configura esta interfaz no abra un camino de comunicación. Esta configuración se realiza con los siguientes comandos:

 # En Intel i5 (EPC)

sudo ip tuntap add name ogstun mode tun

sudo ip addr add 10.45.0.1/16 dev ogstun

sudo ip link set ogstun up

 # NAT para acceso al Internet

sudo iptables -t nat -A POSTROUTING -s 10.45.0.0/16 ! -d 10.45.0.0/16 -j MASQUERADE

sudo iptables -A FORWARD -i ogstun -o [INTERFAZ_INTERNET] -j ACCEPT

sudo iptables -A FORWARD -i [INTERFAZ_INTERNET] -o ogstun -m state --state RELATED,ESTABLISHED -j ACCEPT

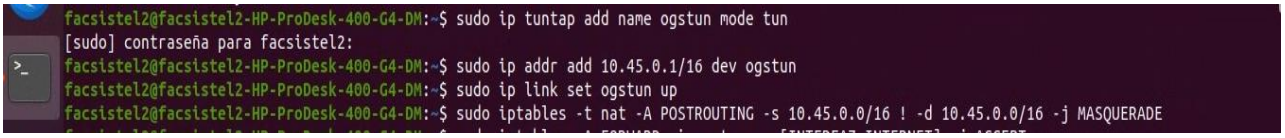


Figura 140 Activar la interfaz de internet

En este aparado se debe cambiar el nombre real de la interfaz de internet que es el caso de la Intel i5 es epn1s0, la configuración queda de la siguiente manera:

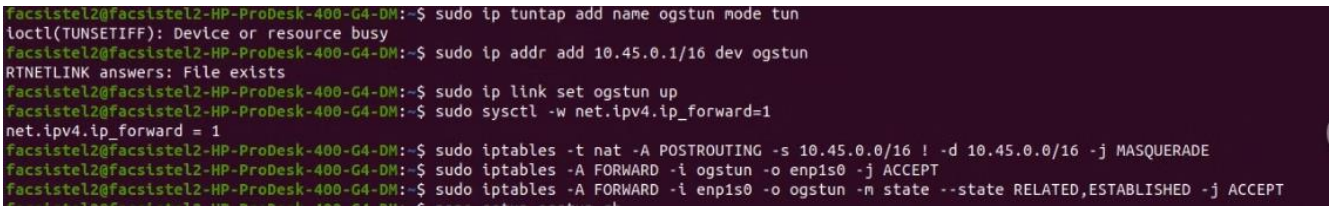


Figura 141 Interfaz configurada

Se creo un archivo para iniciar el eNodeB y los diferentes equipos de usuarios que configuremos se conectaran al EPC, este archivo se llama start_virtual_test.conf

Al momento de crear este script se debe tener cuidado con las rutas, al momento de ejecutar el script deberá encontrar todos los archivos de configuración que serán ejecutados, y si no los encuentre mostrara errores de ejecución o una advertencia mostrando que no existe el directorio.

```
#!/bin/bash
echo "=== Iniciando Prueba UE Virtual ==="
# IP real de tu Intel i5 (cambia por la IP correcta)
EPC_IP="192.168.10.101"
# Rutas correctas para srsRAN (usando tu comando exacto)
SRSENB_CMD="/home/facsistel/srsRAN_4G/build/srsenb/src/srsenb"
SRSUE_CMD="/home/facsistel/srsRAN_4G/build/srsue/src/srsue"
# Rutas de configuración (usando tu estructura)
ENB_CONFIG="/home/facsistel/srsRAN_4G/srsran_configs/enb.conf"  # Nota: .conf no .con
UE_CONFIG="/home/facsistel/srsRAN_4G/srsran_configs/ue_virtual.conf"
```

```
# Verificar conectividad con EPC
echo "Verificando conectividad con EPC..."
if ! ping -c 2 $EPC_IP > /dev/null 2>&1; then
    echo "WARNING: No se puede alcanzar el EPC en $EPC_IP"
    echo "Verifica que el EPC esté corriendo en la Intel i5"
fi
# Verificar que los archivos existen
echo "Verificando archivos..."
if [ ! -f "$SRSENB_CMD" ]; then
    echo "ERROR: No se encuentra $SRSENB_CMD"
    exit 1
fi
if [ ! -f "$SRSUE_CMD" ]; then
    echo "ERROR: No se encuentra $SRSUE_CMD"
    exit 1
fi
if [ ! -f "$ENB_CONFIG" ]; then
    echo "ERROR: No se encuentra $ENB_CONFIG"
    echo "¿El archivo se llama enb.con o enb.conf?"
    ls -la /home/facsistel/srsRAN_4G/srsran_configs/enb.*
    exit 1
fi
if [ ! -f "$UE_CONFIG" ]; then
    echo "ERROR: No se encuentra $UE_CONFIG"
    exit 1
fi
```

En estas líneas va a verificar los archivos de configuración y va a verificar que el EPC este corriendo correctamente y verificara que todo esté en orden. Y por último estarán los logs que serán de errores en caso de los tengamos y se ejecutara el UE virtual y eNodeB

```
# Limpiar procesos srsRAN previos
echo "Limpiando procesos srsRAN previos..."
sudo pkill srsenb 2>/dev/null || true
sudo pkill srsue 2>/dev/null || true
sleep 2
# Crear directorio de logs
mkdir -p /tmp/srsran_logs
# IniciareNodeB en background (usando tu comando exacto)
echo "IniciandoeNodeB virtual..."
sudo $SRSENB_CMD $ENB_CONFIG > /tmp/srsran_logs/enb.log 2>&1 &
ENB_PID=$!
# Esperar queeNodeB esté listo
echo "Esperando queeNodeB esté listo..."
sleep 8
# Verificar queeNodeB está corriendo
if ! kill -0 $ENB_PID 2>/dev/null; then
    echo "ERROR:eNodeB falló al iniciar"
    echo "Ver logs:"
    echo "============================"
    cat /tmp/srsran_logs/enb.log
    echo "============================"
    exit 1

fi
```

Aparecerán mensajes de errores cuando no se haya podido iniciar eNodeB de manera correcta, en algunos casos se quedan activos procesos y se tendrán que finalizar con el comando Kill

```
sudo kill 57331 57350
```

Estos números representan a la eNodeB y otro representa al equipo de usuario, y se podrá liberar este proceso.

Otra forma de verificar que existan y luego cerrarlos es mediante estos comandos:

```
#Matar Procesos Previos
bash# Verificar qué procesos están usando los puertos
sudo lsof -i :2000
sudo lsof -i :2001
# Matar todos los procesos srsRAN
sudo pkill -f srs
# Verificar que se cerraron
pgrep -l srs
# Si aún hay procesos, forzar cierre
sudo pkill -9 -f srs
```

Se iniciará el UE y se podrán ver logs que son los mensajes de errores, también pondrá el código IMSI, guardará los scripts de limpieza cada que inicie el sistema completo.

```
# Iniciar UE
        echo "Iniciando UE virtual con IMSI: 001099000000001..."
        sudo $SRSUE_CMD $UE_CONFIG > /tmp/srsran_logs/ue.log 2>&1 &
        UE_PID=$!
        echo ""
        echo "=== Sistema RAN Iniciado ==="
        echo "eNodeB PID: $ENB_PID"
        echo "UE PID: $UE_PID"
        echo ""
        # Guardar PIDs para scripts de limpieza
        echo $ENB_PID > /tmp/enb.pid
        echo $UE_PID > /tmp/ue.pid
```

Y para ejecutar el sistema se guardará el script y se hará ejecutable para poder iniciar el sistema completo:

```
chmod +x start_virtual_test.sh
sudo ./start_virtual_test.sh
```

Con estos dos comandos el primero es para asegurarnos que se pueda ejecutar sin problemas y el segundo es para ejecutar todo el sistema.

```
facsistel@facsistel-desktop:~$ sudo ./start_virtual_test.sh
[sudo] contraseña para facsistel:
=== Iniciando Prueba UE Virtual ===
Verificando conectividad con EPC...
Verificando archivos...
Limpiando procesos srsRAN previos...
Iniciando eNodeB virtual...
Esperando que eNodeB esté listo...
Iniciando UE virtual con IMSI: 001099000000001...

=== Sistema RAN Iniciado ===
eNodeB PID: 57986
UE PID: 58035

Comandos utilizados:
- eNodeB: sudo /home/facsistel/srsRAN_4G/build/srsenb/src/srsenb /home/facsistel/srsRAN_4G/srsran_configs/enb.conf
- UE: sudo /home/facsistel/srsRAN_4G/build/srsue/src/srsue /home/facsistel/srsRAN_4G/srsran_configs/ue_virtual.conf

Logs disponibles:
- eNodeB: tail -f /tmp/srsran_logs/enb.log
- UE: tail -f /tmp/srsran_logs/ue.log

Comandos útiles:
- Ver procesos RAN: pgrep -l srs
- Monitorear: watch -n 1 'pgrep -l srs'
- Terminar: sudo kill 57986 58035
```

Figura 142 Sistema Iniciado correctamente

Para verificar que exista una comunicación entre el EPC y el eNodeB aparecerá un mensaje cuando se conecte a la EPC y verificara los claves de acceso.

```
05/22 12:46:59.996: [mme] INFO:     ENB_UE_S1AP_ID[1] MME_UE_S1AP_ID[1] TAC[1] CellID[0x19b01] (../src/mme/s1ap-handler.c:607)
05/22 12:46:59.996: [mme] INFO: Unknown UE by GUTI[G:2,C:1,M_TMSI:0xc000041e] (../src/mme/mme-context.c:3840)
05/22 12:46:59.996: [mme] INFO: [Added] Number of MME-UEs is now 1 (../src/mme/mme-context.c:3631)
05/22 12:46:59.996: [emm] INFO: [] Attach request (../src/mme/emm-sm.c:474)
05/22 12:46:59.996: [emm] INFO:     GUTI[G:2,C:1,M_TMSI:0xc000041e] IMSI[Unknown IMSI] (../src/mme/emm-handler.c:246)
05/22 12:47:00.018: [emm] INFO: Identity response (../src/mme/emm-sm.c:444)
05/22 12:47:00.018: [emm] INFO:     IMSI[001099000000001] (../src/mme/emm-handler.c:482)
05/22 12:47:00.086: [mme] INFO: [Added] Number of MME-Sessions is now 1 (../src/mme/mme-context.c:5162)
05/22 12:47:00.128: [sgwc] INFO: [Added] Number of SGWC-UEs is now 1 (../src/sgwc/context.c:239)
05/22 12:47:00.128: [sgwc] INFO: [Added] Number of SGWC-Sessions is now 1 (../src/sgwc/context.c:924)
05/22 12:47:00.128: [sgwc] INFO: UE IMSI[001099000000001] APN[internet] (../src/sgwc/s11-handler.c:268)
05/22 12:47:00.128: [sgwu] INFO: UE F-SEID[UP:0x1d1 CP:0x46f] (../src/sgwu/context.c:170)
05/22 12:47:00.128: [sgwu] INFO: [Added] Number of SGWU-Sessions is now 1 (../src/sgwu/context.c:175)
05/22 12:47:00.128: [gtp] INFO: gtp_connect() [127.0.0.4]:2123 (../lib/gtp/path.c:60)
05/22 12:47:00.129: [smf] INFO: [Added] Number of SMF-UEs is now 1 (../src/smf/context.c:1033)
05/22 12:47:00.129: [smf] INFO: [Added] Number of SMF-Sessions is now 1 (../src/smf/context.c:3190)
05/22 12:47:00.129: [smf] INFO: UE IMSI[001099000000001] APN[internet] IPv4[10.45.0.2] IPv6[] (../src/smf/s5c-handler.c:303)
05/22 12:47:00.132: [upf] INFO: [Added] Number of UPF-Sessions is now 1 (../src/upf/context.c:209)
05/22 12:47:00.132: [gtp] INFO: gtp_connect() [127.0.0.4]:2152 (../lib/gtp/path.c:60)
05/22 12:47:00.132: [gtp] INFO: gtp_connect() [127.0.0.4]:2152 (../lib/gtp/path.c:60)
05/22 12:47:00.132: [upf] INFO: UE F-SEID[UP:0x9a CP:0xb1a] APN[internet] PDN-Type[1] IPv4[10.45.0.2] IPv6[] (../src/upf/context.c:495)
05/22 12:47:00.132: [upf] INFO: UE F-SEID[UP:0x9a CP:0xb1a] APN[internet] PDN-Type[1] IPv4[10.45.0.2] IPv6[] (../src/upf/context.c:495)
05/22 12:47:00.132: [gtp] INFO: gtp_connect() [192.168.10.101]:2152 (../lib/gtp/path.c:60)
05/22 12:47:00.132: [gtp] INFO: gtp_connect() [192.168.10.101]:2152 (../lib/gtp/path.c:60)
05/22 12:47:00.401: [emm] INFO: [001099000000001] Attach complete (../src/mme/emm-sm.c:1548)
05/22 12:47:00.401: [emm] INFO:     IMSI[001099000000001] (../src/mme/emm-handler.c:286)
05/22 12:47:00.401: [emm] INFO:     UTC [2025-05-22T17:47:00] Timezone[0]/DST[0] (../src/mme/emm-handler.c:292)
05/22 12:47:00.401: [emm] INFO:     LOCAL [2025-05-22T12:47:00] Timezone[-18000]/DST[0] (../src/mme/emm-handler.c:296)
05/22 12:47:00.401: [gtp] INFO: gtp_connect() [192.168.10.102]:2152 (../lib/gtp/path.c:60)
```

Figura 143 Conexión al EPC

Como se puede observar en la figura 143, se estableció una conexión exitosa y se estableció sesiones exitosas de conexión para nuestra red y lo que quiere decir que es completamente funcional estos se muestran en las siguientes líneas:

```
ENB_UE_S1AP_ID[1] MME_UE_S1AP_ID[1]
[001099000000001] Attach request
Identity response
#Sesiones establecidas

Number of MME-Sessions is now 1

Number of SGWC-Sessions is now 1

Number of SMF-Sessions is now 1

Number of UPF-Sessions is now 1

#Ip asignada al UE
UE IMSI[001099000000001] APN[internet] IPv4[10.45.0.2]

# Attach Completado:
[001099000000001] Attach complete

#Túneles GTP Funcionando:

gtp_connect() [192.168.10.101]:2152

gtp_connect() [192.168.10.102]:2152
```

Todos estos mensajes muestran que la red funciona correctamente y establece la comunicación entre el UE y la EPC.

Se implemento un sistema de red LTE funcional completamente virtualizado donde utilizamos srsRAN 4G como RAN (Radio Access Netwok) y open5GS como la EPC (Evolved Packet Core), desarrollamos diferentes scripts y tenemos resultados de conexión. Primero se verifico que el sistema funcionara correctamente mediante el comando:

```
ip addr show
```

Con este comando vemos podemos verificar si los equipos de usuarios se mantienen activos, en el caso de la configuración de los UE por defecto se tiene que se conecta y se desconecta cada 36 segundos

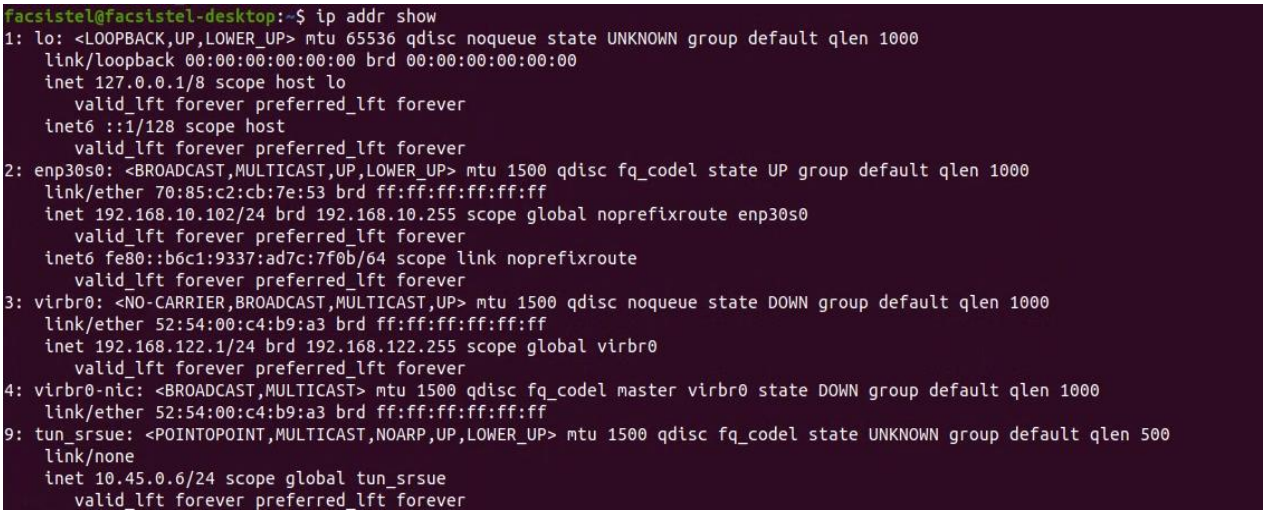


Figura 144 Verificación de la Arquitectura

En la figura 144, se configura la interfaz tun esto quiere decir que se escucha exitosamente y se mantienen activos los equipos de usuarios. Para verificar que los equipos de usuarios tengan conexión haremos un ping:

```
ping 10.45.0.6
```

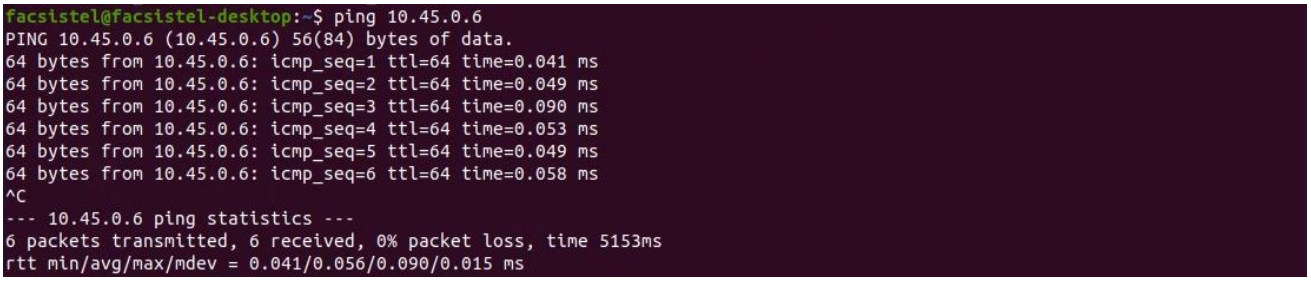


Figura 145 Pin de Conexión

Podemos comprobar que existe una comunicación estable y se estable un pin de conexión en la figura 145. Otro comando que se utiliza para verificar las conexiones es:

Con este comando podemos verificar la conexión de la arquitectura completamente, para entender mejor todo este seguimiento se explicará a continuación:

```
ip route | grep 10.45.0
```

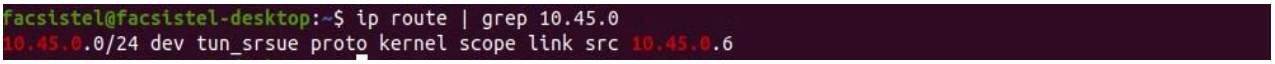


Figura 146 Comunicación con la interfaz tun_srsue

En el siguiente diagrama de flujo daremos a conocer los pasos que se utilizan para crear la conexión con el equipo de usuario.

Figura 147 Flujo de comunicación

Primero se inicia la red mediante ZeroMQ que sirve para virtualizar eNodeB, luego en el apartado Cell Discovery el equipo de usuario se conecta a la celda PC1= 1, Y PRB= 50 con un total de 10 MHz, seguidamente se conecta al RACH (Random Access Channel) el equipo de usuario solicita acceso aleatorio a la red, RCC connectión establece una señalización RRC entre el UE y eNodeB, luego se registra al núcleo de la red informando su identidad, capacidad y estado, y por último se le asigna una IP

Este proceso es para verificar la conexión completa cuando se usa Software de código libre como Open5GS y SrsRAN.

4.1.5. Prueba de claves de seguridad

Se hizo una prueba con una IMSI que no está registrada en la base de datos de MongoDB, por lo tanto, nuestra EPC detectara el UE, pero no lo dejara conectarse en la arquitectura ya que no consta en los registros de la HSS y MME.

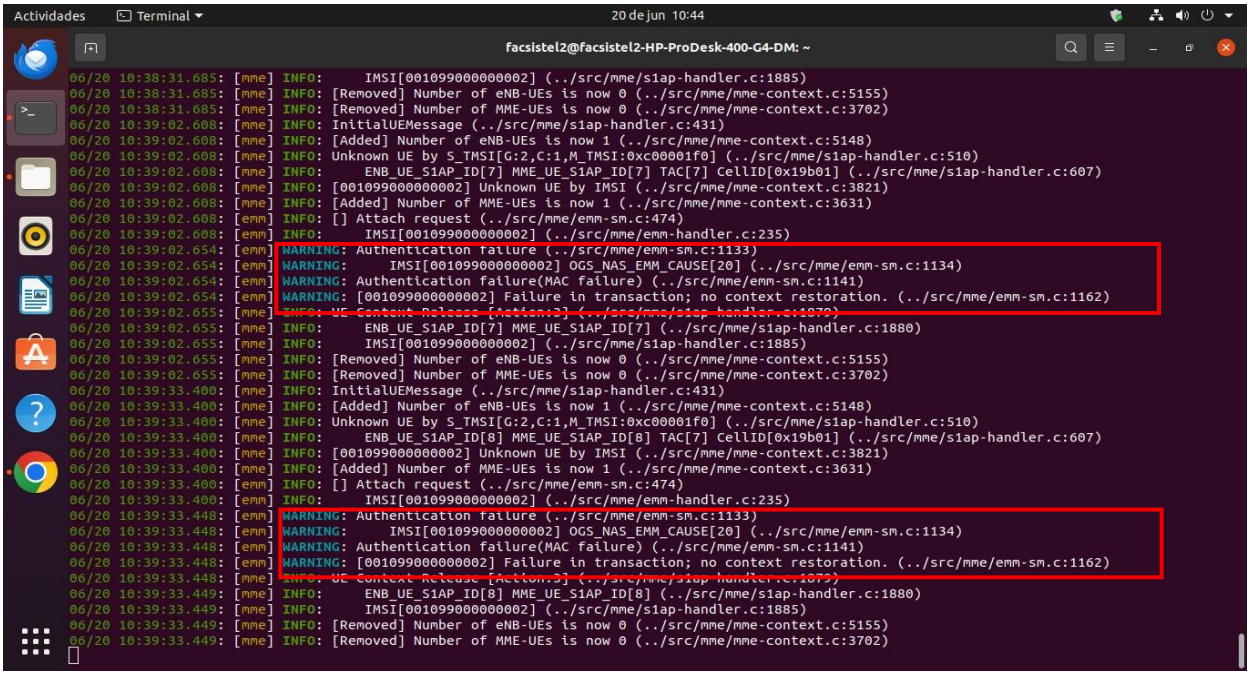


Figura 148 Prueba de claves de seguridad

En la figura 147, se puede visualizar mensajes de warnings que va a detectar la IMSI, pero mas no la dejara conectarse porque no está registrado en la base de datos de MongoSH y que conectan con la HSS y MME.

06/20 10:39:02.654: [emm] WARNING: Authentication failure (../src/mme/emm-sm.c:1133)

06/20 10:39:02.654: [emm] WARNING: IMSI[001099000000002] OGS_NAS_EMM_CAUSE[20] (../src/mme/emm-sm.c:1134)

06/20 10:39:02.654: [emm] WARNING: Authentication failure(MAC failure) (../src/mme/emm-sm.c:1141)

06/20 10:39:02.654: [emm] WARNING: [001099000000002] Failure in transaction; no context restoration. (../src/mme/emm-sm.c:1162)

El mensaje de OGS_NAS_EMM se trata de que la autenticación fallo recibió una solicitud de conexión, pero fallo, debido a que el código de integridad del mensaje no coincide dando esto sucede debido a que las claves K y OPC no coinciden con las que se almacenos en la HSS.

El mensaje MAC Failure, es cuando la entidad MME compara el código de autenticación MAC que envió el UE, con el cálculo interno no coinciden. Esto valida que el HSS recibió la solitud, pero fallo por parámetros incorrectos.

Y por último el MME cierra la transacción porque la autenticación no fue exitosa, y por seguridad no guarda ningún contexto de UE. Este comportamiento es normal según el estándar de LTE.

Estas pruebas son útiles ya que se pueden conectar a la arquitectura con UE clonadas o mal configuradas, validando el correcto funcionamiento de la arquitectura LTE.

4.2. Resultados

Los resultados obtenidos se aprecian de manera tabulada en las siguientes tablas, explicando el proceso que se lleva a cabo al levantar la arquitectura LTE simulada, por medio de los softwares de código abierto srsRAN y Open5GS.

Primero observamos el proceso de la EPC y las entidades que trabajan en todo el proceso de levantamiento de la red, por medio del software Open5Gs.

PROCESOS DE LA EPC	
Descripción del Proceso	**Entidades Involucradas**
Se ejecuta el script start-epc.sh y se detienen procesos Open5GS anteriores	Usuario, Shell, Sistema
Se lanza el MME y se conecta correctamente	MME, FreeDiameter
Se lanza el HSS y se conecta con éxito al MME	HSS, MongoDB, FreeDiameter
Se lanza SGW-C y se inicia su servidor PFCP y GTP	SGW-C
Se lanza SGW-U y se configura su interfaz PFCP y GTP	SGW-U
Se lanza SMF y establece múltiples interfaces (GTP, PFCP, SBI), intenta registrarse con NRF	SMF, PFCP, SBI, NRF
Se lanza UPF, configurando GTP y PFCP para el plano de usuario	UPF, GTP, PFCP
Se lanza PCRF y se conecta con MME y SMF	PCRF, FreeDiameter, MongoDB
Se lanza el NRF y se registran varias funciones de red (NF)	NRF, SBI, SMF, PCRF
El sistema detecta que hay 9 procesos corriendo, cuando lo esperado eran 8	Sistema, Usuario
Establece conexión con el MME mediante la interfaz S1	eNodeB, MME

Tabla 14 Procesos del EPC

A continuación, vemos como se levanta nuestra eNodeB y se ajusta sus parámetros de radio para el SDR por medio del software srsRAN:

PROCESOS eNodeB	
Descripción del Proceso	**Entidades que Intervienen**
Se lanza el binario srsENB desde el directorio de compilación de srsRAN	Usuario, Sistema Operativo, srsRAN
Se carga el archivo de configuración enb_virtual.conf	srsRAN (srsENB)
Se detectan y activan los plugin RF disponibles (ej. BladeRF)	librerías RF, BladeRF
Se abre el canal de comunicación RF con parámetros configurados	srsENB, BladeRF
Se informa frecuencia y parámetros de operación (ej. DL=1850 MHz)	srsENB
Se muestra mensaje en la eNodeB started indicando que está listo	srsENB

Tabla 15 Procesos de eNodeB

Como Punto final tenemos pruebas de conexión, autentificación y reconocimiento de dispositivos UE, por medio de srsRAN y Open5gs, en donde apreciamos el proceso para que nuestros dispositivos virtuales se conecten a nuestra red LTE.

PROCESO DE CONEXIÓN DEL UE REGISTRADO	
Descripción del Proceso	**Entidades que Intervienen**
Conexión de la eNodeB al núcleo de red (MME)	eNodeB, MME
Inicio de conexión del UE (mensaje inicial)	UE, eNodeB, MME
Solicitud y validación de identidad del usuario	EMM, UE, MME
Registro del UE y creación de sesión de control	MME
Configuración del plano de usuario y asignación de APN	SGW-C, SGW-U
Establecimiento de túneles GTP para señalización	GTP, SGW, MME
Gestión de sesión de datos y asignación de dirección IP	SMF
Confirmación de sesiones y asignación final de túneles GTP-U	SGW-U, GTP, eNodeB
Confirmación de identidad y sincronización de zona horaria	EMM, MME

Tabla 16 Prueba de conexión de la UE

Aquí como punto adicional procedemos a verificar la seguridad de nuestra red insertando un UE no registrado en l base de datos, viendo como es rechazado por la MME.

PROCESO DE CONEXIÓN DEL UE NO REGISTRADO	
Descripción del Evento	**Entidad Principal**
El MME recibe el IMSI del UE [001099000000002]	MME
Se agrega el UE a la lista de eNB-ÚES y MME-UES	MME
Se recibe un mensaje InitialUEMessage	MME
Se identifica al UE como desconocido por S-TMSI	MME
Se asocia el UE a un ID interno (ENB_UE_S1AP_ID) y celdas TAC/CellID	MME
Se detecta el IMSI del UE como desconocido y se registra	MME
Se recibe la solicitud de conexión Attach Request	MME
Se inicia el proceso de autenticación con Authentication Request	MME
Se produce fallo de autenticación (MAC Failure) con código OGS_NAS_EMM_CAUSE [20]	MME, NAS
El MME aborta la transacción y libera el contexto del UE (UE Context Reléase)	MME
Se eliminan las entradas del UE en las listas del MME y eNodeB	MME

Tabla 17 Prueba de seguridad y autentificación del UE

4.2.1. Pruebas de tráfico generado con WireShark

Para entender de mejor manera lo que sucede en las pruebas realizadas, usaremos la herramienta Wireshark para monitorear el tráfico que se genera al realizar las pruebas realizadas anteriormente.

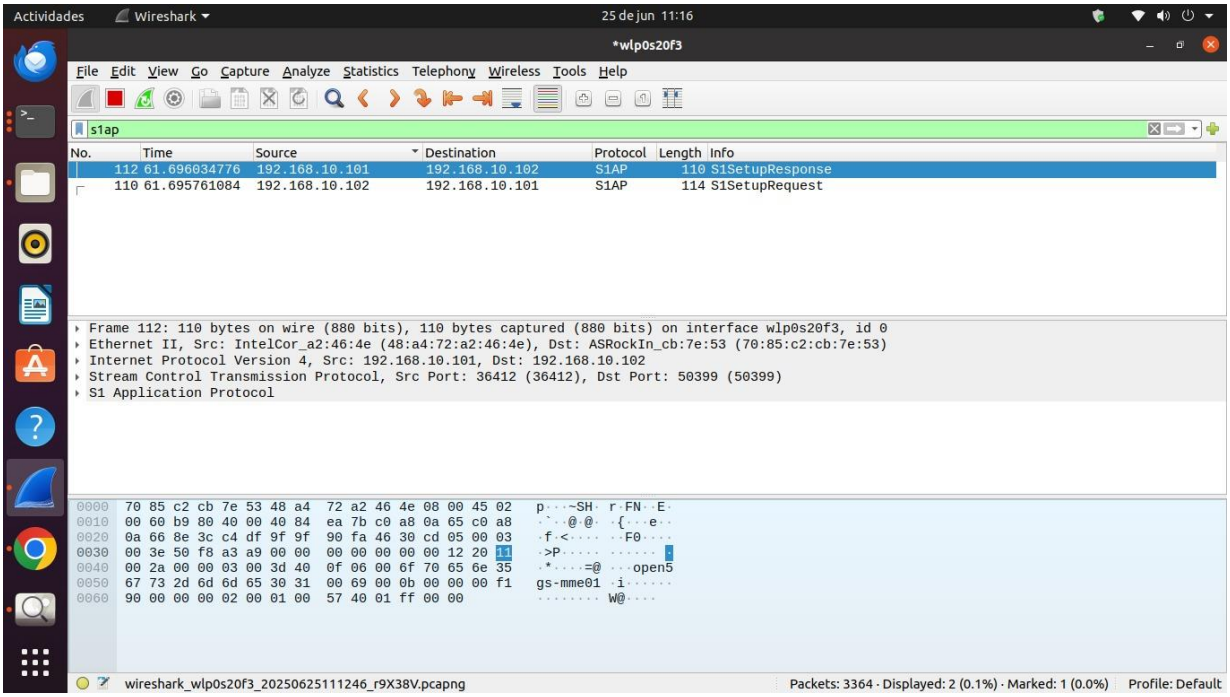
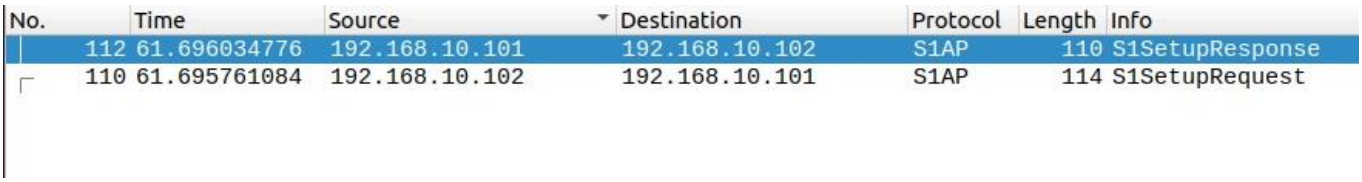


Figura 149 Conexión entre eNodeB y EPC

En la figura anterior se observa como obtenemos información de la eNodeB (192.168.10.102) y la EPC (192.168.10.101), esto luego de haber encendido amabas respectivamente por medio de los comandos que se mencionaron antes.

A continuación analizaremos que está pasando al momento de que se conecte la eNodeB con la EPC.

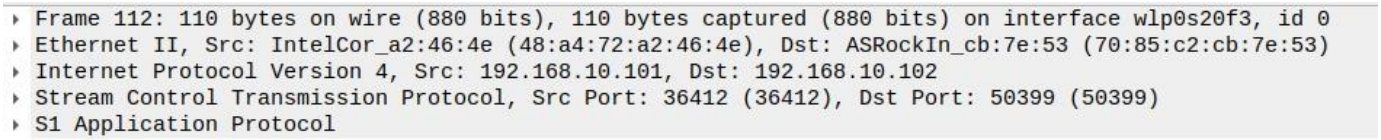


No.	Time	Source	Destination	Protocol	Length	Info
112	61.696034776	192.168.10.101	192.168.10.102	S1AP	110	S1SetupResponse
110	61.695761084	192.168.10.102	192.168.10.101	S1AP	114	S1SetupRequest

Figura 150 EPC y eNodeB por medio de la interfaz S1 en Wireshark

Como se observa en la figura anterior, se nos presenta las direcciones 192.168.10.101 y 192.168.10.102, las cuales están usando el protocolo S1AP, que hace referencia a la interfaz S1 que es la conecta la eNodeB con la MME, que es lo que sucede en una red LTE.

Primero iniciaremos explicando todas las interfaces y protocolos usados en esta conexión.



```
▸ Frame 112: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface wlp0s20f3, id 0
▸ Ethernet II, Src: IntelCor_a2:46:4e (48:a4:72:a2:46:4e), Dst: ASRockIn_cb:7e:53 (70:85:c2:cb:7e:53)
▸ Internet Protocol Version 4, Src: 192.168.10.101, Dst: 192.168.10.102
▸ Stream Control Transmission Protocol, Src Port: 36412 (36412), Dst Port: 50399 (50399)
▸ S1 Application Protocol
```

Figura 151 Interfaces que actúan en la conexión

Como observamos en la Figura anterior esta conexión se verá reflejada mediante la interfaz wlp020f3, Ethernet II, Internet Protocolo versión 4 y el puerto 36412 que es por donde se conecta la eNodeB con la EPC, además del protocoló S1 que se mencionó anteriormente.

Se usa Ethernet y IPv4, ya que tenemos conexión vía cable UTP en la eNodeB y conexión vía WiFi en la EPC.

Ahora veremos qué es lo que sucede al momento de la conexión para entender, Primero revisaremos la eNodeB que es que enviara una solicitud para conectarse a la EPC.

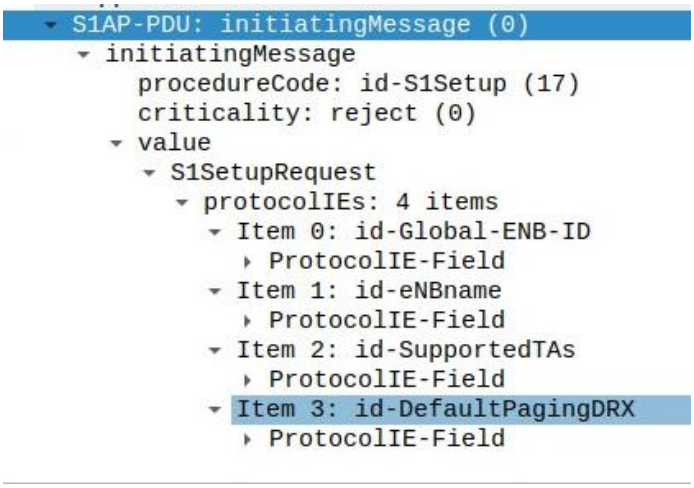


Figura 152 Solicitud de la eNodeB a la MME

Ahora lo que sucede es que se manda la solicitud por medio de un InitiatingMessage, se detecta la Id por medio de la interfaz S1 en S1Setup, a continuación, se verificaran los datos de nuestra eNodeB para su autenticación, mostrándonos la ID, el nombre de asignado, para que pueda ser detectado por la EPC.

Ahora veremos que sucede en la EPC, cuando detecta la eNodeB y la acepta su solicitud

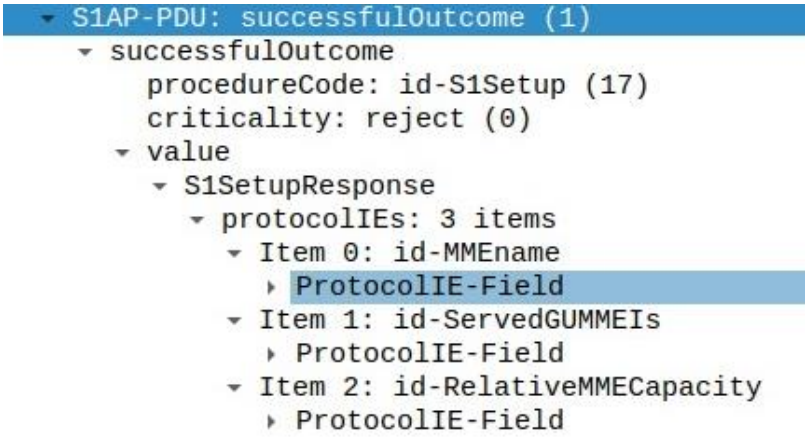


Figura 153 Solicitud de eNodeB aceptada

Como observamos en la Figura anterior, obtenemos a través de la interfaz S1AP, un successfulOutCome, lo nos dice que detecto eNodeB, luego se procede a leer la ID por medio de id-S1Setup, este a su vez envía una respuesta S1SetupResponse y registra los datos de eNodeB en la MME, y los almacena en MMEname, id ServedGUMMEIs e Id- RelativeMMECapacity. Una vez hecho esto se conectaran la EPC y eNodeB y veremos los mensajes en la terminal de la EPC, verificando que están comunicándose una con otra.

```
[mme] INFO: eNB-S1 accepted[192.168.10.102]:46537 in s1_path module (../src/mme/s1ap-sctp.c:114)
[mme] INFO: eNB-S1 accepted[192.168.10.102] in master_sm module (../src/mme/mme-sm.c:108)
[mme] INFO: [Added] Number of eNBs is now 1 (../src/mme/mme-context.c:3031)
[mme] INFO: eNB-S1[192.168.10.102] max_num_of_ostreams : 30 (../src/mme/mme-sm.c:157)
```

Figura 154 Terminal de EPC - Mensajes de conexión

Ahora veremos que sucede si conectamos la UE virtual que tenemos registrada en nuestra base de datos MongoDB.

No.	Time	Source	Destination	Protocol	Length	Info
2688	26.805835495	192.168.10.102	192.168.10.101	S1AP	114	S1SetupRequest
2690	26.806060654	192.168.10.101	192.168.10.102	S1AP	110	S1SetupResponse
44840	507.353227658	192.168.10.102	192.168.10.101	S1AP/N…	150	InitialUEMessage, Attach request, PDN connectivity request
44841	507.375749947	192.168.10.101	192.168.10.102	S1AP/N…	106	DownlinkNASTransport, Identity request
44842	507.400900740	192.168.10.102	192.168.10.101	S1AP/N…	138	UplinkNASTransport, Identity response
44849	507.493224399	192.168.10.101	192.168.10.102	S1AP/N…	138	DownlinkNASTransport, Authentication request
44852	507.527269730	192.168.10.102	192.168.10.101	S1AP/N…	130	UplinkNASTransport, Authentication failure (MAC failure)
44853	507.527507383	192.168.10.101	192.168.10.102	S1AP/N…	106	DownlinkNASTransport, Authentication reject
44854	507.527527354	192.168.10.101	192.168.10.102	S1AP	82	UEContextReleaseCommand [NAS-cause=normal-release]
44857	507.528675578	192.168.10.102	192.168.10.101	S1AP	82	UEContextReleaseComplete

Figura 155 Procesos para la conexión de la UE

En la imagen anterior podemos observar como eNodeB y la EPC, hacen el proceso de conexión y autenticación del UE

```
▾ S1 Application Protocol
  ▾ S1AP-PDU: initiatingMessage (0)
    ▾ initiatingMessage
        procedureCode: id-initialUEMessage (12)
        criticality: ignore (1)
      ▾ value
        ▾ InitialUEMessage
          ▾ protocolIEs: 6 items
            ▾ Item 0: id-eNB-UE-S1AP-ID
              ▸ ProtocolIE-Field
            ▾ Item 1: id-NAS-PDU
              ▾ ProtocolIE-Field
                  id: id-NAS-PDU (26)
                  criticality: reject (0)
                ▾ value
                    NAS-PDU: 1773f7eb1a040741210bf600f190000201c000049002f070…
                  ▾ Non-Access-Stratum (NAS)PDU
                      0001 .... = Security header type: Integrity protected (1)
                      .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
                      Message authentication code: 0x73f7eb1a
                      Sequence number: 4
                      0000 .... = Security header type: Plain NAS message, not security protected (0)
                      .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
```

Figura 156 Solicitud de UE a la EPC

En la imagen anterior observamos el primer proceso que se lleva a cabo en la EPC, (192.168.10.101), donde se comienza recibiendo el InitialUEMessage que es la solicitud de la UE, después se procede a leer las claves de seguridad NAS (Non Access Stratum), como se en el apartado NAS-PDU, siendo detectados por EMM (EPS Mobility Management).

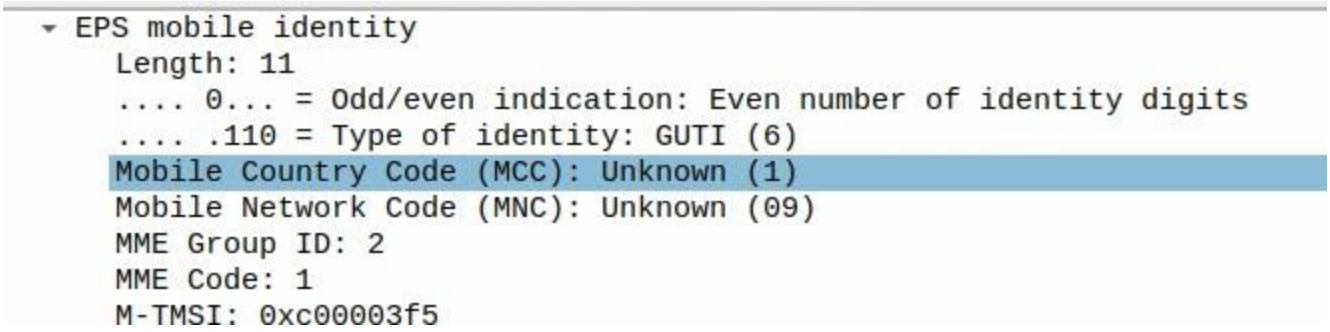


Figura 157 Verificación de MCC y MNC por EPS Mobile Indentity

En la Figura 157 se observa los códigos de identificación de la red móvil LTE MCC y MNC que se le asigno a las IMSI (International Mobile Subscriber Indentity) en MongoDB.

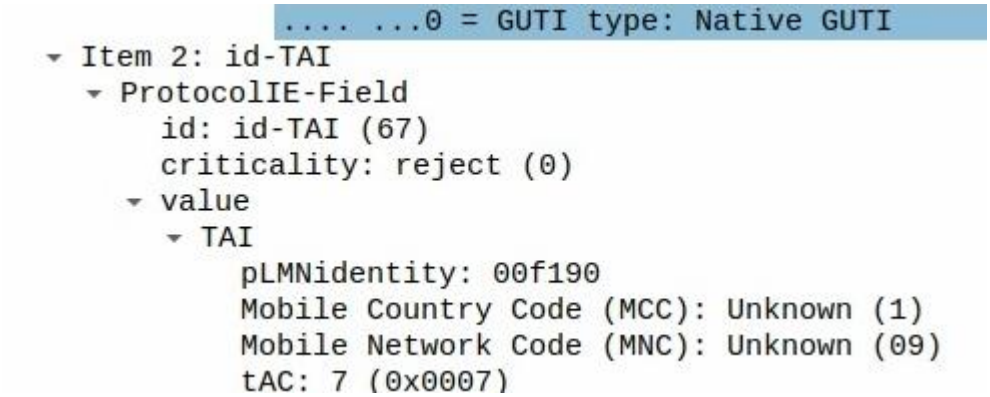


Figura 158 Detección de TAC por medio de MNC y MCC

A continuación, observamos en la Figura 158 la detección del TAC (Tracking Área Code), por medio del MNC y MCC del UE configurado.

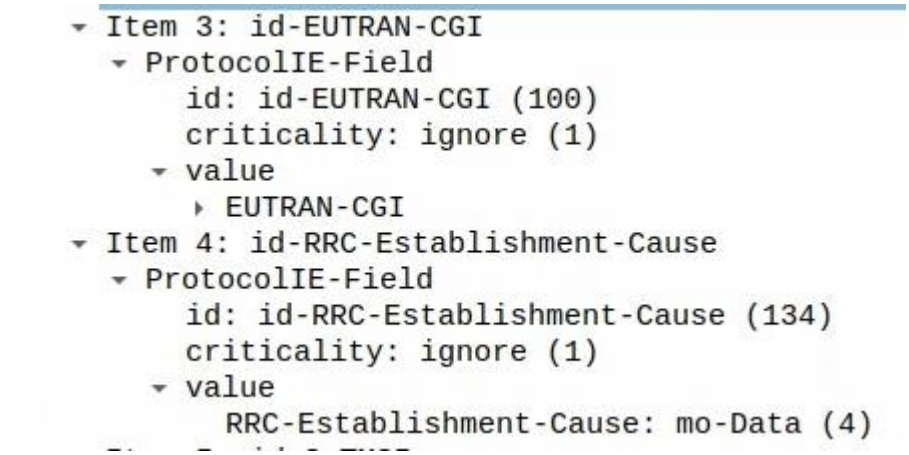


Figura 159 conexión con la E – UTRAN por medio del protocolo RRC

En la Figura 159 se observa que la red de acceso E-UTRAN y el protocolo RRC (Radio Resource Control) que es el encargado de gestionar la conexión entre el UE y eNodeB.

Ahora veremos lo que sucede en eNodeB (192.168.10.102).

```
▾ ProtocolIE-Field
    id: id-NAS-PDU (26)
    criticality: reject (0)
  ▾ value
      NAS-PDU: 075501
    ▾ Non-Access-Stratum (NAS)PDU
        0000 .... = Security header type: Plain NAS message, not security protected (0)
        .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
        NAS EPS Mobility Management Message Type: Identity request (0x55)
        0000 .... = Spare half octet: 0
        .... 0001 = Identity type 2: IMSI (1)
```

Figura 160 Clave NAS del UE detectado por MME

Como se observa en la Figura 160 se procede a enviar un mensaje NAS a la MME, la cual se observa en el apartado NAS-PDU, y se procede con la solicitud para verificar la IMSI del usuario, eso se puede observar en el apartado Indentity Type 2: IMSI (1), donde detecto un UE con la USIM terminada en 0001, que es la primera USIM que configuramos en MongoDB.

Luego el Proceso cambia a la EPC (192.168.10.101)

```
      id: id-eNB-UE-S1AP-ID (8)
      criticality: reject (0)
    ▾ value
        ENB-UE-S1AP-ID: 1
▾ Item 2: id-NAS-PDU
  ▾ ProtocolIE-Field
      id: id-NAS-PDU (26)
      criticality: reject (0)
    ▾ value
        NAS-PDU: 0756080910900900000010
      ▾ Non-Access-Stratum (NAS)PDU
          0000 .... = Security header type: Plain NAS message, not security protected (0)
          .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
          NAS EPS Mobility Management Message Type: Identity response (0x56)
        ▾ Mobile identity - IMSI (001099000000001)
            Length: 8
            0000 .... = Identity Digit 1: 0
            .... 1... = Odd/even indication: Odd number of identity digits
            .... .001 = Mobile Identity Type: IMSI (1)
            IMSI: 001099000000001
          ▾ [Association IMSI: 001099000000001]
              Mobile Country Code (MCC): Unknown (1)
              Mobile Network Code (MNC): Unknown (099)
▾ Item 3: id-EUTRAN-CGI
  ▾ ProtocolIE-Field
      id: id-EUTRAN-CGI (100)
```

Figura 161 Identificación de USIM por parte de NAS

Como vemos en la Figura 161 se recepta la solicitud y se identifica la USIM del UE que vimos antes en eNodeB (192.168.10.102), pero ahora nos muestra la USIM completa 00010990000000001,

identificando el UE que registramos en MongoDB y asociándolo con la base de datos. Esto se aprecia en los apartados Mobility Indentity.

A continuación, volvemos a eNodeB (192.168.10.102).

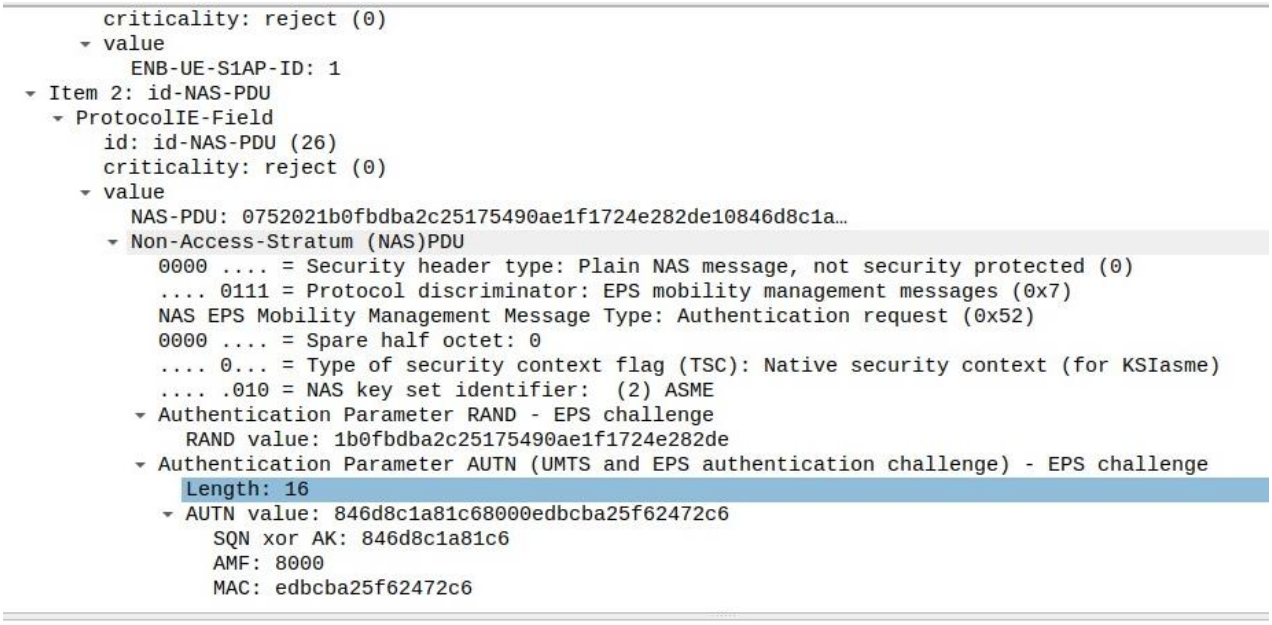


Figura 162 Autenticación del UE

Como observamos en la Figura 162 una vez realizada la identificación del USIM, se procede a enviar un mensaje codificado que es que se aprecia en NAS – PDU, en donde se enviará a la UE para poder Autentificarse, una vez hecho esto la UE deberá responder.

A continuación, volvemos a la EPC (192.168.10.101).

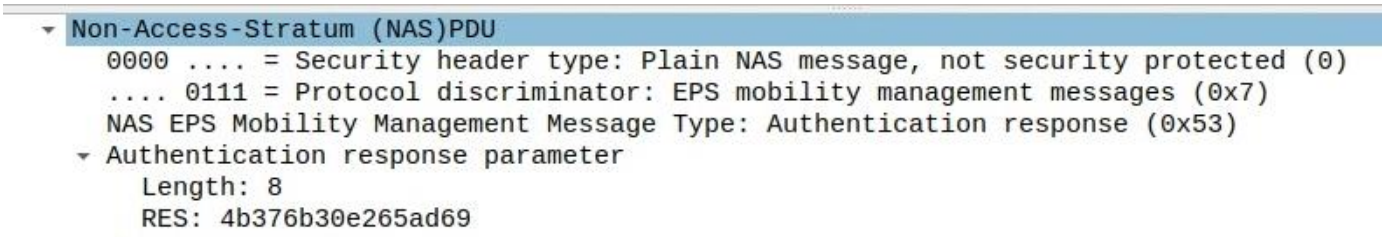


Figura 163 Respuesta de la UE hacia la solicitud de la MME

Como se observa en la Figura 163 se procede a verificar el mensaje de la UE, que se define como RES: 4b376b30e265ad69, entonces la UE está lista para su autenticación.

Ahora volvemos a eNodeB, donde observamos:

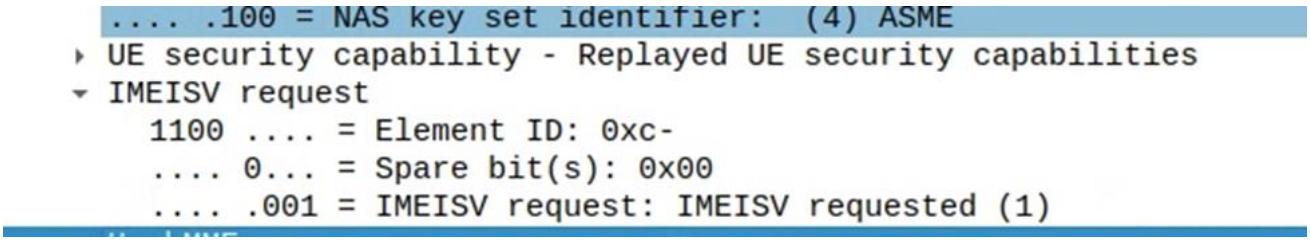


Figura 164 Envió de la IMEI por parte del UE

Se observa en la Figura 164 una identificación IMEI enviada por parte del UE, verificado en el apartado IMEISV Request, donde también identificamos los 3 últimos dígitos de la IMSI.

Ahora regresamos a la EPC donde observamos:

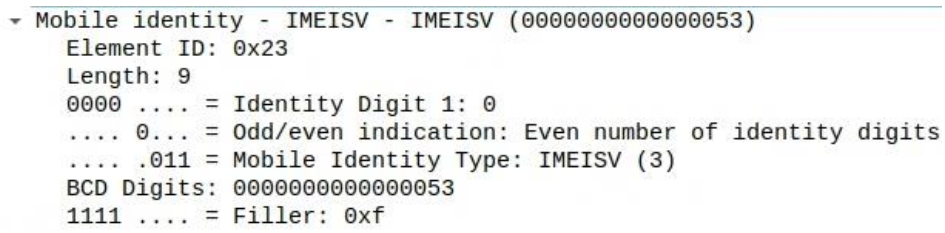


Figura 165 Recepción de la IMEI enviada por la UE

Se recibe el mensaje NAS-PDU y se recepta el IMEI del UE, eso lo observamos en el apartado Mobile Indentity, y observamos el IMEI del dispositivo en BCD Digits: 3534900698733153

Si revisamos el archivo ue_virtual.conf y la base de datos MongodB. Observaremos que la IMEI del UE coinciden, tal y como vemos en la Figura 166

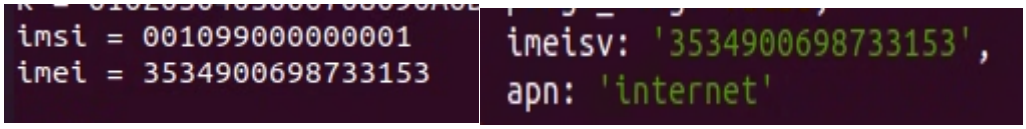


Figura 166 IMEI insertada en la ue_virtual.conf y en MongoDB

Ahora regresamos a eNodeB y revisamos el estado de la autenticación.

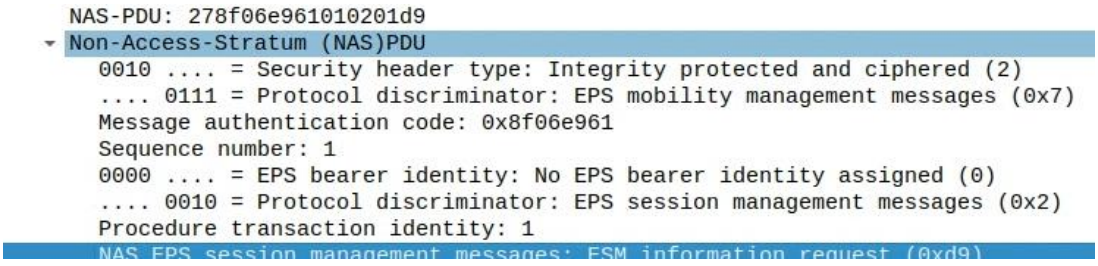


Figura 167 Autenticación y Sesión

Como se observa en la figura 167 se observa el mensaje autentificación, y se comienza a iniciar la sesión, esto lo observamos en EPS sesión Management Message.

Otra vez regresamos a la EPC y observamos:

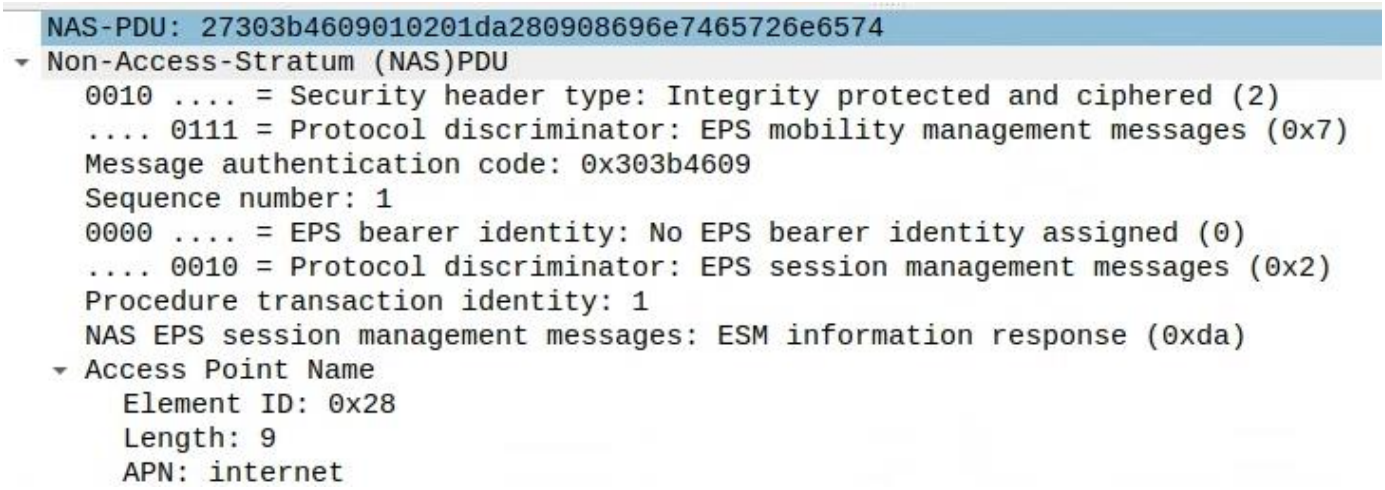


Figura 168 Liberación del Punto de Acceso APN por la MME

Como vemos en la Figura anterior se confirma la sesión y se presenta el APN = Internet, que es el punto de Acceso donde se conectara nuestra UE.

Luego volvemos a eNodeB para verificar que el UE logro iniciar sesión:

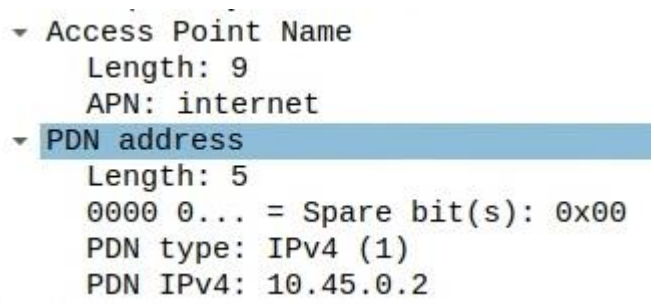


Figura 169 Conexión de UE y asignación de una IP

Como observamos en la Figura anterior, el UE logro conectarse por medio de la APN = internet y se le asigno la dirección IP 10.45.0.2.

Ahora en la EPC observaremos el estado del UE.

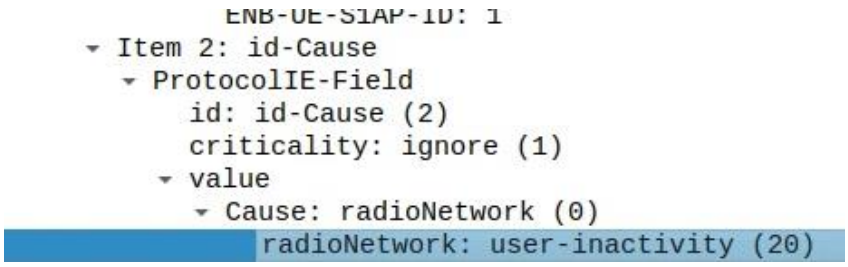


Figura 170 UE Inactivo sin tráfico real

Como vemos en la Figura 170, el UE presenta Inactividad por falta de tráfico real, esto provoca que el UE entre en modo de espera, por lo que se desconectara hasta que generemos tráfico, mostrándonos el mensaje en los logs de eNodeB, como se muestra en la Figura 171.

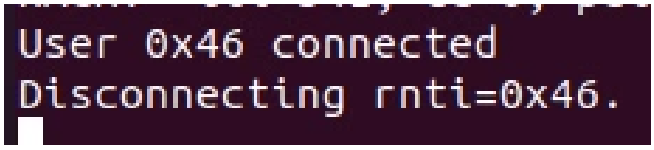


Figura 171 Logs de eNodeB en su terminal

Ahora usaremos os el comando siguiente para generar tráfico y que nuestra UE se vuelva a conectar:

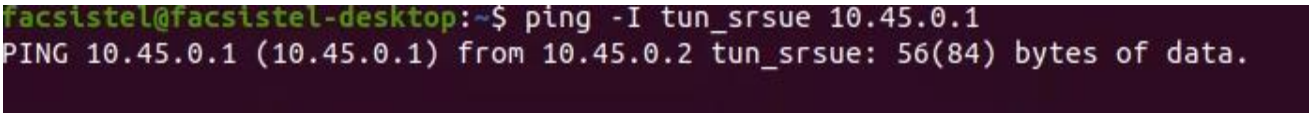


Figura 172 Comando para generar tráfico mediante el servidor tun_srsue

Como ya se mencionó el UE se volverá a conectar y nos mostrará el siguiente mensaje:

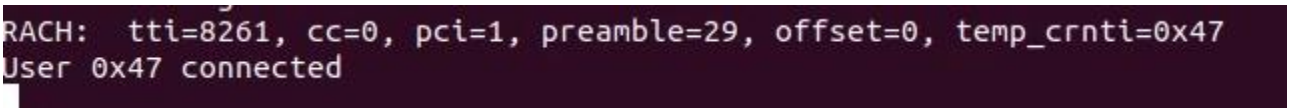


Figura 173 conexión de la UE

Ahora veremos lo que pasa en WireShark, primero comenzaremos por la EPC, se mostrará un mensaje de autenticación esto indicara que el UE virtual fue reconocido y autenticado por la EMM, por medio el protocolo NAS.

Como observamos en la Figura 173 esta vez solo se autenticará al UE, ya que la EPC ya lo tiene registrado, haciendo que el UE salga de modo de espera.

```
▾ Item 1: id-NAS-PDU
  ▾ ProtocolIE-Field
      id: id-NAS-PDU (26)
      criticality: reject (0)
    ▾ value
        NAS-PDU: c7a3e6af
      ▾ Non-Access-Stratum (NAS)PDU
          1100 .... = Security header type: Security header for the SERVICE REQUEST message (12)
          .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
        ▸ KSI and sequence number
        ▾ Short MAC - Message authentication code (short)
            Message authentication code (short): 0xe6af
```

Figura 174 Mensaje de Autenticación.

Se establece conexión y se detecta el tráfico proveniente de eNodeB, como se aprecia en la imagen donde podemos ver la dirección 192.168.10.102.

```
e-RAB-ID: 5
▾ transportLayerAddress: c0a80a66 [bit length 32, 1100 0000  1010 1000  0000 1010  0110 0110 decimal value 323
    transportLayerAddress(IPv4): 192.168.10.102
  gTP-TEID: 00000002
```

Figura 175 Trafico detectado proveniente de 192.168.10.102 (eNodeB)

El tráfico generado lo podemos observar en WireShark filtrando ICMP, donde nos aparecerán los paquetes enviados desde 10.45.0.2 hasta 10.45.0.1, tal y como se observa en la Figura 176.

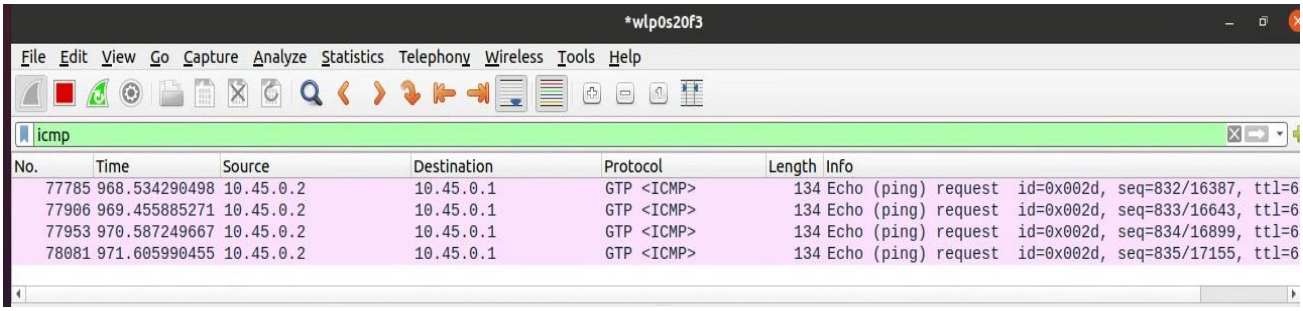


Figura 176 Tráfico de la UE con paquetes ICMP

4.3. Conclusiones

- Se realizo una red LTE simulada por medio de Open5Gs, y srsRAN, ejecutando scripts que simulan la arquitectura de red propia de una LTE, lo que nos ayudó a simularla de acuerdo con las investigaciones realizadas, comprendiendo su funcionamiento.
- Las interfaces de código abierto utilizadas son proyectos basados redes móviles de última generación, por lo tanto, se basan en el estándar internacional 3GPP de redes móviles, lo que nos permite usar scripts que simulan entidades que forman parte de este estándar como MME, HSS y SGW/PGW y los cuales pueden manipularse a través de los scripts para tener una mayor comprensión de estos.
- Se selecciono software de código abierto basado en redes móviles de última generación, entre los cuales están OAI, Open5gs y srsRAN, al principio se intentó hacer la arquitectura completa solo utilizando OAI, pero por problemas de dependencias en nuestra versión de Ubuntu, tuvimos muchas dificultades, al final se optó por usar Open5GS que fue la que simulo la EPC y srsRAN que fue quien simulo la eNodeB.
- Se Implemento una red basada en el estándar 3GPP con Open5Gs y srsRAN, las cuales permitieron crear un entorno manipulable, esto ayuda a poder probar distintos escenarios de manera simulada, enriqueciendo los conocimientos previos que se obtuvieron a lo largo de la carrera.
- Se configuraron las tarjetas USIM programables por medio del Lector MCR3416, el cual nos permitió ver cómo se configuras los chips celulares, a través de claves y asignaciones de usuario, posteriormente se usaron estos datos para nuestra base datos HSS en MongodB, usando claves reales de una USIM, la cual en un futuro puede ser usada para acceder a la red por medio de un dispositivo real.
- Se reviso el tráfico de datos enviados al hacer ping desde el UE virtual (10.45.0.2) y la interfaz ogstun de Open5gs(10.45.0.1), estableciendo conexión con respuesta exitosa, enviado paquetes ICMP los cuales se observaron en WireShark, con el filtro ICMP, en la interfaz wlp0s20f3, obteniendo los parámetros de velocidad por medio del aparatado DL y UL, y la latencia de la misma en TTL, que observamos en Wireshark, también comprobamos la estabilidad de conexión al observar cómo se envían paquetes ICMP de la UE(10.45.0.2) hasta la interfaz ogstun(10.45.0.1) y viceversa.
  Como último punto podemos observar en la consola de WireShark el proceso que realiza la interfaz NAS, para autentificar el UE, además de observar la IP que se asignó.

- La integración de softwares libres como Open5GS y SrsRAN4G nos permitió implementar la red 4G de manera virtualizada, validando los protocolos de comunicaciones, así como el funcionamiento de la infraestructura, donde demuestra ser un entorno funcional, estable y que nos proporciona la posibilidad de modificar sus parámetros, lo cual es muy en útil para realizar pruebas en entornos de investigación y enseñanza.
- A pesar de tener la red implementada, se identificó un obstáculo en el levantamiento de la red en la parte de radio mediante el BladeRF, eNodeB lograr establecer conexión de radio estable con el UE, pero no se conecta a la red, esto fue debido a las antenas del equipo BladeRF XA4 a pesar de trabajar en un rango de frecuencias para LTE solo tiene una ganancia de 5 dBi, por esa razón no trabaja en dichas bandas de frecuencia correctamente ofreciéndonos una ganancia por debajo -6 dBm por lo que no hay transmisión o es muy limitada, ya que lo mínimo que debería tener una antena para una buena transmisión serian -10 dBm.

4.4. Recomendaciones

- El software de código Open5GS actualmente es un proyecto abierto que sigue en constante actualización, y está basada en redes móviles 5G, nosotros realizamos este proyecto bajo redes 4G debido a que en el país aún no se despliega 5G y no contamos con bandas de frecuencia designadas para esta tecnología pero en un futuro el proyecto ser implementado en 5G cuando en el Ecuador decida implementar esta tecnología además del fácil acceso a dispositivos con dicha tecnología, ya que esa fue otra de las razones por la que implementamos 4G, ya que actualmente al no existir una red en el país los dispositivos 5G son muy escasos y todos continúan y la mayoría de dispositivos móviles solo cuentan con tecnología 4G en sus especificaciones.
- Se planteo una solución para el problema de las antenas con la obtención de antenas LTE, las cuales trabajan 15 dBi, lo cual nos permitirá alcanzar los -10 dBm en las bandas que queremos utilizar y así obtener una transmisión adecuada que pueda ser detecta por los dispositivos móviles.
- Se pueda implementar VoLTE con más herramientas como Kamilio IMS y Open5GS, estas plataformas de código abierto permiten crear una propia infraestructura IMS, para acercarse a un entorno más real y completo.
- Se pueden implementar más eNodeB con diferentes computadores para realizar el Handover que consiste en cambiar de celdas cuando se pierda la conexión, consiguiendo otro equipo SDR, para realizar diferentes pruebas en base a celdas.

- Se podría realizar un estudio si se puede implementar la segregación de portadoras para mejorar la red aumentando la capacidad, velocidad de datos ya que el software srsRAN permite la agregación de portadoras
- El Software Open5gs soporta la tecnología CSFB (Circuit Switched Fallback), lo que quiere decir que pueda volver a tecnologías anteriores como redes UMTS, GSM mientras se operan en tecnologías superiores.
- Instalar los softwares en una sola maquina ayudaría a la optimización del proyecto, pero tendría que ser un computador con unas especificaciones muy elevadas.

BIBLIOGRAFÍA

[1] Vega L Andy, “Radio Definido por Software,” Universidad Nacional de Loja. Accessed: Dec. 16, 2024. [Online]. Available: https://telecomunicaciones.edu.ec/repositorio/articulos-blog/redes/radio-definido-por-software

[2] G. González Martínez, “Diseño de un sistema de acceso al medio y modulación en banda base utilizando radio definido por software para comunicación de largo alcance entre boyas,” Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California, 2016. Accessed: Dec. 16, 2024. [Online]. Available: https://cicese.repositorioinstitucional.mx/jspui/handle/1007/296

[3] K. Bhusal, “IMPLEMENTATION AND PERFORMANCE ANALYSIS OF LONG IMPLEMENTATION AND PERFORMANCE ANALYSIS OF LONG-TERM EVOLUTION USING SOFTWARE DEFINED RADIO TERM EVOLUTION USING SOFTWARE DEFINED RADIO”, Accessed: Dec. 16, 2024. [Online]. Available: http://hdl.handle.net/10950/609

[4] Alberto Castilla Gómez, “DESPLIEGUE Y EVALUACIÓN DE UNA RED CELULAR LTE MEDIANTE SOLUCIONES SDR,” UNIVERSIDAD DE CANTABRIA, Cantabria, 2020.

[5] Juan Garcia Cobo, “Estudio e implementación sobre plataforma SDR de una estación base para comunicaciones móviles de última generación,” Universidad de Jaén, Linares, 2022.

[6] Montaño Blacio Manuel Asdruval, “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN BASE TRANSMISORA – RECEPTORA (BTS) GSM PARA SERVICIO DE VOZ Y SMS BASADA EN YATEBTS,” Universidad Nacional de Loja, 2017.

[7] Enrique Rus Arias, “Investigación exploratoria.” Accessed: Dec. 16, 2024. [Online]. Available: https://economipedia.com/definiciones/investigacion-exploratoria.html

[8] Cristina Ortega, “Investigación aplicada: Definición, tipos y ejemplos.” Accessed: Dec. 16, 2024. [Online]. Available: https://www.questionpro.com/blog/es/investigacion-aplicada/

[9] J. R. L. Rebolledo, K. M. Zúñiga, and T. M. M. Parrales, “El avance tecnológico en la telefonía móvil,” *UNESUM - Ciencias. Revista Científica Multidisciplinaria*, vol. 6, no. 3, pp. 49–57, May 2022, doi: 10.47230/UNESUM-CIENCIAS.V6.N3.2022.417.

[10] F. A. P. J. and J. I. R. Z., “Estudio y análisis de los niveles de las densidades de potencia de la banda de GSM y LTE en el corregimiento de Bethania, provincia de Panamá,” *Revista Saberes APUDEP*, vol. 8, no. 1, pp. 105–124, Jan. 2025, doi: 10.48204/J.SABERES.V8N1.A6788.

[11] ELIO SAUL FUENTES GARCIA, “LA EVOLUCION DE LA TELEFONIA MOVIL EN BOLIVIA COCHABAMBA-BOLIVIA,” 2021, Accessed: Feb. 18, 2025. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/66223149/EVOLUCION_DE_LAS_TECNOLOGIAS_MOVILES_EN_BOLIVIA-libre.pdf?1618016746=&response-content-disposition=inline%3B+filename%3DLA_EVOLUCION_DE_LA_TELEFONIA_MOVIL_EN_BO.pdf&Expires=1739989174&Signature=VuXsWeYI-paV2VGLg4a608L3Rg6-0QxRvE-R0B7HQrIoNcaKCoIATTqVZ8K~P3yXgzbGuTkjRI68XI7xU2LOM~rGoTppNeZVWu58h3Yk60mMgZSXotViffBdegpkDb2D0T4RpTE9cvmjhLgKppGV7uQwgSJoujQXuKGaq3iEJW-9ly-efwPCNSoyUCBTA77j1MI9WDIwQy4~TYmcfzr1DjIBHEaDgGau7zoyInmmCfIqa0~i6w8A06GRFKhvmFmPMLRQO-tEvKpkJunhtyQsTaNqz8Aefckxtis6zkfyE2bDksJbWPxcRa7JRVS3LcDi84rBgOiYd8q8sSssLMuQmA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[12] J. Manuel and F. Loayza, “Gestión de videos basada en tecnología 3G para el control del transporte minero de la empresa Operaciones Mineras S. A.,” *Universidad Continental*, 2022, Accessed: Feb. 18, 2025. [Online]. Available: https://repositorio.continental.edu.pe/handle/20.500.12394/12806

[13] Javier Estalin Muñoz Cardenas, “PROTOTIPOS DE REDES LTE EN BASE A TECNOLOGÍA DE RADIO DEFINIDA POR SOFTWARE E INTEGRACIÓN DE LA TECNOLOGÍA DE RADIO

SOBRE FIBRA EN REDES MÓVILES," Oct. 2022, Accessed: Feb. 19, 2025. [Online]. Available: https://bibdigital.epn.edu.ec/bitstream/15000/23311/1/CD%2012727.pdf

[14] Pamela Leonar Guerrero Aliaga, "MPLEMENTACIÓN DE UN PROTOTIPO DE REALIDAD MIXTA XR MEDIANTE LA PLATAFORMA MICROSOFT HOLOLENS SOBRE UNA RED 4G LTE PARA CONTROLAR DISPOSITIVOS ELÉCTRICOS," 2024, Accessed: Feb. 19, 2025. [Online]. Available: http://dspace.espoch.edu.ec/bitstream/123456789/22971/1/98T00455.pdf

[15] Harald Remmert, "¿Qué es la arquitectura de red 5G?" Accessed: Feb. 19, 2025. [Online]. Available: https://es.digi.com/blog/post/5g-network-architecture

[16] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55765–55779, 2018, doi: 10.1109/ACCESS.2018.2872781.

[17] Vitor Sales, "6G: ¿Qué sabemos sobre esta nueva tecnología?" Accessed: May 06, 2025. [Online]. Available: https://www.sydle.com/es/blog/6-g-64f0ee77e85f4a4b92eb0649

[18] Adam Volle, "Long Term Evolution," *Frontiers in Communications and Networks*, vol. 1, Jan. 2025, doi: 10.3389/FRCMN.2020.00001/FULL.

[19] E. A. Vaca Rogel, "Diseño de una radiobase móvil con tecnología LTE ubicado en el barrio Atahualpa de la ciudad de Quito," 2021. Accessed: Feb. 20, 2025. [Online]. Available: http://dspace.ups.edu.ec/handle/123456789/20582

[20] Azar, "DESCRIPCIÓN GENERAL DEL EPS: ¿QUÉ ES EL EPS?" Accessed: Feb. 20, 2025. [Online]. Available: https://www-techtrained-com.translate.goog/overall-eps-overview-what-is-eps/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sge

[21] Calle Rivera Frank Jhonatan and Nazario Cubas Guillermo, "Despliegue de un nodo de gestión de movilidad LTE con openairinterface para interactuar con una red estándar real," Jul. 2024,

Accessed: Feb. 24, 2025. [Online]. Available: http://repositorio.unprg.edu.pe/handle/20.500.12893/13458

[22] Juan Garica Cobo, “Estudio e implementación sobre plataforma SDR de una estación base para comunicaciones móviles de última generación,” Feb. 2022, Accessed: Feb. 24, 2025. [Online]. Available: https://crea.ujaen.es/items/edc7dcd1-7633-4954-b995-739e8d003043

[23] J. Navarro-Ortiz, N. Chinchilla-Romero, F. Delgado-Ferro, J. J. Ramos-Munoz, and D. Saucedo Aranda, “Rendimiento de Redes 4G/5G usando una estación base real,” *JITEL 2021: XV Jornadas de Ingeniería Telemática: A Coruña, España: Octubre 27-29, 2021: libro de actas*, pp. 134–138, Oct. 2021, Accessed: Feb. 25, 2025. [Online]. Available: https://digibug.ugr.es/handle/10481/71140

[24] Sergio Fabian Garcia Minchala, “Diseño e implementación de un prototipo de Estación base (Nodo B) LTE, como herramienta experimental utilizando el SDR BladeRF,” 2023, Accessed: Feb. 24, 2025. [Online]. Available: https://dspace.unl.edu.ec/jspui/bitstream/123456789/28252/1/SergioFabian_GarciaMinchala.pdf

[25] Ladino Camargo Marlon, “Actualización, implementación y pruebas de red 4G LTE para empresa de telecomunicaciones móviles,” Jun. 2023, Accessed: Feb. 25, 2025. [Online]. Available: https://repository.udistrital.edu.co/server/api/core/bitstreams/d27531c5-c0e2-4949-ac9f-b09eb90fb0de/content

[26] A. D. Rodriguez Daza, “PROPUESTA PARA LA CONECTIVIDAD MÓVIL CELULAR EN ZONAS RURALES DE COLOMBIA EN REDES DE ACCESO 4G Y 5G.” Accessed: Mar. 09, 2025. [Online]. Available: https://repository.usta.edu.co/server/api/core/bitstreams/1b0339a0-5155-4061-b8aa-ea93c6865bf3/content

[27] Stephanie Burrell, “Comprensión de las bandas LTE FDD: una guía sencilla para todos.” Accessed: May 06, 2025. [Online]. Available: https://wraycastle.com/es/blogs/knowledge-base/lte-fdd-bands?srsltid=AfmBOoqjL_APUTLIbiFKwoh86d3hM7quNz9WoXaRUTv-EbFpS4kNciLa

[28] “B28 (700 MHz) | Portal de información de Powertec.” Accessed: Mar. 09, 2025. [Online]. Available: https://portal.powertec.com.au/technical-library/wireless/wireless-technologies/cellular/4g-lte/4g-frequency-bands/b28-700-mhz

[29] APAZA CALLO YURI RENEE, “ANÁLISIS DE LOS INDICADORES DE DESEMPEÑO DE UNA RED LTE-ADVANCED Y PROPUESTA DE DISEÑO PARA LA MEJORA DE LA CALIDAD DE SERVICIO EN LA CIUDAD DE SICUANI – CUSCO,” 2024, Accessed: Mar. 09, 2025. [Online]. Available: https://repositorio.unsaac.edu.pe/bitstream/handle/20.500.12918/8572/253T20240169_TC.pdf?sequence=1&isAllowed=y

[30] P. Mishel. Benítez Basantes, “Estudio de propagación de sistemas de telefonía móvil en la banda 1900 Mhz (LTE) mediante mediciones de campo eléctrico para establecer una recomendación de un modelo de propagación existente.,” 2022, Accessed: Mar. 10, 2025. [Online]. Available: http://dspace.espoch.edu.ec/handle/123456789/20764

[31] “LTE B4 (1700/2100 MHz AWS 1) Frequency - Carrier and Device Compatibility.” Accessed: Mar. 09, 2025. [Online]. Available: https://www.frequencycheck.com/bands/lte-band-4-aws-1#google_vignette

[32] “Banda 41 de 4G LTE - Redes 4G LTE.” Accessed: Mar. 09, 2025. [Online]. Available: https://www.4g-lte.net/about/lte-frequency-bands/lte-band-41/

[33] Tapuy Welligton, “4G LTE de Ecuador.” Accessed: Mar. 15, 2025. [Online]. Available: https://kupdf.net/download/4g-lte-de-ecuador_6630dd43e2b6f52964974e31_pdf

[34] Comunidad Huawei Enterprise, “HCIA-LTE ｜Canal físico LTE.” Accessed: Mar. 09, 2025. [Online]. Available: https://forum.huawei.com/enterprise/intl/es/thread/HCIA-LTE-%E4%B8%A8Canal-f%C3%ADsico-LTE/667228813064552448?blogId=667228813064552448

[35] Techlteworld, “In-depth Overview of Channels in LTE.” Accessed: Mar. 09, 2025. [Online]. Available: https://techlteworld.com/overview-of-channels-in-lte/

[36] Benavides Rivera Cristofer Jesus, “Predicción de comportamiento en tráfico de red LTE y ajuste de parametrización para maximizar performance de red,” 2021, Accessed: Mar. 09, 2025. [Online]. Available: https://repositorio.uchile.cl/handle/2250/180470

[37] Guerra Sanchez Sergio Andres, “Evaluación de técnicas de acceso al medio inalámbrico basadas en aprendizaje por refuerzo para IoT,” *Ad Hoc Networks*, vol. 106, Dec. 2023, doi: 10.1016/j.adhoc.2020.102200.

[38] J. M. Muñoz Rodríguez, “Optimización de una red LTE en la provincia de Santa Elena mediante la herramienta de simulación Atoll para el estudio de las estrategias de Packet Scheduling: Round Robin y Proportional Fair,” Jan. 2024, Accessed: Mar. 10, 2025. [Online]. Available: https://repositorio.upse.edu.ec/handle/46000/10709

[39] Figueroa Rodriguez George Michael, “Optimización de una red LTE de la provincia de Santa Elena mediante el software ATOLL utilizando estrategias Máximum C/I MAX C/I y Proportional Demand PD de Packet Scheduling,” May 2023, Accessed: Feb. 24, 2025. [Online]. Available: https://repositorio.upse.edu.ec/handle/46000/9621

[40] N. N. Sirhan and M. Martinez-Ramon, “LTE Cellular Networks Packet Scheduling Algorithms in Downlink and Uplink Transmission, A Survey,” *International Journal of Wireless & Mobile Networks*, vol. 14, no. 2, pp. 1–15, Apr. 2022, doi: 10.5121/ijwmn.2022.14201.

[41] Alberto Castilla Gomez, “DESPLIEGUE Y EVALUACIÓN DE UNA RED CELULAR LTE MEDIANTE SOLUCIONES SDR,” Jul. 2020, Accessed: Feb. 24, 2025. [Online]. Available: https://repositorio.unican.es/xmlui/bitstream/handle/10902/19014/425930.pdf?sequence=1&isAllowed=y

[42] Wilson, “LTE Authentication Introduction: EPS-AKA’.” Accessed: Feb. 25, 2025. [Online]. Available: https://free5gc.org/blog/20231018/20231018/#introduction

[43] Rinku Yadav, “Modelo de seguridad de la red LTE (4G) y sus inconvenientes contemplados en la red 5G.” Accessed: Feb. 25, 2025. [Online]. Available: https://www.copperpodip.com/post/lte-4g-network-security-model-and-its-drawbacks-covered-in-5g-network

[44] Moazzam Tiwana, “¿Cuál es la diferencia entre la señalización de estrato de acceso (AS) y estrato sin acceso en 5G?” Accessed: Feb. 25, 2025. [Online]. Available: https://drmoazzam.com/what-is-difference-between-access-stratum-as-and-non-access-stratum-signalling-in-5g

[45] Carolina Gonzalez Valenzuela, “Esta es la historia de los teléfonos móviles: desde su origen hasta la actualidad.” Accessed: Feb. 26, 2025. [Online]. Available: https://computerhoy.20minutos.es/moviles/historia-telefonos-moviles-origen-actualidad-1181484

[46] Bronwyn Hemus, “SIM, eSIM vs iSIM: What’s the Difference?” Accessed: Feb. 26, 2025. [Online]. Available: https://www-emnify-com.translate.goog/blog/sim-vs-esim-vs-isim?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sge#sim

[47] Thales Group, “Lectores de tarjetas inteligentes.” Accessed: Feb. 26, 2025. [Online]. Available: https://cpl.thalesgroup.com/es/access-management/smart-card-readers

[48] “Amazon.com: MCR3516 Lector SIM Escritor Mini Nano Micro 2FF, 3FF, 4FF Tarjeta SIM+5pcs Programable Blanco LTE USIM 4G Tarjeta WCDMA GSM + Sim Personalizar Software :

Celulares y Accesorios.” Accessed: Apr. 26, 2025. [Online]. Available: https://www.amazon.com/-/es/MCR3516-Escritor-Programable-Personalizar-Software/dp/B07PVSPLLH

[49] C. F. ALDAZ CORRALES, “ESTUDIO DE LA TECNOLOGÍA SDR (SOFTWARE DEFINED RADIO) Y POSIBLES APLICACIONES EN COMUNICACIONES INALÁMBRICAS,” ESCUELA POLITECNICA NACIONAL, Quito, 2009. Accessed: Feb. 13, 2025. [Online]. Available: https://bibdigital.epn.edu.ec/bitstream/15000/1429/1/CD-2092.pdf

[50] T. Ulversøy, “Software Defined Radio: Challenges and Opportunities”, doi: 10.1109/SURV.2010.032910.00019.

[51] Nuand, “bladeRF 2.0 micro xA4 - Nuand.” Accessed: Jun. 29, 2025. [Online]. Available: https://www.nuand.com/product/bladerf-xa4/

[52] Nuand, “bladeRF x40 - Nuand.” Accessed: Jun. 29, 2025. [Online]. Available: https://www.nuand.com/product/bladerf-x40/

[53] Nuand, “bladeRF 2.0 micro xA9 - Nuand.” Accessed: Jun. 29, 2025. [Online]. Available: https://www.nuand.com/product/bladerf-xa9/

[54] Nuand, “bladeRF 2.0 micro xA5.” Accessed: Jun. 29, 2025. [Online]. Available: https://www.nuand.com/product/bladerf-xa5/

[55] Ettus Research, “USRP Software Defined Radio (SDR) online catalog -.” Accessed: Jun. 29, 2025. [Online]. Available: https://www.ettus.com/products/

[56] RedHat, “¿Qué es el open-source o código abierto?” Accessed: Jun. 29, 2025. [Online]. Available: https://www.redhat.com/es/topics/open-source/what-is-open-source

[57] OAI, “OpenAirInterface – 5G software alliance for democratising wireless innovation.” Accessed: Jun. 29, 2025. [Online]. Available: https://openairinterface.org/

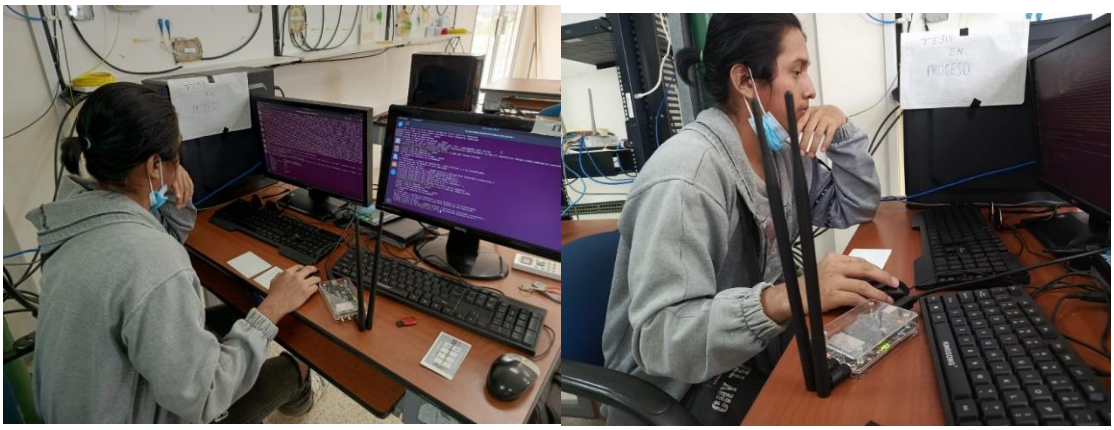
[58] srsRAN, “4g - srsRAN 4G.” Accessed: Jun. 29, 2025. [Online]. Available: https://www.srslte.com/4g

[59] Open5GS, “Open5GS.” Accessed: Apr. 27, 2025. [Online]. Available: https://open5gs.org/

[60] Concepto.de, “Windows -Concepto, para qué sirve y listado de versiones.” Accessed: Jun. 29, 2025. [Online]. Available: https://concepto.de/windows-2/

[61] Ubuntu, “Download Ubuntu DesktoP.” Accessed: Jun. 29, 2025. [Online]. Available: https://ubuntu.com/download/desktop

ANEXOS



Anexo 1 Pruebas de Funcionamiento de la Arquitectura Simulada

Anexo 2 Pruebas de potencia con el Equipo NARDA

Anexo 3 Revisión Equipos necesarios para la implementación

Anexo 4 Pruebas de Funcionamiento de las Antenas

Tribanda por medio del VNA

Anexo 5 Simulación y Explicación de la Arquitectura Simulada