



Universidad
de Jaén

Escuela Politécnica Superior
de Linares

Trabajo Fin de Grado

HERRAMIENTA DOCENTE PARA LA ENSEÑANZA DE REDES DE ÚLTIMA GENERACIÓN

Alumno: Juan Luis Herreros Bódalo

Tutores: Prof. D. Damián Martínez Muñoz
Prof. D. Antonio Jesús Yuste Delgado

Depto.: Ingeniería de Telecomunicación

Junio, 2024

ÍNDICE

RESUMEN.....	11
ABSTRACT	12
GLOSARIO DE TÉRMINOS.....	13
1. INTRODUCCIÓN	17
2. OBJETIVOS	18
3. ESTADO DEL ARTE	19
3.1. EVOLUCIÓN DE LOS SISTEMAS DE TELEFONÍA MÓVIL	19
3.1.1. 1G.....	19
3.1.2. 2G.....	20
3.1.3. 3G.....	20
3.1.4. 4G.....	22
3.2. 5G	23
3.2.1. ARQUITECTURA DE RED	26
3.2.2. NÚCLEO DE RED.....	27
3.2.3. PILAS DE PROTOCOLOS 5G.....	29
3.2.3.1. PLANO DE CONTROL: LA PILA DE PROTOCOLOS UE – AMF Y UE – SMF	29
3.2.3.2. PLANO DE USUARIO: LA PILA DE PROTOCOLOS UE – AMF Y UE – SMF	30
3.2.4. RED DE ACCESO.....	30
3.2.4.1. ARQUITECTURA “NON-STAND ALONE” (NSA)	31
3.2.4.2. ARQUITECTURA “STAND ALONE” (SA)	32
3.2.4.3. ARQUITECTURA GENERAL DE PROTOCOLOS DE LA INTERFAZ RADIO.....	34
3.2.4.4. CAPA FÍSICA	34
3.2.4.5. ACCESO MÚLTIPLE	36
3.2.4.6. ASIGNACIÓN DE RECURSOS Y VELOCIDAD DE BITS.....	38
3.2.4.7. BANDAS DE OPERACIÓN	38
3.3. IMPLEMENTACIONES DE 5G	40
3.4. RADIO DEFINIDA POR SOFTWARE (SDR)	46
4. SOLUCIÓN ADOPTADA.....	49
4.1. DOCKER_OPEN5GS.....	52
4.1.1. GITHUB WORKFLOWS.....	53
4.1.2. PORTAINER	56
4.1.2.1. INSTALACIÓN	56

4.1.2.2.	CONFIGURACIÓN INICIAL	57
4.1.2.3.	USO.....	59
4.2.	HERRAMIENTAS PARA LA REALIZACIÓN DE PRÁCTICAS CON LA INTERFAZ RADIO	60
4.2.1.	ANÁLISIS DE LA INTERFAZ RADIO.....	60
4.3.	HERRAMIENTAS PARA LA REALIZACIÓN DE PRÁCTICAS CON EL NÚCLEO DE RED	64
4.3.1.	ANÁLISIS DE RED.....	64
4.3.2.	SIMULACIÓN DE LA INTERFAZ RADIO	67
4.3.2.1.	PLUGIN PARA WIRESHARK.....	70
5.	RESULTADOS.....	71
5.1.	RENDIMIENTO DE LA RED.....	71
5.2.	GUIONES DE PRÁCTICAS	75
5.2.1.	PRÁCTICA RELACIONADA CON LA INTERFAZ RADIO.....	75
5.2.1.1.	SELECCIÓN DE BANDAS DE FRECUENCIA.	75
5.2.1.2.	ANÁLISIS DEL ESPECTRO.....	76
5.2.1.3.	CONFIGURACIÓN DE LA ESTACIÓN BASE.	76
5.2.1.4.	IMPLEMENTACIÓN Y PRUEBAS	76
5.2.1.5.	CUESTIÓN TEÓRICA.	78
5.2.2.	PRÁCTICA RELACIONADA CON EL NÚCLEO DE RED.....	82
6.	CONCLUSIONES.....	84
7.	ANEXOS.....	85
7.1.	ANEXO I: PUESTA EN MARCHA Y CONFIGURACIÓN INICIAL DEL SISTEMA 5G IMPLEMENTADO.....	85
7.1.1.	REQUISITOS PREVIOS	85
7.1.2.	INSTALACIÓN DE DOCKER	86
7.1.2.1.	UBUNTU	86
7.1.2.2.	WINDOWS	87
7.1.3.	DESCARGA DEL REPOSITORIO DOCKER_OPEN5GS.....	88
7.1.4.	CONFIGURACIÓN BÁSICA DEL SISTEMA 5G SA CON IMS.....	89
7.1.4.1.	CONFIGURACIÓN CON UN ÚNICO EQUIPO.	89
7.1.4.2.	CONFIGURACIÓN CON VARIOS EQUIPOS.....	90
7.1.4.3.	CONFIGURACIÓN DE LA INTERFAZ RADIO CON SRSRAN	90
7.1.5.	DESPLIEGUE DEL SISTEMA 5G SA CON IMS	91
7.1.6.	APROVISIONAMIENTO DE ABONADOS AL 5GC E IMS.....	92
7.1.6.1.	APROVISIONAMIENTO AUTOMÁTICO DE ABONADOS	92
7.1.6.2.	APROVISIONAMIENTO MANUAL DE ABONADOS	93
7.1.7.	PROGRAMACIÓN DE LAS TARJETAS SIM.....	97
7.2.	ANEXO II: ARCHIVOS DE CONFIGURACIÓN DEL SISTEMA 5G	101

7.2.1.	<i>.ENV</i>	101
7.2.2.	<i>NAMED.CONF</i>	104
7.2.3.	<i>GNB.YML</i>	105
7.2.4.	<i>QOS.YML</i>	106
7.3.	ANEXO III: SCRIPTS ELABORADOS PARA LA CONFIGURACIÓN DEL SISTEMA	110
7.3.1.	<i>SIM-PROGRAM</i>	110
7.3.2.	<i>OPEN5GS-DBCONF</i>	112
7.3.3.	<i>PYHSS-DBCONF</i>	117
7.3.4.	<i>DBCONF-ALL.SH</i>	122
8.	BIBLIOGRAFÍA	123

ÍNDICE DE FIGURAS

Figura 1: Línea temporal de los distintos estándares de telefonía móvil.....	23
Figura 2: Mejora de las capacidades fundamentales de las IMT-Avanzadas a las IMT-2020. Figura M.2083-03 de la Recomendación de la UIT-R M.2083-0.	24
Figura 3: Importancia de las capacidades fundamentales para distintos casos de utilización. Figura M.2083-04 de la Recomendación de la UIT-R M.2083-0.	25
Figura 4: Línea temporal que muestra la evolución del 5G (15).....	26
Figura 5: Diagrama simplificado de la arquitectura de red de 5G (16).	26
Figura 6: Pila de protocolos del plano de control entre UE, 5G-AN, AMF y SMF. TS 23.501, sección 8.2.....	29
Figura 7: Pila de protocolos del plano de usuario entre el UE, el 5G-AN y la UPF. TS 23.501, sección 8.3.....	30
Figura 8: La arquitectura NSA de 5G.....	32
Figura 9: La arquitectura SA de 5G.	33
Figura 10: Arquitectura del protocolo de interfaz radio alrededor de la capa física. Figura extraída de la especificación técnica 38.201 del 3GPP.	34
Figura 11: Logotipo de OpenAirInterface.....	40
Figura 12: Esquema de la Red de Acceso Radio (RAN) implementada por la plataforma OpenAirInterface.....	40
Figura 13: Implementación actual del Núcleo de Red o Red Central (CN) de 5G en OpenAirInterface.....	41
Figura 14: Logotipo de Open5GS.....	41
Figura 15: Esquema general de la Arquitectura 5G en el que se diferencia por colores las distintas implementaciones de Open5gs. En verde se muestran los componentes y las conexiones propias del núcleo 4G/5G NSA, mientras que en rosa se muestran los propios de 5G SA.	42
Figura 16: Logotipo de free5GC.	43
Figura 17: Logotipo de UERANSIM.....	44
Figura 18: Logotipo de srsRAN Project.	45
Figura 19: SDR USRP B210.....	46
Figura 20: SDR USRP X310.....	46
Figura 21: SDR USRP N310	47

Figura 22: SDR LimeSDR.....	47
Figura 23: SDR RTL-SDR.	47
Figura 24: SDR HackRF One.	48
Figura 25: SDR bladeRF.....	48
Figura 26: Logotipo de GitHub.....	53
Figura 27: Ventana “Choose a workflow” de GitHub.	53
Figura 28: Edición de un workflow en GitHub.....	54
Figura 29: Opciones de seguridad del repositorio en GitHub.....	55
Figura 30: Instalación de Portainer como una extensión de Docker Desktop.	57
Figura 31: Configuración inicial de Portainer, creación de usuario administrador.	58
Figura 32: Configuración inicial de Portainer, 'Environment Wizard'.	58
Figura 33: Pantalla principal de Portainer, selección de entorno Docker.	59
Figura 34: Información general sobre el entorno Docker al que nos hemos conectado con Portainer.	59
Figura 35: Información de los contenedores que existen en el entorno Docker.....	60
Figura 36: Logotipo de Grafana.....	61
Figura 37: Panel principal de Grafana donde se muestran las métricas ofrecidas por srsRAN en tiempo real.....	64
Figura 38: Logotipo de Edgeshark.....	64
Figura 39: Logotipo de Ghostwire.....	64
Figura 40: Visualización de conexiones de red con la herramienta Ghostwire.	65
Figura 41: Logotipo del plugin cshargextcap.	66
Figura 42: Detalle de la ventana de diálogo generada por la herramienta Ghostwire que abre la aplicación Wireshark para capturar el tráfico de un contenedor Docker.	66
Figura 43: Pasos a seguir para capturar el tráfico de un contenedor Docker directamente desde Wireshark gracias al plugin cshargextcap.	67
Figura 44: Log del gNB simulado con UERANSIM donde se muestra que está registrado en la red y que un UE se ha registrado en él.	68
Figura 45: Log del UE simulado con UERANSIM donde se muestra información acerca del intercambio de mensajes con el gNB hasta que se autentica correctamente y crea una conexión a la red proporcionada por el sistema 5G.	69

Figura 46: Traceroute a “google.es” ejecutado a través de la interfaz “uesimtun0” en el UE simulado por UERANSIM.	69
Figura 47: Traceroute a 'google.es' ejecutado en el UE simulado por UERANSIM.	70
Figura 48: Arquitectura del sistema 5G desplegado.....	71
Figura 49: Test de velocidad realizado en el terminal móvil Xiaomi Redmi 12 5G.	73
Figura 50: Panel de Grafana donde se muestran las métricas del gNodeB relacionadas con el número de UE, uso de MIMO e índices de calidad de las conexiones descendente y ascendente	74
Figura 51: Panel de Grafana donde se muestran las métricas del gNodeB relacionadas con el número de paquetes enviados en los enlaces ascendente y descendente.....	74
Figura 52: Espectro de los enlaces ascendente y descendente durante la prueba de velocidad.....	74
Figura 53: Práctica relacionada con el interfaz radio. Resultados prueba de velocidad en el terminal móvil.....	79
Figura 54: Práctica relacionada con el interfaz radio. Estadísticas del gNodeB relacionadas con el número de UE, uso de MIMO e índices de calidad de las conexiones descendente y ascendente	79
Figura 55: Práctica relacionada con el interfaz radio. Estadísticas del gNodeB relacionadas con el número de paquetes enviados en los enlaces ascendente y descendente.....	80
Figura 56: Práctica relacionada con el núcleo de red. Elementos que componen el sistema 5G desplegado por el software “docker_open5gs”	82
Figura 57: Interfaz de usuario de Docker Desktop.	88
Figura 58: Detalle de los contenedores ejecutándose en Docker visto desde la aplicación Docker Desktop en Windows.....	88
Figura 59: Interfaz para añadir abonados a la base de datos de Open5GS	94
Figura 60: Datos de un abonado a la base de datos de Open5GS.....	95

ÍNDICE DE TABLAS

Tabla 1: Número de símbolos OFDM por intervalo, intervalos por trama e intervalos por subtrama. Tablas 4.3.2-1 y 4.3.2-2 de la especificación técnica 38.211 del 3GPP.....	37
Tabla 2: Numerologías de transmisión soportadas. Tabla 4.2-1 de la especificación técnica 38.211 del 3GPP.	38
Tabla 3: Definición de los rangos de frecuencia. Tabla 5.1-1 de la especificación técnica 38.104 del 3GPP	39
Tabla 4: Número de bloques de recursos para FR1. Tabla 5.3.2-1 de la especificación técnica 38.104 del 3GPP.	39
Tabla 5: Número de bloques de recursos para FR2-1. Tabla 5.3.2-2 de la especificación técnica 38.104 del 3GPP.	39
Tabla 6: Número de bloques de recursos para FR2-2. Tabla 5.3.2-3 de la especificación técnica 38.104 del 3GPP.	39
Tabla 7: Características de los terminales móviles Xiaomi Redmi 12 5G y Motorola moto g73 5G.....	50
Tabla 8: Bandas de frecuencia de NR compatibles con los terminales móviles Xiaomi Redmi 12 5G y Motorola Moto g73 5G.	51
Tabla 9: Configuración APN en los terminales móviles.	52
Tabla 10: Tabla 5.1.3.1-3 de la especificación técnica 38.214 del 3GPP.....	81
Tabla 11: Plataformas compatibles con Docker (40).	85
Tabla 12: Compatibilidad de componentes con los distintos sistemas operativos.	86
Tabla 13: Abonados al sistema 5G	93
Tabla 14: Configuración APN de cada abonado en la base de datos de Open5GS.....	94
Tabla 15: Detalle del número IMSI.....	100

RESUMEN

El presente Trabajo Fin de Grado tiene como objetivo principal el diseño y desarrollo de prácticas de laboratorio enfocadas al estudio de la interfaz radio y núcleo de red de un sistema 5G. Para ello, se parte de un sistema 5G previamente implementado en Docker.

El desarrollo de estas prácticas tiene una doble finalidad: por un lado, proporcionar a los estudiantes una herramienta práctica y didáctica que les permita comprender y experimentar con las tecnologías subyacentes en la red 5G, y por otro lado, contribuir al fortalecimiento del conocimiento y las capacidades técnicas en el ámbito de las telecomunicaciones.

El sistema 5G utilizado como base para estas prácticas está compuesto por una serie de contenedores Docker que simulan las distintas funciones de la red, tanto en la interfaz radio como en el núcleo de la red. A lo largo de este trabajo, se detallan los pasos necesarios para la configuración y despliegue de estos contenedores, así como las metodologías y herramientas empleadas para la realización de las prácticas.

Las prácticas diseñadas abarcan una amplia gama de temas, desde la configuración de la red de acceso radio (RAN), hasta la implementación y pruebas de funciones de red centrales (5GC), pasando por la monitorización del rendimiento y análisis de tráfico. Además, se incluyen guías detalladas y materiales complementarios para facilitar la comprensión y ejecución de las mismas por parte de los estudiantes.

Finalmente, se presentan los resultados obtenidos durante el desarrollo y prueba de las herramientas de uso docente, así como una evaluación de su efectividad y posibles mejoras futuras. Este trabajo supone una contribución significativa al ámbito educativo en telecomunicaciones, proporcionando un recurso valioso para la formación práctica en tecnologías de última generación como el 5G.

ABSTRACT

The main objective of this Project is the design and development of laboratory practices focused on the study of the radio interface and network core of a 5G system. For this, we start from a 5G system previously implemented in Docker.

The development of these practices has a dual purpose: on the one hand, to provide students with a practical and didactic tool that allows them to understand and experiment with the underlying technologies in the 5G network, and on the other hand, to contribute to the strengthening of knowledge and technical skills in the field of telecommunications.

The 5G system used as a basis for these practices is composed of a series of Docker containers that simulate the different functions of the network, both in the radio interface and in the core of the network. Throughout this work, the necessary steps for the configuration and deployment of these containers are detailed, as well as the methodologies and tools used for the realization of the practices.

The designed practices cover a wide range of topics, from radio access network (RAN) configuration, to core network functions (5GC) implementation and testing, to performance monitoring and traffic analysis. In addition, detailed guides and supplementary materials are included to facilitate student understanding and execution.

Finally, the results obtained during the development and testing of the teaching tools are presented, as well as an evaluation of their effectiveness and possible future improvements. This work is a significant contribution to the telecommunications education field, providing a valuable resource for practical training in next-generation technologies such as 5G.

GLOSARIO DE TÉRMINOS

3GPP	<i>3rd Generation Partnership Project</i>	(Proyecto de Asociación de Tercera Generación)
3GPP2	<i>3rd Generation Partnership Project 2</i>	(Proyecto de Asociación de Tercera Generación 2)
5G GUTI	<i>5G Globally Unique Temporary Identity</i>	(Identidad Temporal Global Única de 5G)
5G NR	<i>New Radio</i>	
5GC	<i>5G Core</i>	(Núcleo de Red 5G)
AF	<i>Application Function</i>	(Función de Aplicación)
AMF	<i>Access and Mobility Management Function</i>	(Función de Gestión de Acceso y Movilidad)
AMPS	<i>Advanced Mobile Phone System</i>	(Sistema Telefónico Móvil Avanzado)
AN	<i>Access Network</i>	(Red de Acceso)
API	<i>Application Programming Interface</i>	Interfaz de Programación de Aplicaciones
CDMA	<i>Code Division Multiple Access</i>	(Acceso Múltiple por División de Código)
CN	<i>Core Network</i>	(Núcleo de Red)
CP	<i>Control Plane</i>	(Plano de Control)
CP	<i>Cyclic Prefix</i>	(Prefijo Cíclico)
CP-OFDM	<i>Cyclic Prefix-Orthogonal Frequency Division Multiplexing</i>	(OFDM con Prefijo Cíclico)
CUPS	<i>Control & User Plane Separation</i>	(Separación de los Planos de Control y Usuario)
DCI	<i>Downlink Control Information</i>	(Información de Control en el enlace Descendente)
DECT	<i>Digital Enhanced Cordless Telecommunications</i>	(Telecomunicaciones Inalámbricas Mejoradas Digitalmente)
DFT-S-OFDM	<i>Discrete Fourier Transform-Spread-Orthogonal Frequency Division Multiplexing</i>	(OFDM con precodificación de Transformada Discreta de Fourier)
DL	<i>Downlink</i>	(Enlace Descendente)

EDGE	<i>Enhanced Data Rates for GSM Evolution</i>	(Tasas de Datos Mejoradas para la Evolución del GSM)
EMBB	<i>enhanced Mobile Broadband</i>	Banda Ancha Móvil mejorada
EPC	<i>Evolved Packet Core</i>	(Núcleo de Paquete Evolucionado)
ETACS	<i>Extended Total Access Communications System</i>	(Sistema de Comunicación Totalmente Automático Extendido)
ETSI	<i>European Telecommunications Standards Institute</i>	(Instituto Europeo de Normas de Telecomunicaciones)
FDD	<i>Frequency division duplexing</i>	Duplexación por División en Frecuencia
FQDN	<i>Fully Qualified Domain Name</i>	Nombre de Dominio Completo
GPRS	<i>General Packet Radio Service</i>	(Servicio General de Paquetes vía Radio)
GSM	<i>Global System for Mobile communications</i>	(Sistema Global para las comunicaciones Móviles)
GTP	<i>GPRS Tunneling Protocol</i>	(Protocolo de Túnel de GPRS)
HSDPA	<i>High Speed Downlink Packet Access</i>	(Acceso Descendente de Paquetes de Datos a Alta Velocidad)
HSPA+	<i>High Speed Packet Access Evolution</i>	(Acceso de Paquetes de Datos a Alta Velocidad Evolucionado)
HSUPA	<i>High Speed Uplink Packet Access</i>	(Acceso Ascendente de Paquetes de Datos a Alta Velocidad)
IMEI	<i>International Mobile Equipment Identity</i>	(Identidad Internacional del Equipo Móvil)
IMS	<i>IP Multimedia Subsystem</i>	(Subsistema Multimedia IP)
IMSI	<i>International Mobile Subscriber Identity</i>	Identidad Internacional del Abonado Móvil
IMT-2000	<i>International Mobile Telecommunications 2000</i>	(Telecomunicaciones Móviles Internacionales 2000)
ITU / UIT	<i>International Telecommunication Union</i>	(Unión Internacional de Telecomunicaciones)
LDPC	<i>Low Density Parity Check</i>	(Comprobación de Paridad de Baja Densidad)
LTE	<i>Long Term Evolution</i>	(Evolución a Largo Plazo)
MCS-L1	<i>Mobile Communication System - Layer 1</i>	(Sistema de Comunicación Móvil - Capa 1)

MMS	<i>Multimedia Messaging Service</i>	(Servicio de Mensajería Multimedia)
MMTC	<i>massive Machine-Type Communications</i>	(Comunicaciones masiva de Tipo Máquina)
NEF	<i>Network Exposure Function</i>	(Función de Exposición de Red)
NF	<i>Network Function</i>	(Función de Red)
NMT	<i>Nordic Mobile Telephony</i>	(Telefonía Móvil Nórdica)
NRF	<i>Network function Repository Function</i>	(Función de Repositorio de Red)
NSA	<i>(5G) Non-Stand Alone</i>	((5G) no independiente)
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>	(Multiplexación por División de Frecuencias Ortogonales)
OMV	<i>Mobile Virtual Network Operator (MVNO)</i>	(Operador Móvil Virtual)
PAPR	<i>Peak-to-Average Power Ratio</i>	(Relación de Potencia Pico a Promedio)
PCF	<i>Policy Control Function</i>	(Función de Control de Políticas)
PFCP	<i>Packet Forwarding Control Protocol</i>	(Protocolo de Control de Reenvío de Paquetes)
PRB	<i>Physical Resource Block</i>	(Bloque Físico de Recursos)
PSK	<i>Phase Shift Keying</i>	(Modulación por Desplazamiento de Fase)
QAM	<i>Quadrature Amplitude Modulation</i>	(Modulación de Amplitud de Cuadratura)
QOS	<i>Quality of Service</i>	Calidad de Servicio
RAN	<i>Radio Access Network</i>	(Interfaz de Acceso Radio)
S-NSSAI	<i>Single-Network Slice Selection Assistance Information</i>	(Información de Asistencias para la Selección de Segmentos de Red Única)
SA	<i>(5G) Stand Alone</i>	((5G) independiente)
SD	<i>Service Differentiator</i>	(Diferenciador de Segmento)
SDR	<i>Software Designed Radio</i>	(Radio Definida por Software)
SIM	<i>Subscriber Identity Module</i>	(Módulo de Identidad del Suscriptor)
SMF	<i>Session Management Function</i>	(Función de Gestión de Sesiones)
SMS	<i>Short Message Service</i>	(Servicio de Mensajes Cortos)
SST	<i>Slice Service Type</i>	(Tipo de Segmento/Servicio)

SUCI	<i>Subscription Concealed Identifier</i>	(Identificador Oculto de Suscripción)
SUPI	<i>Subscription Permanent Identifier</i>	(Identificador Permanente de Suscripción)
TACS	<i>Total Access Communication System</i>	(Sistema de Comunicación Totalmente Automático)
TD-SCDMA	<i>Time-Division Synchronous Code-Division Multiple Access</i>	(Acceso Múltiple por División de Código Síncrono de División de Tiempo)
TDD	<i>Time-Division Duplexing</i>	Duplexación por División en Tiempo
TDMA	<i>Time Division Multiple Access</i>	(Acceso Múltiple por División de Tiempo)
TMA		(Telefonía Móvil Automática)
TMSI	<i>Temporary Mobile Subscriber Identity</i>	(Identidad Temporal del Suscriptor Móvil)
UCI	<i>Uplink Control Information</i>	(Información de Control en el enlace Ascendente)
UDM	<i>Unified Data Management</i>	(Gestión de Datos Unificada)
UDR	<i>Unified Data Repository</i>	(Repositorio de Datos Unificado)
UE	<i>User Equipment</i>	(Equipo de Usuario)
UL	<i>Uplink</i>	(Enlace Ascendente)
UMTS	<i>Universal Mobile Telecommunications System</i>	(Sistema Universal de Telecomunicaciones Móviles)
UP	<i>User Plane</i>	(Plano de Usuario)
UPF	<i>User Plane Function</i>	(Función del Plano de Usuario)
URLLC	<i>Ultra-High Reliability & Low Latency</i>	(Comunicaciones de Gran Fiabilidad y Baja Latencia)
UTRAN	<i>UMTS Terrestrial Radio Access Network</i>	(Red de Acceso Radio Terrestre UMTS)
WCDMA	<i>Wideband Code Division Multiple Access</i>	(Acceso Múltiple por División de Código de Banda Ancha)

1. INTRODUCCIÓN

La evolución de las tecnologías de comunicación móvil ha sido vertiginosa en las últimas décadas, con cada generación de redes móviles superando a la anterior en términos de velocidad, capacidad y funcionalidad. La quinta generación de redes móviles (5G) representa un salto cualitativo significativo, ofreciendo mejoras notables en el rendimiento de la red, la reducción de la latencia y la capacidad para soportar una densidad de dispositivos mucho mayor. Estos avances no solo prometen mejorar la experiencia del usuario final, sino que también habilitan una nueva gama de aplicaciones y servicios en sectores como la salud, la industria y el entretenimiento.

El presente Trabajo Fin de Grado se enfoca en el diseño y desarrollo de prácticas de laboratorio que permitan a los estudiantes explorar y comprender en profundidad la RAN (Interfaz de Acceso Radio) y el 5GC (Núcleo de Red 5G) de un sistema 5G. Partiendo de un sistema 5G implementado en Docker, se busca proporcionar un conjunto de prácticas que no solo sean técnicamente rigurosas, sino también accesibles y didácticas.

Las prácticas diseñadas abordarán aspectos cruciales de la red 5G, desde la configuración y gestión de la RAN, hasta la implementación y pruebas de las funciones de la red. Además, se considerarán herramientas de monitorización y análisis de tráfico para ofrecer una visión completa y práctica del funcionamiento de la red 5G.

Este trabajo se estructura de la siguiente manera: en primer lugar, se presenta una revisión del estado del arte en sistemas de telefonía móvil y las distintas implementaciones de sistemas 5G de código abierto. A continuación, se describe la metodología empleada para el diseño y desarrollo de las prácticas de laboratorio. Posteriormente, se detallan las prácticas diseñadas y, finalmente, se discuten las conclusiones proponiendo posibles líneas de mejora y futuros desarrollos.

Con este trabajo, se espera contribuir al fortalecimiento de la formación práctica en telecomunicaciones, proporcionando a los estudiantes de la Universidad de Jaén una herramienta valiosa para su aprendizaje y preparación profesional en el ámbito de las tecnologías de redes móviles de última generación.

2. OBJETIVOS

Partiendo de una implementación basada en Docker de un sistema 5G sobre plataforma SDR (Radio Definida por Software) se desplegará una red 5G SA (Stand Alone) que contará con un IMS (Subsistema Multimedia IP) para ofrecer el servicio de SMS (Servicio de Mensajes Cortos). A este despliegue se podrán conectar distintos dispositivos móviles, cada uno equipado con una tarjeta SIM (Módulo de Identidad del Suscriptor) que posibilitará su registro en la red.

Una vez hayamos alcanzado los objetivos mencionados anteriormente, se diseñarán y desarrollarán distintas prácticas de laboratorio enfocadas al estudio tanto de la interfaz radio como del núcleo de red.

Como material adicional a las prácticas de laboratorio, se desarrollarán distintos documentos relacionados con la instalación, puesta en marcha y configuración del sistema 5G al completo (5GC + IMS + RAN).

3. ESTADO DEL ARTE

3.1. Evolución de los sistemas de telefonía móvil

3.1.1. 1G

Esta generación de telefonía móvil se caracterizaba por soportar únicamente llamadas de voz y ser analógica, la tecnología se basaba en el uso de un conjunto de celdas o células interconectadas que daban servicio a los dispositivos que se encontraban dentro de su amplia zona de cobertura (1).

En 1977, Bell Laboratories construyó la primera red de telefonía móvil alrededor de Chicago y estuvo realizando pruebas con ella en 1978, aunque la primera red comercial fue lanzada en Japón por Nippon Telegraph and Telephone Corporation (NTT) en 1979.

Tras ellos, los países nórdicos diseñaron juntos el estándar NMT (Telefonía Móvil Nórdica), lanzado en primer lugar en Suecia en 1981. NMT fue la primera red de telefonía móvil que incluyó *roaming* internacional (2).

En España, se introdujo la telefonía móvil de primera generación en 1982 con la telefonía móvil automática (TMA), una tecnología que tomó como referencia el modelo nórdico NMT en la modalidad de 900 MHz. Al principio se desplegó la red TMA-450 que solo estaba disponible en grandes ciudades y operaba a la frecuencia de 450 MHz. En 1989, se introdujo la red evolucionada TMA-900 que extendió la cobertura a más zonas y permitió reducir el tamaño de las células, con la consiguiente reducción de la potencia radiada, la longitud de onda y el tamaño de los componentes al operar a 900 MHz (3).

Mientras tanto, Bell Laboratories desarrolló el AMPS (Sistema Telefónico Móvil Avanzado), otro sistema de telefonía móvil que se implementó en 1983 en Estados Unidos y que se llegó a implementar también en Inglaterra y Japón con los nombres TACS (Sistema de Comunicación Totalmente Automático) y MCS-L1 (Sistema de Comunicación Móvil - Capa 1) respectivamente (4). Una variante de esta tecnología, la ETACS (Sistema de Comunicación Totalmente Automático Extendido), llegó a España en 1993 con el lanzamiento de MoviLine, la primera y única red 1G del país (3).

La tecnología ETACS utilizaba también la banda de frecuencia de 900 MHz, la cual disponía de 1320 canales dúplex de 25 kHz de ancho de banda para voz analógica.

3.1.2. 2G

En 1992, las primeras redes europeas de segunda generación iniciaron su actividad. Esta segunda generación de los sistemas de telefonía móvil introdujo la digitalización del enlace radio y el cifrado de llamadas de voz.

La tecnología más común globalmente fue GSM (Sistema Global para las comunicaciones Móviles), desarrollado por el ETSI (Instituto Europeo de Normas de Telecomunicaciones) y basado en TDMA (Acceso Múltiple por División de Tiempo).

Mientras que en Norteamérica, las tecnologías presentes fueron Digital AMPS, basada en TDMA, y cdmaOne, basado en CDMA (Acceso Múltiple por División de Código).

Las principales mejoras que presentaban las redes de segunda generación frente a sus predecesoras fueron:

- La encriptación digital de las llamadas telefónicas entre el teléfono móvil y la estación base.
- El uso más eficiente del espectro aumentando la capacidad de usuarios por banda de frecuencia.
- Servicios de datos para los teléfonos móviles que permitieron el envío y recepción de mensajes de texto cortos o SMS y más tarde de MMS (Servicio de Mensajería Multimedia).

La máxima velocidad de transferencia de datos era de 40 Kbps con el sistema 2.5G GPRS (Servicio General de Paquetes vía Radio), mientras que con la tecnología 2.75G EDGE (Tasas de Datos Mejoradas para la Evolución del GSM) se alcanzaban 384 Kbps (5).

3.1.3. 3G

En 1998 se creó el 3GPP (Proyecto Asociación de Tercera Generación), una colaboración de organizaciones de estandarización que desarrollan protocolos para telecomunicaciones móviles. El proyecto se estableció con el objetivo de desarrollar una especificación para un sistema de telefonía móvil 3G basado en el sistema GSM, UMTS (Sistema Universal de Telecomunicaciones Móviles) (6). Más tarde el objetivo se amplió incluyendo el desarrollo y mantenimiento de:

- El GSM, incluyendo las tecnologías evolucionadas (GPRS, EDGE).
- Un sistema de tercera generación evolucionado y más allá del sistema móvil basado en las redes de núcleo evolucionadas del 3GPP y las tecnologías

de acceso radio apoyadas por los miembros del proyecto (como por ejemplo la tecnología UTRAN (Red de Acceso Radio Terrestre UMTS) y sus modos FDD (Duplexación por División en Frecuencia) y TDD (Duplexación por División en Tiempo)).

- Un IMS desarrollado en un acceso de manera independiente.

También en 1998, se creó el 3GPP2 (Proyecto Asociación de Tercera Generación 2), una colaboración entre las organizaciones de estandarización de telecomunicaciones de Japón, China, Corea del Sur y Estados Unidos (7).

Ambos grupos colaboraron en el diseño de sistemas de tercera generación creando la especificación IMT-2000 (Telecomunicaciones Móviles Internacionales 2000) definida por la UIT. Los estándares que definieron fueron:

- IMT-2000 CDMA Direct Spread. También conocido como WCDMA (Acceso Múltiple por División de Código de Banda Ancha), utilizado en UMTS, sucesor de GSM.
- IMT-2000 CDMA Multi-Carrier. También conocido como CDMA2000, sucesor del CDMA utilizado en 2G (IS-95).
- IMT-2000 CDMA TDD. También conocido como TD-SCDMA (Acceso Múltiple por División de Código Síncrono de División de Tiempo).
- IMT-2000 TDMA Single Carrier. También conocido como EDGE, una tecnología 2.75G (a medio camino entre 2G y 3G).
- IMT-2000 FDMA/TDMA. También conocido como DECT (Telecomunicaciones Inalámbricas Mejoradas Digitalmente).

Para cumplir con los estándares IMT-2000, un sistema debe proporcionar velocidades de datos máximas de 384 Kbit/s para estaciones móviles y 2 Mbit/s para estaciones fijas (8).

El grupo de trabajo 3GPP estandarizó otra serie de tecnologías enmarcadas en la tercera generación de sistemas de telefonía móvil como fueron HSDPA (Acceso Descendente de Paquetes de Datos a Alta Velocidad) en 2002, HSUPA (Acceso Ascendente de Paquetes de Datos a Alta Velocidad) en 2004 y HSPA+ (Acceso de Paquetes de Datos a Alta Velocidad Evolucionado) en 2007, siendo este la fusión y evolución de las dos anteriores. Este nuevo estándar permitía llegar a velocidades de datos tan altas como 337Kbit/s en el enlace descendente y 34Kbit/s en el enlace ascendente (9).

3.1.4. 4G

Al igual que en la generación anterior, la UIT creó un comité para definir las especificaciones de la cuarta generación de telefonía móvil, el IMT Advanced. Estas especificaciones publicadas en 2008 incluían los siguientes requisitos (10):

- Red basada completamente en IP (conmutación de paquetes).
- Interoperabilidad con redes existentes.
- Velocidad de datos de 100 Mbps mientras que el cliente se mueva físicamente a velocidades altas en relación con la estación base, y de 1 Gbps mientras que ambos se encuentren en posiciones relativamente fijas.
- Compartir y utilizar recursos de red dinámicamente para soportar un mayor número de usuarios simultáneos por celda.
- Ancho de banda escalable entre 5 y 20 MHz o hasta 40 MHz.
- Capacidad para ofrecer contenido multimedia de alta calidad a través de la red.

El estándar LTE (Evolución a Largo Plazo), publicado por el 3GPP a finales de 2008 se propuso como candidato para ser una tecnología 4G conforme a los requisitos de la UIT. Sin embargo, no fue hasta 2011 que el 3GPP publicó el estándar LTE Advanced, cumpliendo ahora sí con los requisitos del IMT Advanced.

El último estándar enmarcado en la cuarta generación de telefonía móvil fue LTE Advanced Pro, publicado en 2016 (11).

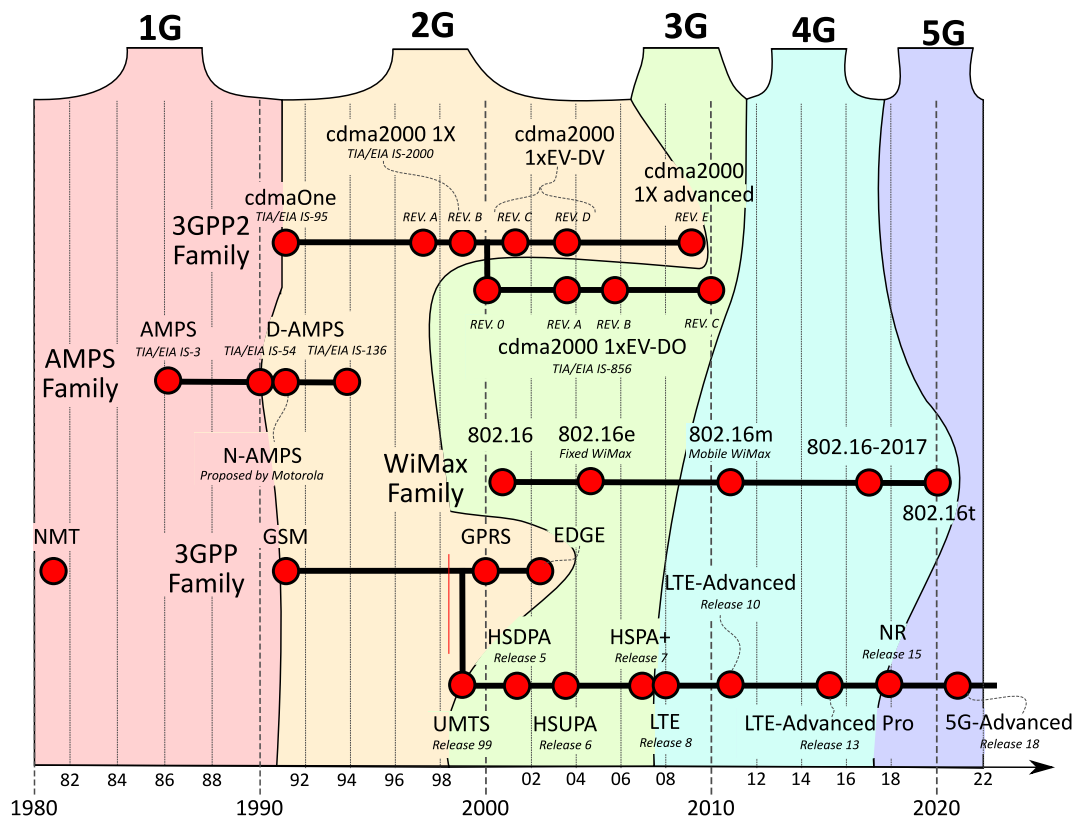


Figura 1: Línea temporal de los distintos estándares de telefonía móvil.

3.2. 5G

En septiembre de 2015, la UIT publicó la recomendación UIT-R M.2083-0 en la que estableció el marco y objetivos generales del futuro desarrollo de las IMT para 2020 y adelante.

Esta recomendación estableció las capacidades de la futura IMT-2020 (12):

- Máxima velocidad de datos: 20 Gbit/s.
- Velocidad de datos experimentada por el usuario: 100 Mbit/s.
- Latencia: 1 ms.
- Movilidad: 500 km/h.
- Densidad de la conexión: 10^6 dispositivos/km².
- Eficiencia energética: 100 veces mayor a la establecida en la IMT Advanced.
- Eficiencia espectral: 3 veces mayor a la establecida en la IMT Advanced.
- Capacidad de tráfico de superficie: 10 Mbit/s/m².

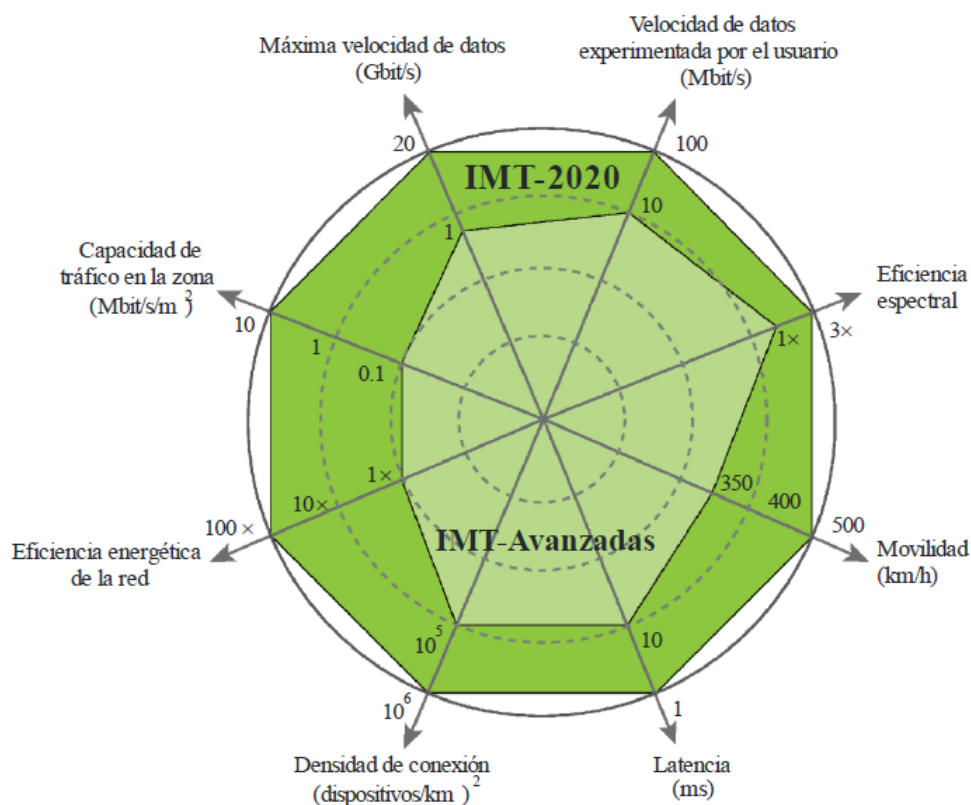


Figura 2: Mejora de las capacidades fundamentales de las IMT-Avanzadas a las IMT-2020. Figura M.2083-03 de la Recomendación de la UIT-R M.2083-0.

La UIT-R define en esta recomendación tres áreas de aplicación principales para las capacidades mejoradas de 5G: eMBB (Banda Ancha Móvil mejorada), URLLC (Comunicaciones de Gran Fiabilidad y Baja Latencia) y mMTC (Comunicaciones masiva de Tipo Máquina):

- **eMBB.** La velocidad de datos experimentada por el usuario, la capacidad de tráfico de superficie, la máxima velocidad de datos, la movilidad, la eficiencia energética y la eficiencia del espectro son todos parámetros muy importantes, aunque la movilidad y la velocidad de datos experimentada por el usuario no tendrá la misma importancia simultáneamente en todos los casos. Por ejemplo, en las zonas de acceso inalámbrico, se requerirá una elevada velocidad de datos del usuario pero reducida movilidad en comparación con el caso de cobertura de área extensa.
- **URLLC.** En algunos casos, el parámetro de baja latencia es de la mayor importancia, por ejemplo para permitir aplicaciones de seguridad esenciales. Dicha capacidad también será imprescindible en casos de elevada movilidad, por ejemplo, en la seguridad del transporte, mientras que una elevada velocidad de datos, por ejemplo, podría ser menos importante.

- **mMTC.** En esta situación, se requiere una elevada densidad de conexión para poder dar servicio al ingente número de dispositivos en la red que, por ejemplo, transmiten sólo de vez en cuando, a pequeña velocidad de bits y con movilidad escasa o nula. Para este caso de utilización es fundamental disponer de dispositivo de bajo coste con una larga vida útil operativa.

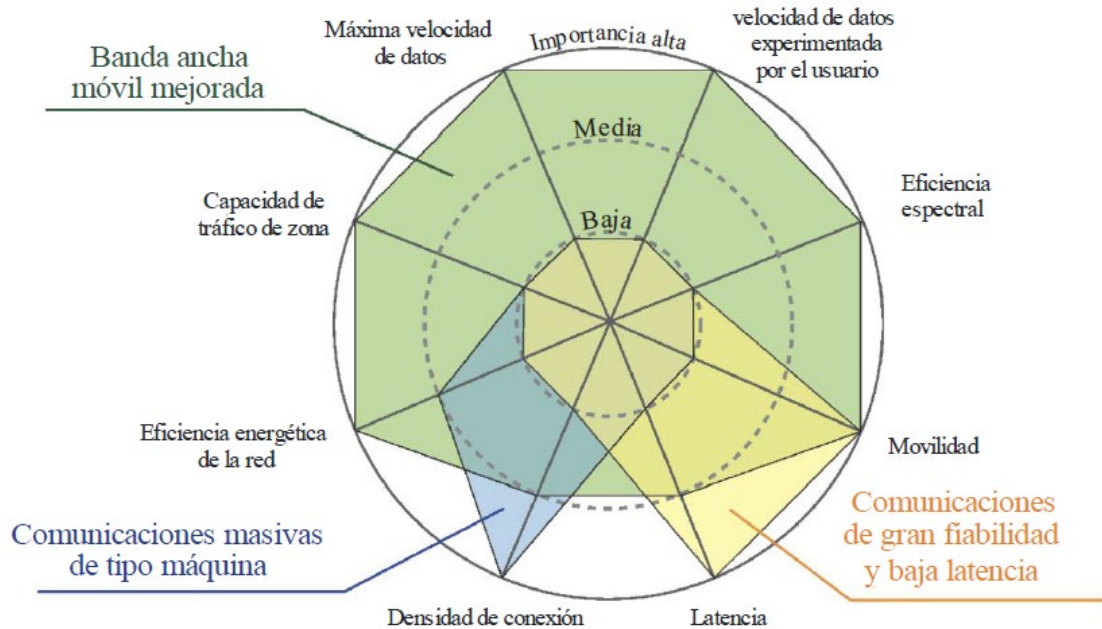


Figura 3: Importancia de las capacidades fundamentales para distintos casos de utilización. Figura M.2083-04 de la Recomendación de la UIT-R M.2083-0.

En febrero de 2017, la UIT estableció los requisitos técnicos mínimos que debía cumplir la nueva interfaz radio de la futura IMT-2020 (13).

Más tarde, en el segundo cuatrimestre de 2018, el 3GPP publica su *Release 15*, en la que se presenta el 5G NR (New Radio), la interfaz radio propia de 5G, así como un nuevo IMS (14).

El 1 de febrero de 2021 se publicó la recomendación UIT-R M.2150-0, titulada “Especificaciones detalladas de las interfaces radioeléctricas terrenales de las Telecomunicaciones Móviles Internacionales 2000 (IMT 2020)”.

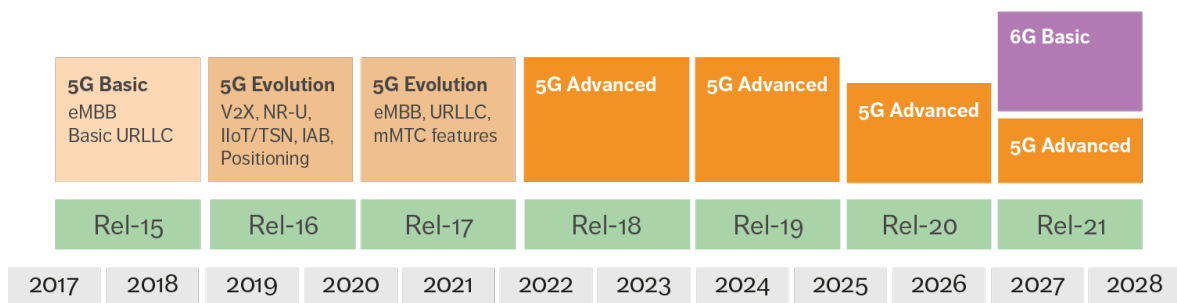


Figura 4: Línea temporal que muestra la evolución del 5G (15).

3.2.1. Arquitectura de red

5G utiliza una arquitectura de servicios basada en la nube construida desde cero donde cada NF (Función de Red) puede comunicarse con otra NF, usando HTTP/2 como protocolo de comunicación básico.

La arquitectura de 5G está diseñada para disminuir las dependencias entre la AN (Red de Acceso) y el CN (Núcleo de Red) empleando una red central unificada independiente del acceso. Esto lo logra mediante una interfaz común entre la red de acceso y el núcleo de red compatible con tecnologías de acceso 3GPP y no 3GPP.

Además, el núcleo de red desacopla el UP (Plano de Usuario) del CP (Plano de Control). Esta función, conocida como CUPS (Separación de los Planos de Control y Usuario), se introdujo por primera vez en la versión 14 del 3GPP. Una característica importante de esta función es que, en caso de un pico de tráfico, puede escalar dinámicamente las funciones del CP sin afectar a las operaciones del UP. Lo que permite la implementación de UPF (Función del Plano de Usuario) más cercanas a la RAN y al UE (Equipo de Usuario) para soportar casos de uso que requieran URLLC (16).

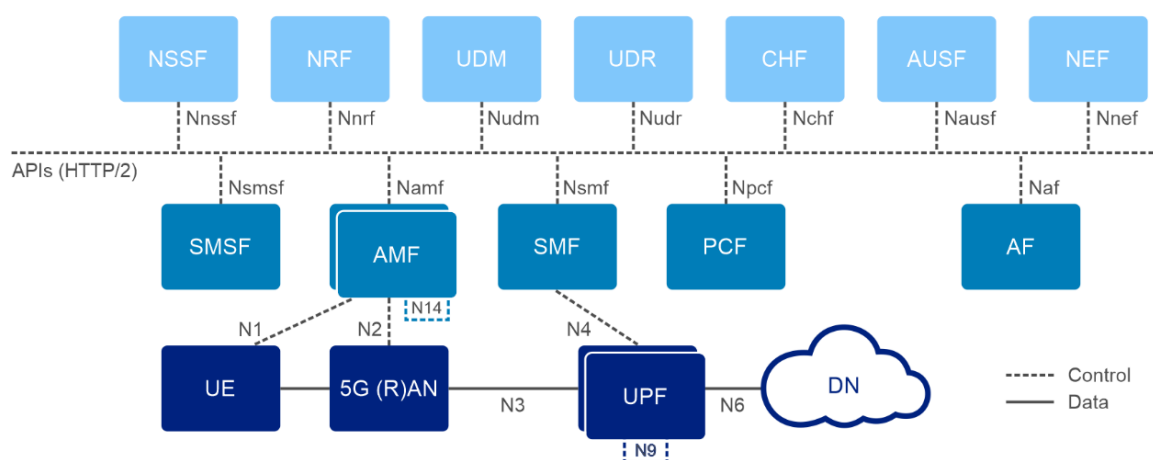


Figura 5: Diagrama simplificado de la arquitectura de red de 5G (16).

3.2.2. Núcleo de red

El núcleo de red se compone de varias NF, donde cada una tiene un propósito único. Estas funciones se comunican interna y externamente a través de interfaces estándar bien definidas, lo que hace que la red 5G sea muy ágil y flexible.

A continuación se describen con más detalle algunas de las funciones clave del núcleo de red de 5G:

- **Función de Plano de Usuario (UPF).** Supervisa la gestión de los datos del usuario durante el proceso de transmisión de los datos. La UPF sirve como punto de conexión entre la RAN y la red de datos. Toma datos de usuario de la RAN y realiza una serie de funciones como la inspección de paquetes, el enrutamiento de tráfico, el procesamiento de paquetes y el cumplimiento de la QoS (Calidad de Servicio) antes de entregarlos a la red de datos o Internet. Esta función permite acercar el plano de datos al borde de la red, lo que genera velocidades de datos más rápidas y latencias más cortas.
- **Función de Gestión de Sesiones (SMF).** Es responsable de establecer, mantener y finalizar sesiones de red para UE. El SMF lleva a cabo estas tareas utilizando protocolos de red como el PFCP (Protocolo de Control de Reenvío de Paquetes).
SMF se comunica con otras funciones de red como PCF, AMF y UPF para garantizar un flujo de datos fluido, una aplicación eficaz de las políticas y un uso eficiente de los recursos de red. También desempeña un papel importante en el manejo de los parámetros de QoS, información de enrutamiento y características de cobro para sesiones de red individuales.
- **Función de Gestión de Acceso y Movilidad (AMF).** Supervisa la gestión de las conexiones y la movilidad. Recibe información de autenticación, control de políticas y relacionada con la sesión de los dispositivos finales y pasa la información de la sesión al PCF, SMF y otras funciones de red.
Además, AMF recupera información de suscripción y admite el servicio de SMS. Identifica un segmento de red utilizando S-NSSAI (Información de Asistencias para la Selección de Segmentos de Red Única), que incluye el SST (Tipo de Segmento/Servicio) y el SD (Diferenciador de Segmento). Las operaciones de la AMF permiten la gestión del registro, la accesibilidad, la conexión y la movilidad de la UE, lo que la convierte en un componente esencial del núcleo de red.

- **Función de Control de Políticas (PCF).** Proporciona el marco para crear políticas que serán consumidas por otras funciones de red del plano de control. Estas políticas pueden incluir aspectos como el control de la calidad de servicio, el monitoreo de uso/gasto de suscriptores, la administración de segmentación de red y la administración de suscriptores, aplicaciones y recursos de la red. Se comunica con otros elementos de la red, como AMF, SMF y UDM, para adquirir información crítica y tomar decisiones políticas acertadas.
 - **Gestión de Datos Unificada (UDM) y Repositorio de Datos Unificado (UDR).** La UDM mantiene datos de suscriptores, políticas y otra información asociada, mientras que la UDR almacena estos datos.

- **Función de Exposición de Red (NEF).** Permite a los operadores de red exponer de forma segura la funcionalidad y las interfaces de red a nivel granular mediante la creación de un puente entre el núcleo de red y la aplicación externa (como puede ser Edge Computing). La NEF también proporciona un medio para que las Funciones de Aplicación (AF) proporcionen información de forma segura a la red 3GPP.

Al abrir las capacidades de la red a aplicaciones de terceros, NEF permite una conexión perfecta entre las capacidades de la red y los requisitos comerciales, optimizando la asignación de recursos de la red y mejorando la experiencia comercial general.

- **Función de Repositorio de Red (NRF).** Sirve como depósito centralizado para todas las instancias de NF. Se encarga de gestionar el ciclo de vida de los perfiles NF, lo que incluye registrar nuevos perfiles, actualizar los antiguos y dar de baja aquellos que ya no estén en uso. La NRF ofrece una API (Interfaz de Programación de Aplicaciones) basada en estándares para el registro y descubrimiento de 5G NF.

Técnicamente, NRF opera almacenando datos sobre todas las instancias de NF, incluidas las funcionalidades, servicios y capacidades admitidos. Cuando se crea una nueva instancia de NF, se registra en la NRF y proporciona todos los detalles necesarios. Posteriormente, cualquier NF que necesite comunicarse con otra NF puede consultar a la NRF los detalles de la instancia de la NF objetivo. Al recibir esta consulta, la NRF responde con la información de instancia de NF más adecuada según el servicio y la capacidad solicitados.

3.2.3. Pilas de protocolos 5G

Una pila de protocolos se define para las comunicaciones entre varias de las funciones de red NF mencionadas anteriormente. A continuación se destacan algunas de las principales (17):

3.2.3.1. Plano de control: la pila de protocolos UE – AMF y UE – SMF

La pila de protocolos entre el UE y la SMF, a través de la AMF, se muestra en la siguiente figura:

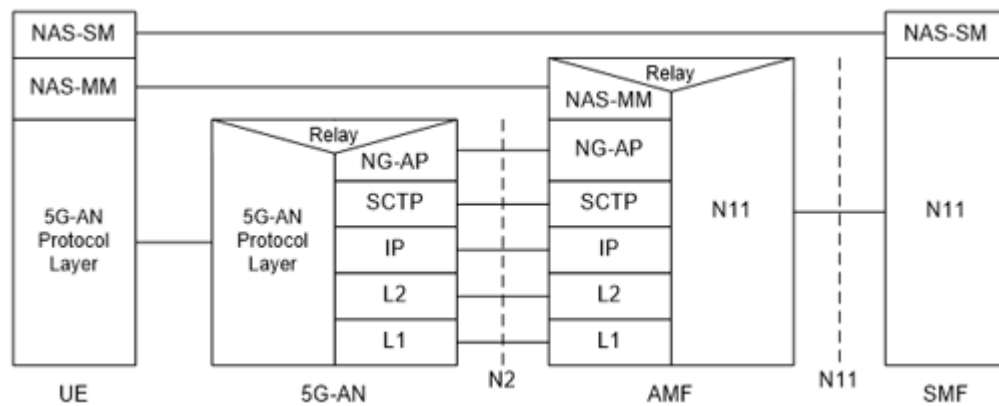


Figura 6: Pila de protocolos del plano de control entre UE, 5G-AN, AMF y SMF. TS 23.501, sección 8.2.

- **NAS-SM:** soporta el manejo de la Gestión de Sesiones entre el UE y la SMF. Admite el establecimiento, modificación y liberación de sesiones de PDU en el plano de usuario. Se transfiere a través de la AMF y es transparente para la AMF.
- **NAS-MM:** soporta funcionalidad de gestión de registros, funcionalidad de gestión de conexiones y activación y desactivación de conexiones en el plano de usuario. También es responsable del cifrado y protección de la integridad de la señalización NAS.
- **5G-AN Protocol Layer:** este conjunto de protocolos/capas depende de la red de acceso. En el caso de NG-RAN, el protocolo de radio entre el UE y el nodo NG-RAN (eNodeB o gNodeB) se especifica en la especificación de E-UTRA & E-UTRAN (para eNodeB) y en la especificación NR (para gNodeB).
- **Protocolo de Aplicación NG (NG-AP):** Protocolo de capa de aplicación entre el nodo 5G-AN y el AMF.
- **Protocolo de Transmisión de Control de Flujo (SCTP):** este protocolo garantiza la entrega de mensajes de señalización entre AMF y el nodo 5G-AN (N2).

3.2.3.2. Plano de usuario: la pila de protocolos UE – AMF y UE – SMF

La siguiente figura ilustra la pila de protocolos para el transporte del plano de usuario relacionado con una sesión PDU (17):

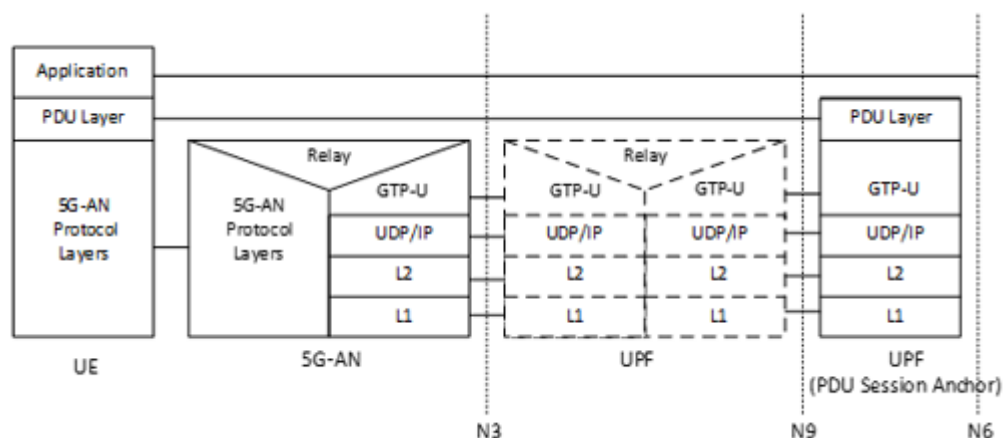


Figura 7: Pila de protocolos del plano de usuario entre el UE, el 5G-AN y la UPF. TS 23.501, sección 8.3.

- **PDU Layer:** esta capa corresponde a la PDU transportada entre el UE y el DN a través de la sesión de PDU. Cuando el tipo de sesión de PDU es UPv4 o IPv6 o IPv4v6, corresponde a paquetes IPv4 o paquetes IPv6 o ambos. Cuando el tipo de sesión de PDU es Ethernet, corresponde a tramas Ethernet, etc.
- **Protocolo de Túnel GPRS para el Plano de Usuario (GTP-U):** este protocolo admite el túnel de datos de usuario a través de N3 (es decir, entre el nodo 5G-AN y la UPF) y N9 (es decir, entre diferentes UPF del 5GC) en la red troncal. GTP encapsulará todas las PDU de usuario final. Proporciona encapsulación a nivel de sesión de PDU. Esta capa lleva también el marcado asociado con un flujo QoS.
- **Pila de protocolos 5G-AN:** este conjunto de protocolos/capas depende de la AN. L2 también se denomina “capa de enlace de datos” y L1 es la “capa física”.
- **UDP/IP:** estos son los protocolos de la red troncal.

3.2.4. Red de acceso

La red de acceso, es la parte de la infraestructura de la red que conecta los dispositivos de usuario a la red central. El estándar New Radio de 5G es la interfaz y tecnología de acceso por radio más reciente.

Es compatible con múltiples bandas de frecuencia incluyendo bandas sub-6 gigahercios y bandas de ondas milimétricas (mmWave), como 24 GHz, 28 GHz y

superiores. Las bandas mmWave ofrecen velocidades de descarga de 1 Gbps o más, pero tienen alcances reducidos en comparación con los servicios sub-6 GHz (18).

Para la L1, capa física, NR utiliza OFDM (Multiplexación por División de Frecuencias Ortogonales) con prefijo cíclico en el enlace descendente (DL); mientras que para el enlace ascendente (UL) puede utilizar OFDM o DFT-s-OFDM (OFDM con precodificación de Transformada Discreta de Fourier). DFT-s-OFDM mejora la cobertura de UL pero tiene una PAPR (Relación de Potencia Pico a Promedio) más baja y está limitada a la transmisión de una sola capa.

Alguna de las características clave de la capa 1 de 5G es que se extiende en múltiples rangos de frecuencia, para permitir el despliegue en frecuencias por país o por región. Las portadoras van desde 400 MHz hasta 100 GHz, pero las bandas licenciadas van desde 600 MHz hasta 39 GHz. Estas frecuencias son reagrupadas en bandas de TV analógica (UHF) y de algunos sistemas satelitales, sin interferencias ya que se utilizan en diferentes ubicaciones (17).

Para terrestre se identifican 3 rangos principales de frecuencias:

- **Hasta 1 GHz:** con sus mejores características de propagación, este conjunto está destinado a cubrir grandes áreas, normalmente para despliegue rural. El ancho de banda máximo para un operador es de 100 MHz.
- **De 1 a 6 GHz:** esta gama intermedia es para el despliegue de 5G en un contexto urbano o suburbano. También en este caso el ancho de banda máximo es de 100 MHz.
- **Superior a 6 GHz:** con su peor propagación pero su mayor ancho de banda para el usuario (ancho de banda máximo de 400 MHz), este rango está pensado para entornos urbanos densos (tipo de cobertura *hot-spot*).

Por otro lado, en cuanto a la implementación, se definen dos opciones para 5G: “Non-Stand Alone” (NSA) y “Stand-Alone” (SA).

3.2.4.1. Arquitectura “Non-Stand Alone” (NSA)

En esta implementación, la AN de 5G y su nueva interfaz radio, NR, se utilizan junto con la red central de infraestructura LTE y EPC (Núcleo de Paquete Evolucionado) existente (radio 4G y núcleo 4G respectivamente), haciendo así que la tecnología NR esté disponible sin reemplazo de red. En esta configuración sólo se admiten los servicios 4G, pero disfrutan de las capacidades que ofrece 5G New Radio (menor latencia, etc.).

La arquitectura NSA se ilustra en la siguiente figura:

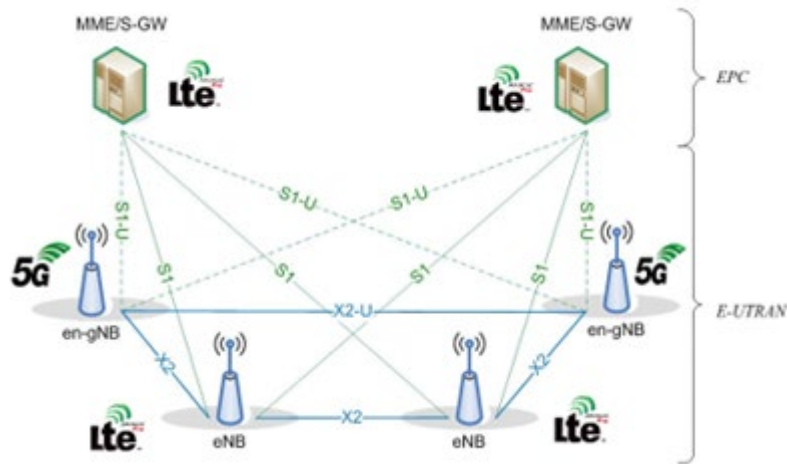


Figura 8: La arquitectura NSA de 5G.

La arquitectura NSA puede verse como un paso temporal hacia una implementación “5G completa”, donde la red de acceso 5G está conectada a la red central 4G. En la arquitectura NSA, la estación base (5G) NR (nodo lógico “en-gNB”) se conecta a la estación base (4G) LTE (nodo lógico “eNB”) a través de la interfaz X2. La interfaz X2 se introdujo antes de la *Release 15* para conectar dos eNB. Sin embargo, a partir de la *Release 15*, también admite la conexión de un eNB y un en-gNB para proporcionar NSA.

La arquitectura NSA ofrece conectividad dual, tanto a través de la red de acceso 4G (E-UTRA) como de la red de acceso de 5G (NR). Por eso también se denomina “EN-DC”, por “E-UTRAN y NR Dual Connectivity” (17).

En EN-DC, el eNB del 4G es el nodo maestro, mientras que el en-GNB del 5G es el nodo secundario.

3.2.4.2. Arquitectura “Stand Alone” (SA)

La arquitectura SA puede verse como la implementación “5G completa”, sin necesidad de ninguna parte de una red 4G para funcionar. Solo en esta implementación se admite el conjunto completo de servicios 5G.

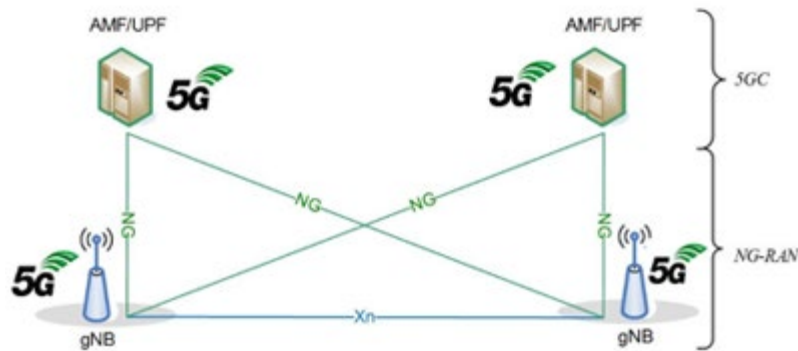


Figura 9: La arquitectura SA de 5G.

Las estaciones base NR (nodo lógico “gNB”) se conectan entre sí a través de la interfaz Xn, y la red de acceso (llamada “NG-RAN para arquitectura SA”) se conecta al núcleo de la red utilizando la interfaz NG.

Más que una nueva interfaz de radio, la red 5G introduce una serie de nuevas tecnologías clave. Algunas de ellas se presentan brevemente aquí:

- **Network Slicing:** es la capacidad de desplegar y utilizar simultáneamente diferentes CN, cada uno especializado en el aprovisionamiento de un determinado conjunto de servicios y/o un determinado conjunto de suscriptores. Por ejemplo, un segmento puede soportar a los suscriptores del operador de red “habitual”, otro segmento puede estar dedicado a soportar a los suscriptores de un OMV (Operador Móvil Virtual), otro puede manejar un servicio específico como el seguimiento de contenedores a través de M2M, etc.
- **Virtualización de funciones de red:** como se muestra en la Figura 5: Diagrama simplificado de la arquitectura de red de 5G (16)., todas las funciones de red se comunican a través de una interfaz común y, por tanto pueden ubicarse en cualquier lugar. Esto permite una flexibilidad mucho mayor en la implementación de la red. El mantenimiento también se simplifica enormemente, ya que se puede establecer fácilmente una NF temporal.
- **EDGE computing:** se introduce cierta potencia computacional lo más cerca posible (físicamente) del usuario final. De hecho, algunas aplicaciones como la realidad virtual, las fábricas del futuro o la conducción autónoma son muy exigentes en cuanto al tiempo de respuesta de la propagación/red. Para reducir este tiempo, se introducen algunas “replicaciones locales” de un servidor principal más cerca del usuario final.

3.2.4.3. Arquitectura General de Protocolos de la interfaz radio

La arquitectura general de protocolos de la interfaz radio es similar a la de LTE en 4G. Se definen canales lógicos, canales de transporte y canales físicos al igual que en LTE. Los canales lógicos se caracterizan por el tipo de información transmitida y los canales de transporte se caracterizan por el modo en el que se transmite información (19).

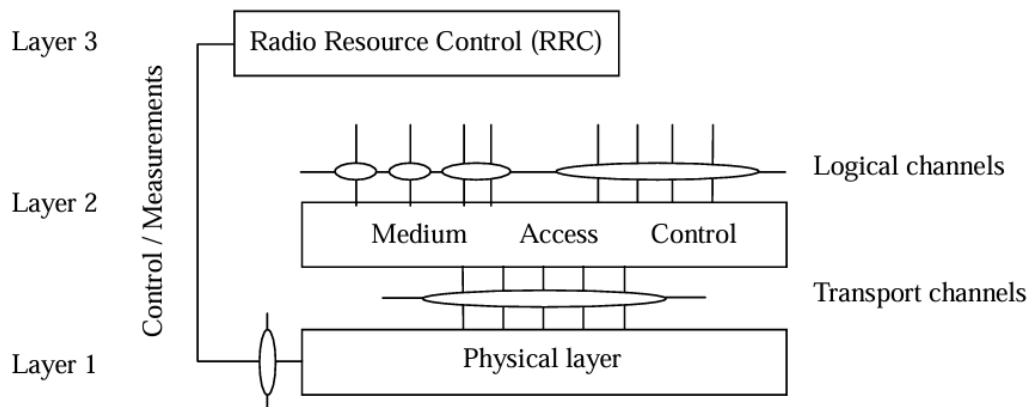


Figura 10: Arquitectura del protocolo de interfaz radio alrededor de la capa física. Figura extraída de la especificación técnica 38.201 del 3GPP.

3.2.4.4. Capa física

La capa física detalla todos los procedimientos y mecanismos relacionados con el acceso múltiple, los canales físicos y la modulación, así como la codificación de canal, los procedimientos, las mediciones y los servicios que se proporcionan a las capas superiores.

New Radio mantiene la OFDM de LTE dotándola de gran flexibilidad para afrontar los retos del 5G. Además, se permite el uso opcional de técnicas de enventanado y filtrado para confinar el espectro cuando múltiples servicios comparten la misma portadora (19).

Se mantienen los canales físicos de LTE:

En el enlace descendente:

- PDSCH. Canal físico compartido descendente.
- PDCCH. Canal físico común de control descendente.
- PBCH. Canal de difusión descendente.

En el enlace ascendente:

- PRACH. Canal físico de acceso aleatorio.
- PUSCH. Canal físico compartido ascendente.
- PUCCH. Canal físico común de control ascendente.

A todos estos canales se les dota de mayor flexibilidad operativa.

También se mantienen las señales físicas de LTE (19):

Señales de referencia (RS) y sincronización primaria (PSS) y secundaria (SSS), pero con una periodicidad configurable en lugar de fija, como en LTE. Si hay conformación de haz, el gNB transmite las señales PSS y SSS separadamente en cada haz. La red puede configurar múltiples PSS y SSS en el dominio de la frecuencia.

Se introduce una nueva señal de seguimiento de fase (PT-SS) para corregir posibles errores de fase en los símbolos PDSCH que no contengan la señal de demodulación (DM-RS).

Para el multiacceso se admiten dos opciones (19):

1. OFDM con prefijo cíclico (CP-OFDM) en ambos enlaces, ascendente y descendente.
2. CP-OFDM en el enlaces descendente y DFT-S-OFDM en el ascendente. Esta segunda opción se utiliza en LTE con el nombre de SC-FDMA, aunque el nuevo nombre es más preciso ya que implica un procesamiento previo de expansión con la transformada discreta de Fourier (DFT-spread).

En cuanto a las modulaciones, se consolida 256QAM para los enlaces ascendentes y descendentes y se introduce $\pi/2$ BPSK para la señalización de control de enlace ascendente, quedando así (19):

- Enlace descendente: QPSK, 16QAM, 64QAM y 256QAM.
- Enlace ascendente: QPSK, 16QAM, 64QAM y 256QAM para CP-OFDM y $\pi/2$ BPSK, QPSK, 16QAM, 64QAM y 256QAM para DFT-S-OFDM con CP.

Se mantiene el módulo básico de asignación de recursos en el dominio de la frecuencia, que, al igual que en LTE, es el PRB (Bloque Físico de Recursos) formado por 12 subportadoras consecutivas.

Las transmisiones son sincrónicas y programadas, aunque se permite al menos una opción de transmisión ascendente sin programación para aplicaciones eMBB y URLLC.

Para la codificación de canal de las señales de datos, se emplean nuevos códigos, específicamente la familia de códigos de paridad y baja densidad (LDPC) con tasas de 1/3 o 1/5. Estos códigos, propuestos en los años 70, permiten la implementación de

decodificadores con baja latencia y ofrecen un alto rendimiento de datos. El uso de estos códigos se complementa con el entrelazado de bits y el ajuste de tasa, *rate matching*, mediante *puncturing* o repetición de bits, como se hace en LTE.

Para la señalización de control entre las capas L1 y L2 en los mensajes DCI (Información de Control en el enlace Descendente) y UCI (Información de Control en el enlace Ascendente), se emplean diferentes tipos de códigos según la longitud de los mensajes. Si los mensajes tienen más de 11 bits, se utiliza la codificación polar. Entre 3 y 11 bits, se usan códigos Reed-Muller, para mensajes de 2 bits se utiliza la codificación *simplex*, y para mensajes de 1 bit, se emplea un código de repetición (19).

3.2.4.5. Acceso múltiple

Como se ha mencionado anteriormente, el acceso múltiple se basa en CP-OFDM con la opción de DFT-S-OFDM y CP para el enlace ascendente. La transmisión puede utilizar espectro pareado con FDD o no pareado con TDD. Además, se define si especificar explícitamente las anchuras de banda. Aunque, en cada caso, el número de bloques de recursos a utilizar se ajustará al ancho de banda disponible (19).

En el dominio del tiempo, se define la unidad de tiempo de base como:

$$T_C = \frac{1}{\Delta f_{max} \cdot N_F} \text{ (segundos)} \quad (1)$$

Con $\Delta f_{max} = 480 \text{ kHz}$ y $N_{ref} = 4096$. Se mantiene también el tiempo de referencia T_S que se estableció para LTE, con la nueva definición:

$$T_S = \frac{1}{\Delta f_{ref} \cdot N_{ref}} \text{ (segundos)} \quad (2)$$

Siendo $\Delta f_{ref} = 15 \text{ kHz}$ y $N_{ref} = 2048$, valores adoptados en LTE. La relación entre estas magnitudes es $K = T_S/T_C = 64$

Al igual que en LTE la trama de radio se define con una duración de 10 ms. Esta trama se divide en 10 subtramas de 1 ms cada una, como en LTE. La subtrama puede contener uno o varios intervalos, *slots*, consecutivos, mientras que en LTE solo contiene dos. Cada intervalo aloja 14 símbolos adyacentes, en contraste con LTE, donde pueden ser 3, 6 o 7 dependiendo del CP (19).

Para la multiplexación TDD, la trama se divide en dos semitramas numeradas del 0 al 4 y del 5 al 9, respectivamente.

En FDD, una subportadora soporta un conjunto de tramas para el enlace ascendente y otra portadora maneja el conjunto de tramas para el enlace descendente. La transmisión ascendente comienza un tiempo T_{TA} (TA, *timing advanced*) antes del inicio de la trama descendente correspondiente, siendo:

$$T_{TA} = (N_{TA} + N_{TA\,DHset}) \cdot T_C \text{ (segundos)} \quad (3)$$

$N_{TA\,DHset}$ depende de la banda de frecuencias donde estén las portadoras ascendente y descendente.

Como es sabido, en OFDM, el periodo útil de un símbolo debe ser igual a la inversa de la separación entre las subportadoras para mantener la ortogonalidad. Dado que el número de símbolos por intervalo es fijo e igual a 14, pero el número de intervalos por subtrama es variable, también variará el periodo de símbolo y, por consiguiente, la separación entre las subportadoras. Esta separación se define en función de un parámetro μ en potencias de 2. Por lo tanto, el número de símbolos consecutivos por subtrama será (19):

$$N_{símbolos}^{subtrama,\mu} = N_{símbolos}^{slot} \times N_{slot}^{subtrama,\mu} \quad (4)$$

Los intervalos se numeran, en orden creciente, dentro de la subtrama. Para la configuración con el parámetro μ :

$$n_s^\mu \in (0, 1 \dots N_{slot}^{subtrama,\mu} - 1) \quad (5)$$

Con Prefijo cíclico normal			
μ	$N_{símbolos}^{slot}$	$N_{slot}^{trama,\mu}$	$N_{slot}^{subtrama,\mu}$
0	14	10	1
1	14	20	2
2	14	40	4
3	14	80	8
4	14	160	16
5	14	320	32
6	14	640	64
Con Prefijo cíclico extendido			
2	12	40	2

Tabla 1: Número de símbolos OFDM por intervalo, intervalos por trama e intervalos por subtrama. Tablas 4.3.2-1 y 4.3.2-2 de la especificación técnica 38.211 del 3GPP.

En el dominio de la frecuencia, se establecen diferentes numerologías. Se entiende por numerología la combinación de valores de separación entre subportadoras y CP. Estos valores proceden de las capas superiores del protocolo (20).

μ	$\Delta f = 15 \cdot 2^\mu \text{ kHz}$	Tipo de CP
0	15	N
1	30	N
2	60	N – E
3	120	N
4	240	N
5	480	N
6	960	N

CP: N, normal; E, extendido

Tabla 2: Numerologías de transmisión soportadas. Tabla 4.2-1 de la especificación técnica 38.211 del 3GPP.

3.2.4.6. Asignación de recursos y velocidad de bits

La asignación de recursos, al igual que en LTE, se realiza mediante bloques de recursos que constan de 12 subportadoras consecutivas en frecuencia y un periodo de símbolo en el tiempo.

En una subportadora hay $N_{mod} = \log_2 M$ bits siendo M el número de estados de modulación (2, 4, 16, 64, 256). En un RB se tendrán $12 \cdot N_{mod}$ bits. Como la duración de un símbolo OFDM es:

$$t_{simb} = \frac{1}{14 \cdot 2^\mu} \text{ milisegundos} \quad (6)$$

La tasa o velocidad de bits resultante será $12 \cdot N_{mod}/t_{simb}$ es decir

$$R = N_{mod} \cdot N_{RB} \cdot 2^\mu \cdot 168 \text{ kbit/s} \quad (7)$$

En el caso de transmisión con MIMO, el valor de R habrá que multiplicarlo por el número de flujos de MIMO (2, 4, 8, ...) (19).

3.2.4.7. Bandas de operación

Se han definido dos rangos de frecuencias, uno es el rango “sub-6-GHz”, designado como FR1 (*Frequency Range 1*), y el otro es la banda de ondas milimétricas, conocido como FR2 (*Frequency Range 2*).

La anchura de banda y la separación entre subportadoras varían según el rango. Para FR1, la anchura máxima de banda es de 100 MHz, mientras que para FR2 es de 400 MHz. En cuanto a la separación entre subportadoras, los valores de 15 y 30 kHz solo se pueden utilizar en FR1, el valor de 120 kHz solo en FR2, y el valor de 60 kHz se puede utilizar tanto en FR1 como en FR2 (19) (20).

Designación del rango		Rango de frecuencias correspondientes
FR1		410 MHz – 7125 MHz
FR2	FR2-1	24250 MHz – 52600 MHz
	FR2-2	52600 MHz – 71000 MHz

Tabla 3: Definición de los rangos de frecuencia. Tabla 5.1-1 de la especificación técnica 32.104 del 3GPP

Los valores de las bandas de frecuencias de enlaces ascendentes y descendentes, anchuras de banda y modo de duplexión se definen en la especificación técnica 38.104 del 3GPP.

También se establece el siguiente número máximo de bloques de recursos para cada anchura de banda, gama de frecuencias y separación de subportadoras (20):

SCS (kHz)	3 MHz	5 MHz	10 MHz	15 MHz	20 MHz	25 MHz	30 MHz	35 MHz	40 MHz	45 MHz	50 MHz	60 MHz	70 MHz	80 MHz	90 MHz	100 MHz
	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}
15	15	25	52	79	106	133	160	188	216	242	270	-	-	-	-	-
30	-	11	24	38	51	65	78	92	106	119	133	162	189	217	245	273
60	-	-	11	18	24	31	38	44	51	58	65	79	93	107	121	135

Tabla 4: Número de bloques de recursos para FR1. Tabla 5.3.2-1 de la especificación técnica 38.104 del 3GPP.

SCS (kHz)	50 MHz	100 MHz	200 MHz	400 MHz
	N_{RB}	N_{RB}	N_{RB}	N_{RB}
60	66	132	264	-
120	32	66	132	264

Tabla 5: Número de bloques de recursos para FR2-1. Tabla 5.3.2-2 de la especificación técnica 38.104 del 3GPP.

SCS (kHz)	100 MHz	400 MHz	800 MHz	1600 MHz	2000 MHz
	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}
120	66	264	-	-	-
480	-	66	124	248	-
960	-	33	62	124	148

Tabla 6: Número de bloques de recursos para FR2-2. Tabla 5.3.2-3 de la especificación técnica 38.104 del 3GPP.

3.3. Implementaciones de 5G

A continuación, se presentan algunas de las diferentes herramientas de código abierto que permiten desplegar completamente un sistema 5G.

- **OpenAirInterface5G:**



Figura 11: Logotipo de OpenAirInterface.

Se trata de una plataforma creada por Eurecom capaz de desplegar tanto la CN como la RAN de 4G y 5G. El proyecto está dividido en dos ramas que tratan por separado la parte de acceso radio (OpenAirInterface-RAN) y el núcleo de red (OpenAirInterface-Core Network).

Actualmente, la interfaz radio soporta (21):

- NSA gNB.
- SA gNB.
- 5G NSA & SA UE.

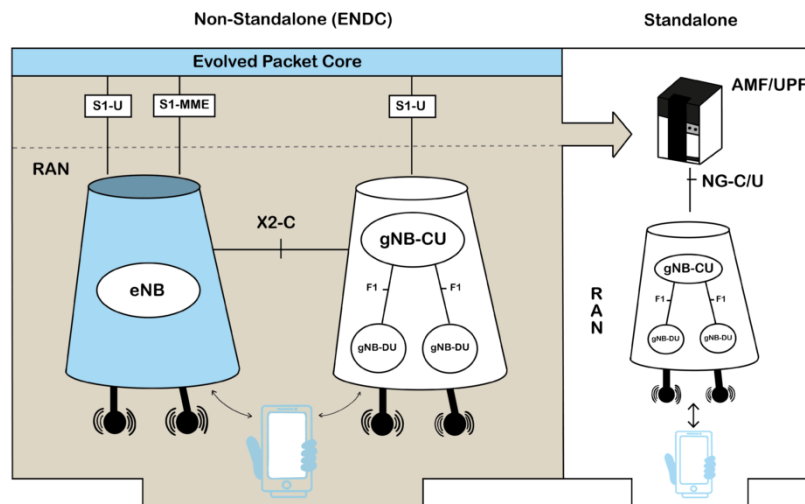


Figura 12: Esquema de la Red de Acceso Radio (RAN) implementada por la plataforma OpenAirInterface.

Por otro lado, el núcleo de red cuenta actualmente con las siguientes características (22):

- Registro de UE, baja de registro y solicitud de servicio.
- Gestión de sesiones (establecimiento, modificación y liberación de sesiones de PDU).

- Transferencia N2.
- HTTP/2.
- Compatibilidad con FQDN (Nombre de Dominio Completo).
- Paginación.
- *Network Slicing* (parcialmente compatible).

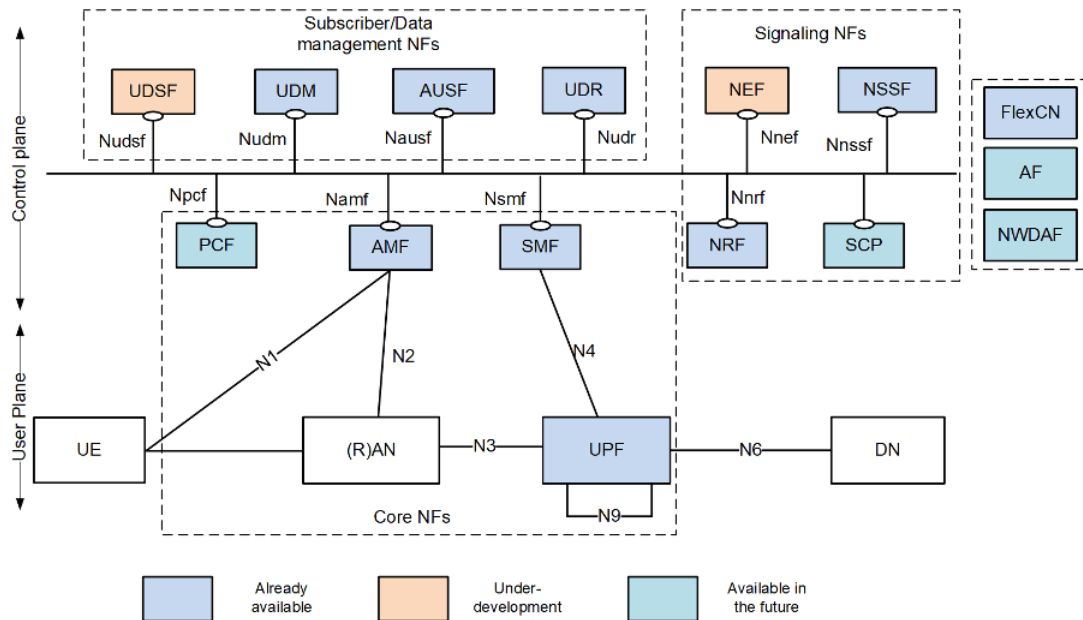


Figura 13: Implementación actual del Núcleo de Red o Red Central (CN) de 5G en OpenAirInterface.

- **Open5GS:**



Figura 14: Logotipo de Open5GS.

Se trata de una herramienta basada en la *Release 17* que es capaz de desplegar el CN de 4G/5G NSA y 5G SA (23).

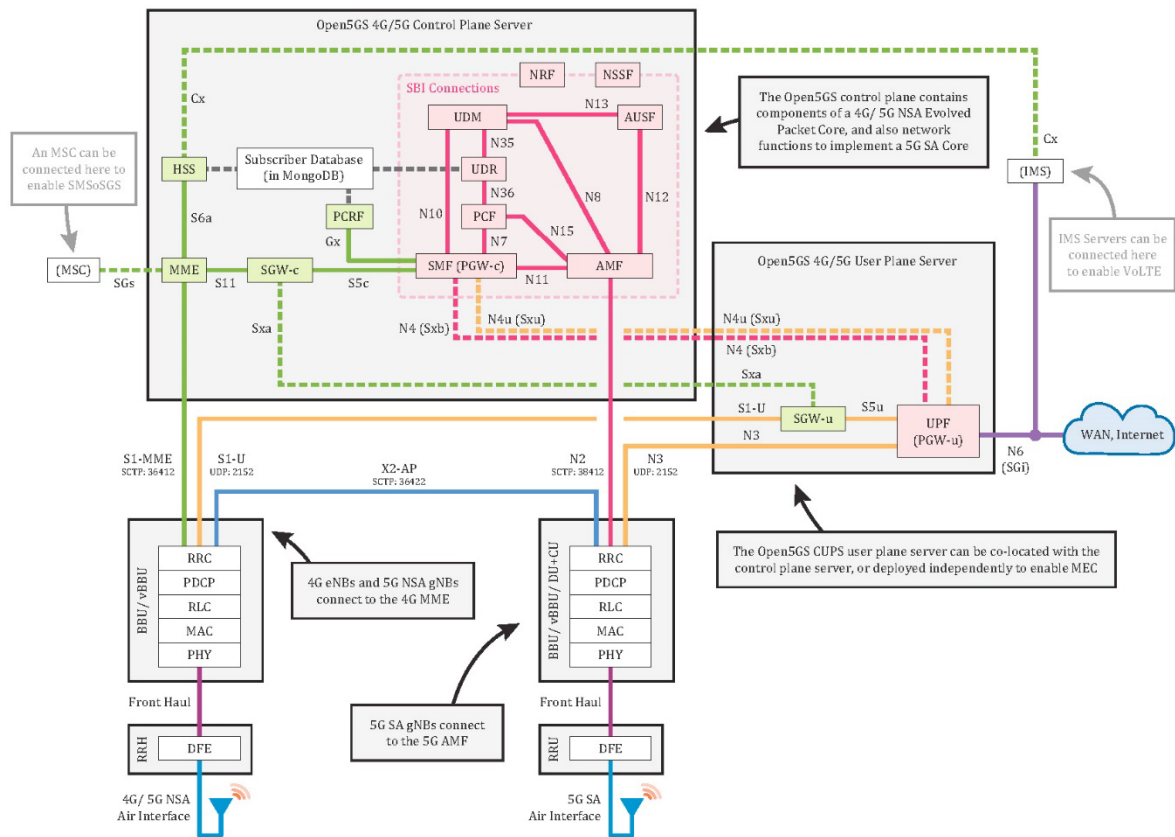


Figura 15: Esquema general de la Arquitectura 5G en el que se diferencia por colores las distintas implementaciones de Open5gs. En verde se muestran los componentes y las conexiones propias del núcleo 4G/5G NSA, mientras que en rosa se muestran los propios de 5G SA.

Componentes del núcleo 4G/5G NSA:

- MME – Entidad de Gestión de la Movilidad.
- HSS – Base de datos de abonado.
- PCRF – Función de políticas y reglas de cobro.
- SGWC – Plano de control de puerta de enlace de servicio.
- SGWU – Plano de usuario de puerta de enlace de servicio.
- PGWC/SMF – Plano de control de puerta de enlace de paquetes. (Contenido dentro de Open5GS SMF).
- PGWU/UPF – Plano de usuario de puerta de enlace de paquetes. (Contenido dentro de Open5GS UPF).

Componentes del núcleo 5G SA:

- NRF – Función de Repositorio de NF.
- SCP – Proxy de Comunicación de Servicio.
- SEPP – Proxy de Protección de Borde de Seguridad.
- AMF – Función de Gestión de Acceso y Movilidad.
- SMF – Función de Gestión de Sesiones.

- UPF – Función de Plano de Usuario.
 - AUSF – Función del Servidor de Autenticación.
 - UDM – Gestión de Datos Unificada.
 - UDR – Repositorio de Datos Unificado.
 - PCF – Función de Política y Cobro.
 - NSSF – Función de Selección de Segmentos de Red.
 - BSF – Función de Soporte Vinculante.
- **free5GC:**



Figura 16: Logotipo de free5GC.

Se trata de una herramienta basada en la *Release 15* y posteriores capaz de desplegar la Red Central de 5G (24).

Cuenta actualmente con las siguientes características (25):

- Registro inicial, periódico, de movilidad, sin autenticación (debido a que la RAN hizo la reelección de AMF).
- Redirección de NAS cuando el AMF no puede llevar a cabo el registro del UE con un NSSAI incorrecto.
- Autenticación 5G-AKA y EAP-AKA.
- Seguridad NAS:
 - Cifrado: NEA0, NEA1, NEA2, NEA3.
 - Integridad: NIA0, NIA1, NIA2, NIA3.
- Baja de UE.
- Petición de servicio activada por UE o por la red.
- *AN Release*.
- Establecimiento de sesión de PDU.
- Características de cobro.
- Liberación de sesión de PDU.
- *Handover* (N2, Xn).
- QoS en el plano de control.
- Métricas de uso de la interfaz N4.

- **UERANSIM:**



Figura 17: Logotipo de UERANSIM.

Se trata de una herramienta capaz de simular la RAN y los UE. Implementa parcialmente la interfaz radio 5G-NR, simulando la comunicación entre gNodeB y UE sobre el protocolo UDP (26).

Cuenta con las siguientes características (27):

- Funciones del NAS:
 - Autenticación primaria y acuerdo de claves.
 - Control del modo de seguridad.
 - Identificación.
 - Actualización de configuración de UE genérica.
 - Registro Inicial y Periódico.
 - Baja iniciada por UE o por la red.
 - Establecimiento de sesión de PDU iniciado por UE.
 - Liberación de sesión de PDU iniciada por UE o por la red.
 - Petición de servicio.
 - Paginación.
 - Implementa todos los algoritmos de integridad y cifrado. (IA1, IA2, IA3, EA1, EA2, EA3).
 - Implementa todos los procedimientos de autenticación primaria y acuerdo de claves, (5G AKA, EAP AKA).
- Funciones NGAP:
 - Configuración de recursos de sesión de PDU.
 - Liberación de recursos de sesión de PDU.
 - Configuración del contexto inicial.
 - Liberación de contexto UE (nodo NG-RAN iniciado y AMF iniciado).
 - Modificación del contexto UE.
 - Mensaje inicial de UE.
 - Paginación.
 - Transporte NAS de enlace descendente.

- Transporte NAS de enlace ascendente.
- Indicación de no entrega del NAS.
- Redirigir solicitud de NAS.
- Configuración NG.
- Plano de usuario:
 - Implementa el protocolo GTP (Protocolo de Túnel de GPRS).
 - Solo admite IPv4.
- **srsRAN project:**



Figura 18: Logotipo de srsRAN Project.

Se trata de una herramienta basada en la *Release 17*, la cual ofrece soluciones software para la implementación de la RAN de 4G y 5G SA/NSA en equipos SDR. Además cuenta con un simulador de equipo de usuario (srsUE) incluido en la implementación 4G del proyecto (srsRAN 4G) (28).

Cuenta con las siguientes características (29):

- Soporte de FDD/TDD, todas las bandas FR1.
- Soporta todos los anchos de banda.
- Espaciado de subportadoras de 15/30 kHz.
- Soporta todos los canales físicos, incluido el PUCCH Format 1 y 2,, excluyendo Sounding-RS.
- Codificador/decodificador LDPC y Polar altamente optimizado para ARM Neon y x86 AVX2/AVX512
- Todos los procedimientos del RRC.
- Todos los procedimientos MAC.
- Compatibilidad con Split 7.2 utilizando la biblioteca OFH interna.
- Soporte para QAM-256.
- 4x4 MIMO.
- Slicing.
- Soporta aceleración por hardware.
- Soporta NTN GEO.

3.4. Radio Definida por Software (SDR)

La Radio Definida por Software es un sistema de radiocomunicaciones donde varios de los componentes típicamente implementados en hardware (mezcladores, filtros, moduladores/demoduladores, detectores, etc.) son implementados en software (30).

A continuación, se presentan algunos de los dispositivos disponibles actualmente en el mercado.

- **USRP B210.**



Figura 19: SDR USRP B210.

Se trata de un dispositivo de SDR *full-duplex* que cuenta con dos canales RX y dos canales TX con un ancho de banda que va desde los 70 MHz hasta los 6 GHz. Es compatible con el *driver* UHD, lo que la hace compatible con multitud de aplicaciones software como son GNU Radio u OpenBTS. Soporta conectividad USB 3.0 y MIMO 2x2. Tiene un coste aproximado de 2.225 € (31).

- **USRP X310.**



Figura 20: SDR USRP X310.

Se trata de un dispositivo de SDR *full-duplex* que cuenta con dos canales RX y dos canales TX con un ancho de banda que va desde la frecuencia continua hasta los 6 GHz. También es compatible con el *driver* UHD. El equipo incluye interfaces PCIe, 10Gigabit Ethernet y Gigabit Ethernet. Soporta MIMO 4x4. Tiene un coste aproximado de 9.898 € (32).

- **USRP N310.**



Figura 21: SDR USRP N310

Se trata de un dispositivo SDR *full-duplex* que cuenta con cuatro canales RX y cuatro canales TX con un ancho de banda que va desde los 10 MHz hasta los 6 GHz. También es compatible con el *driver* UHD. El equipo incluye interfaces 10Gigabit Ethernet, Gigabit Ethernet y SFP+. Soporta MIMO. Tiene un coste aproximado de 18.334 € (33).

- **LimeSDR.**



Figura 22: SDR LimeSDR.

Se trata de un dispositivo SDR *full-duplex* que cuenta con seis canales RX y cuatro canales TX con un ancho de banda que va desde los 100 kHz hasta los 3.8 GHz. Utiliza un *driver* específico. El equipo soporta conectividad USB 3.0 y MIMO 2x2 (34). Tiene un coste aproximado de 449 € (35).

- **RTL-SDR.**



Figura 23: SDR RTL-SDR.

Se trata de un dispositivo SDR basado en el *chipset* R820T2 que cuenta con un canal RX con un ancho de banda que va desde los 500 kHz hasta 1.766 GHz. Utiliza un *driver* propio. El equipo soporta conectividad USB. Tiene un coste aproximado de 40 € (36).

- **HackRF One.**



Figura 24: SDR HackRF One.

Se trata de un dispositivo SDR *half-duplex* que cuenta con un canal TX/RX con un ancho de banda que va desde 1 MHz hasta los 6 GHz. Utiliza un *driver* propio. El equipo soporta conectividad USB 2.0 (37). Tiene un coste aproximado de 310 € (38).

- **bladeRF.**

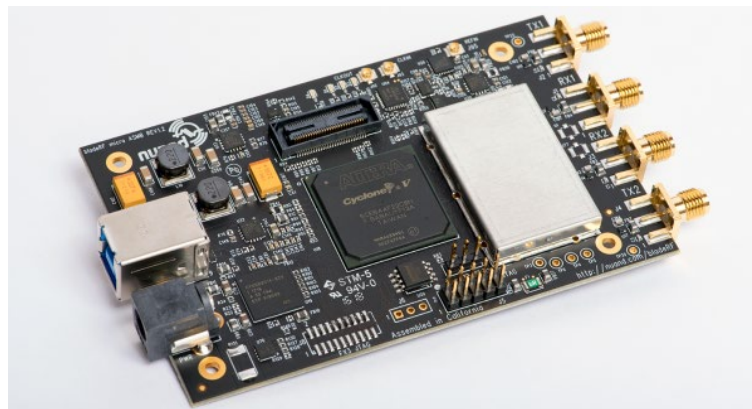


Figura 25: SDR bladeRF.

Se trata de un dispositivo SDR *full-duplex* que cuenta con dos canales con un ancho de banda que va desde los 300 MHz hasta 3.8 GHz. Utiliza un *driver* propio. El equipo soporta conectividad USB 3.0 y 2x2 MIMO. Tiene un coste aproximado de \$ 670 en su variante xA5 (39).

4. SOLUCIÓN ADOPTADA

En primer lugar, se realiza un análisis de las plataformas anteriormente descritas (Free5GC, Openairinterface5G, Open5gs, ...). Tras este análisis, se propone implementar la plataforma Open5GS como núcleo de red y utilizar Kamailio IMS como subsistema multimedia para ofrecer servicio de voz móvil y SMS. Además, se localiza un repositorio en GitHub mantenido por el usuario herlesupreeth llamado “docker_open5gs” que ofrece una solución Docker para la implementación de un sistema 4G o 5G SA/NSA que se toma como base para la implementación de la red 5G.

La principal restricción que existe a la hora de comenzar con este trabajo, es la limitada variedad de plataformas SDR de las que dispone la Universidad de Jaén: HackRF One y bladeRF.

Considerando las limitaciones de la plataforma HackRF One que, al tratarse de una plataforma *half-duplex*, no es capaz de simular una estación base, se opta por utilizar la plataforma SDR bladeRF. No obstante, tras examinar las características de la plataforma srsRAN, se puede comprobar que no soporta el despliegue 5G con bladeRF, sino únicamente el de un sistema 4G.

Debido a esta limitación de la plataforma srsRAN se opta por estudiar la viabilidad de Openairinterface5G, que ofrece una solución 5G SA compatible, a priori, con la plataforma SDR bladeRF. Sin embargo, tras varias pruebas se comprueba que esta plataforma tampoco soporta el despliegue de un sistema 5G con una placa bladeRF.

En definitiva, se observa que el despliegue 5G de las plataformas srsRAN y Openairinterface5G no es compatible con las plataformas SDR disponibles en la Universidad de Jaén, por lo que se exploran distintas alternativas para el desarrollo del trabajo.

La primera alternativa consiste en realizar el despliegue de una red 4G, ya que no se cuentan con los medios necesarios para un despliegue 5G.

Otra alternativa explorada consiste en la compra, por parte de la Universidad, de la plataforma SDR necesaria para el despliegue 5G. Para ello se realiza en primer lugar una revisión de los requisitos para el despliegue de un sistema 5G mediante el uso de las plataformas mencionadas anteriormente (Openairinterface5G, srsRAN).

Tras la revisión de requisitos, se concluye que la plataforma SDR “USRP” desarrollada por Ettus Research es la escogida por todos los proyectos de código abierto

relacionados con la interfaz radio de 5G. Esto seguramente se deba al uso de un controlador único, el *driver* UHD (USRP Hardware Driver™), para todos los modelos disponibles de esta plataforma. Esta característica de la plataforma SDR “USRP” permite desarrollar y hacer pruebas con sus modelos más básicos (B200, B210), para después implementar soluciones con sus modelos de más alta gama (X310, N310).

Finalmente, se lleva la propuesta de compra del USRP B210 a la Universidad, que la acepta. Una vez adquirida la plataforma SDR USRP B210, se procede al despliegue con éxito de la primera solución RAN explorada (srsRAN).

Además, la Universidad de Jaén adquiere, a su vez, dos UE compatibles con 5G SA. En concreto, se trata de los terminales Xiaomi Redmi 12 5G y el Motorola moto g73 5G que cuentan con las siguientes características principales:

	Xiaomi Redmi 12 5G	Motorola Moto g73 5G
Pulgadas	6.79”	6.5”
Resolución	FHD+	FHD+
Superficie útil	85% útil	85% útil
Relación aspecto	21:9	20:9
Dimensiones (Ancho x Alto x Grosor) (mm)	76.3 x 168.6 x 8.2	73.8 x 161.4 x 8.3
Peso	199 g	181 g
Procesador	SnapD, 4 Gen2. Octa-Core	Dimensity 930. Octa-Core
RAM	4 GB	8 GB
Almacenamiento	128 GB	256 GB
Batería	5000 mAh	5000 mAh
Cámara principal	50 Mpx – f/1.8	50 Mpx – f/1.8
Cámara frontal	5 Mpx – f/2.2	16 Mpx – f/2.4
Bandas	2G / 3G / 4G / 5G	2G / 3G / 4G / 5G
Sistema Operativo	Android 13 Tiramisu	Android 13 Tiramisu

Tabla 7: Características de los terminales móviles Xiaomi Redmi 12 5G y Motorola moto g73 5G.

Sin embargo, la característica más importante que debemos conocer sobre estos terminales móviles es las bandas de 5G que soportan. Estas bandas se resumen en la siguiente tabla.

Banda	Nombre	Modo	Ancho de banda DL / UL (MHz)	DL Frecuencia central (MHz)	UL Frecuencia central (MHz)	Xiaomi Redmi 12 5G	Motorola Moto g73 5G
n1	2100	FDD	60	2140	1950	✓	✓
n3	1800	FDD	75	1842.5	1747.5	✓	✓
n5	850	FDD	25	881.5	836.5	✓	✓
n7	2600	FDD	70	2655	2535	✓	✓
n8	800 GSM	FDD	35	942.5	897.5	✓	✓
n20	800	FDD	30	806	847	✓	✓
n28	700 APT	FDD	45	780.5	725.5	✓	✓
n38	TD 2600	TDD	50	2595	-	✓	✓
n40	TD 2300	TDD	100	2350	-	✓	✓
n41	TD 2600+	TDD	194	2593	-	✓	✓
n66	AWS	FDD	90 / 70	2155	1745	✓	✗
n77	TD 3700	TDD	900	3750	-	✓	✓
n78	TD 4700	TDD	500	3550	-	✓	✓

Tabla 8: Bandas de frecuencia de NR compatibles con los terminales móviles Xiaomi Redmi 12 5G y Motorola Moto g73 5G.

Para poder comprobar que el despliegue de la red 5G funciona correctamente, se programan varias tarjetas sysmoSIM-SJA2. Se trata de tarjetas SIM programables de la marca sysmocom. En los anexos se encuentra un documento con la guía detallada de la configuración de estas tarjetas SIM.

Tras configurar el sistema 5G y añadir la información de las tarjetas SIM a las bases de datos del núcleo de red y subsistema multimedia, podemos encender los terminales móviles y acceder a la red.

Una vez encendidos los terminales comprobamos que tenemos cobertura de la red desplegada, sin embargo aún no tenemos servicio de voz o datos. Para obtenerlo debemos

configurar correctamente los nombres de punto de acceso que hayamos configurado previamente en el núcleo de red.

Para ello accedemos a la configuración de los nombres de punto de acceso o APN del terminal móvil y añadimos dos nuevos perfiles, uno para acceder a la red de datos y el otro para acceder al subsistema multimedia y disfrutar del servicio de SMS. Los perfiles deberán seguir la siguiente configuración:

Nombre	internet	ims
APN	internet	ims
Tipo	default	ims
Protocolo del APN	IPv4	IPv4

Tabla 9: Configuración APN en los terminales móviles.

Tras realizar la configuración podemos comprobar que funcionan correctamente los servicios de datos y SMS, sin embargo, el servicio de voz no se encuentra disponible. Esto es debido a que el subsistema multimedia implementado en el despliegue de la red 5G, Kamailio IMS, no es compatible aún con VoNR. Además, no es posible utilizar en este caso VoLTE debido a que se ha desplegado una red 5G SA, que por definición solo soporta servicios propios de 5G. Aun así, sería posible cursar una llamada mediante VoLTE mediante un *fallback* a otra red 4G o 5G NSA desplegada.

4.1. docker_open5gs

Se trata de un proyecto de código abierto mantenido por el usuario herlesupreeth que ofrece una solución Docker para la implementación de un sistema 4G o 5G SA/NSA. Este proyecto incluye a su vez a los siguientes proyectos de código abierto:

- **Open5GS.** Núcleo de red.
- **Kamailio IMS.** Subsistema multimedia.
- **srsRAN.** Red de Acceso Radio 4G y 5G.
- **UERANSIM.** Simulador de la RAN.

El objetivo de este Trabajo Fin de Grado es el de ofrecer una herramienta didáctica con la que realizar distintas prácticas de laboratorio relacionadas tanto con el núcleo de red como con la interfaz radio. Es por ello que se realizarán una serie de modificaciones a este proyecto para conseguirlo. Se adjunta como anexo una guía en la que se detalla la puesta en marcha y configuración inicial de todo el sistema 5G.

4.1.1. GitHub workflows

En concreto, uno de los principales problemas de este trabajo es que debemos de compilar todas las imágenes en el propio equipo, lo que ralentiza el despliegue de la red. Por ello, la primera modificación consiste en la compilación de todas las imágenes en la nube.

Para ello se utiliza la propia plataforma GitHub, que ofrece este tipo de soluciones en conjunto con Docker Hub, un repositorio público de imágenes Docker.



Figura 26: Logotipo de GitHub.

En primer lugar, debemos de registrarnos en Docker Hub (<https://hub.docker.com/>).

A continuación, accedemos al repositorio en GitHub y seleccionamos la pestaña *Actions* de la barra superior. Una vez ahí, pulsamos en *New workflow* y nos aparecerán distintas plantillas. Escogeremos la plantilla *Docker image*. En caso de que no nos aparezca, podremos buscarla con la barra de búsqueda.

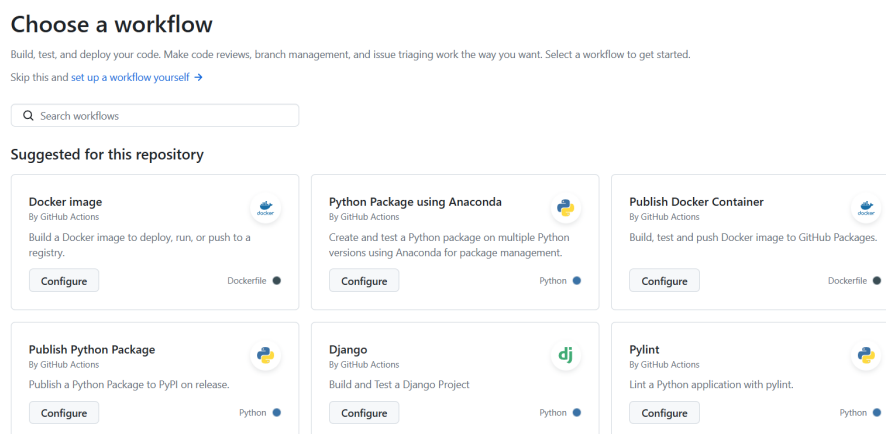


Figura 27: Ventana "Choose a workflow" de GitHub.

Entonces, se creará un nuevo archivo que podremos modificar para compilar las imágenes de nuestro trabajo.

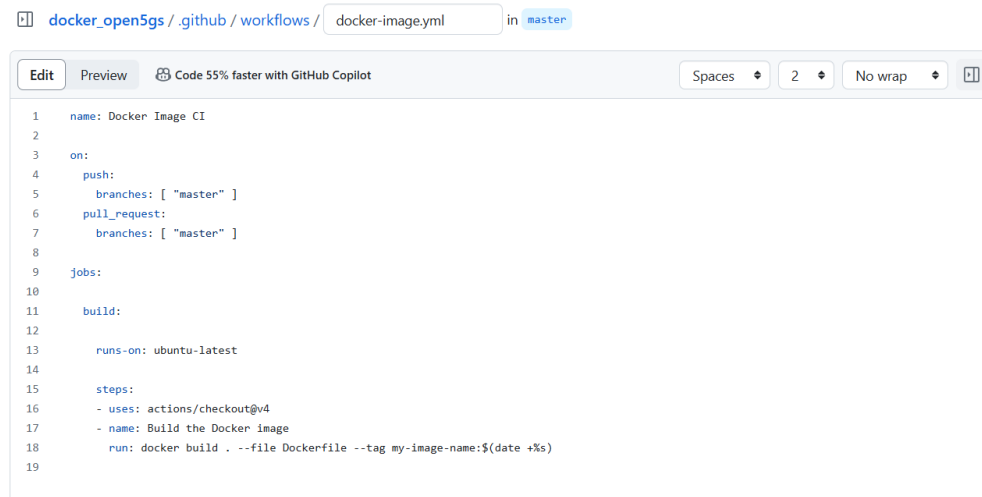


Figura 28: Edición de un workflow en GitHub.

Esta plantilla solo compilará la imagen Docker, por lo que debemos añadirle los pasos necesarios para poder subir esa imagen al repositorio Docker Hub. Para ello, partiremos de la siguiente plantilla modificada:

```
name: Docker open5gs

on:
  push:
    paths:
      - base/Dockerfile          # Ruta al archivo Dockerfile de la imagen
  workflow_dispatch:            # Necesario para ejecutar manualmente

jobs:

  build:

    runs-on: ubuntu-latest

    steps:
      - name: Check out the repo
        uses: actions/checkout@v3

      - name: Log in to Docker Hub
        uses: docker/login-
action@f4ef78c080cd8ba55a85445d5b36e214a81df20a
        with:
```

```

    username: ${ secrets.DOCKER_USERNAME }
    password: ${ secrets.DOCKER_PASSWORD }

- name: Build the Docker image
  run: cd base && docker build --no-cache --force-rm -t
juanluss31/docker_open5gs:latest .

- name: Push Image
  run: docker push juanluss31/docker_open5gs:latest

```

En primer lugar, es importante que el nombre de la imagen siga el siguiente esquema:

Nombre de usuario de Docker Hub
juanluss31 / docker_open5gs : latest
Nombre de la imagen docker

Por otro lado, si se presta atención al paso en el que se inicia sesión en Docker Hub, no hay ningún usuario y contraseña escritos en el script. Esto es debido a que se utiliza *GitHub secrets*, una característica de GitHub que nos permite ocultar información sensible en este tipo de scripts.

Para configurarlo debemos de ir a los ajustes del repositorio y pulsar en la opción *Actions* dentro de *Secrets and variables*.

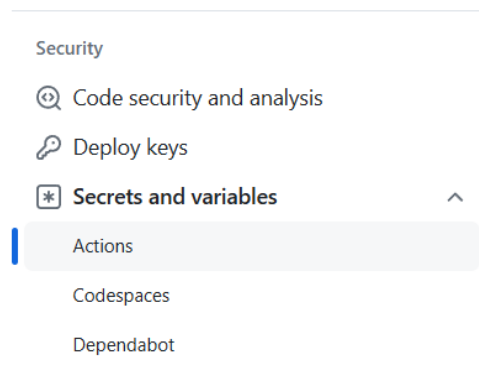


Figura 29: Opciones de seguridad del repositorio en GitHub.

Allí, podremos configurar el nombre de nuestra variable secreta y su contenido. En este caso se han creado dos variables secretas, “DOCKER_USERNAME” y “DOCKER_PASSWORD”.

Una vez hayamos configurado la compilación de imágenes Docker en la nube, es necesario modificar todos los archivos *Docker compose* para que hagan referencia a las imágenes que hemos creado. Así, cuando vayamos a desplegar el sistema 5G, buscará las imágenes Docker en la nube y no en local, donde deberíamos haberlas compilado previamente.

4.1.2. Portainer

Se trata de una herramienta de gestión de contenedores que puede funcionar tanto con Docker como con Kubernetes para hacer que la implementación y gestión de aplicaciones y servicios en contenedores sea más fácil y eficiente.

En nuestro caso la utilizaremos para comprobar que todos los contenedores funcionan correctamente y poder comprobar en cualquier momento los puertos que cada uno tiene abiertos, algo realmente útil ya que contamos con diferentes servicios web en todo el sistema 5G.

Portainer cuenta con dos versiones distintas, una versión premium conocida como *Business Edition* (BE) y otra, gratuita y de código abierto, conocida como *Community Edition* (CE). En nuestro caso trabajaremos con la versión gratuita y de código abierto.

4.1.2.1. Instalación

En primer lugar, en el equipo que está ejecutando el núcleo del sistema 5G, creamos el volumen que Portainer utilizará para guardar su base de datos:

```
docker volume create portainer_data
```

Después, descargamos e instalamos el contenedor de Portainer:

```
docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always  
-v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data  
portainer/portainer-ce:latest
```

En cambio, si estamos utilizando Windows, podemos instalar Portainer fácilmente como una extensión de Docker Desktop, para ello simplemente buscamos la extensión y la instalamos.

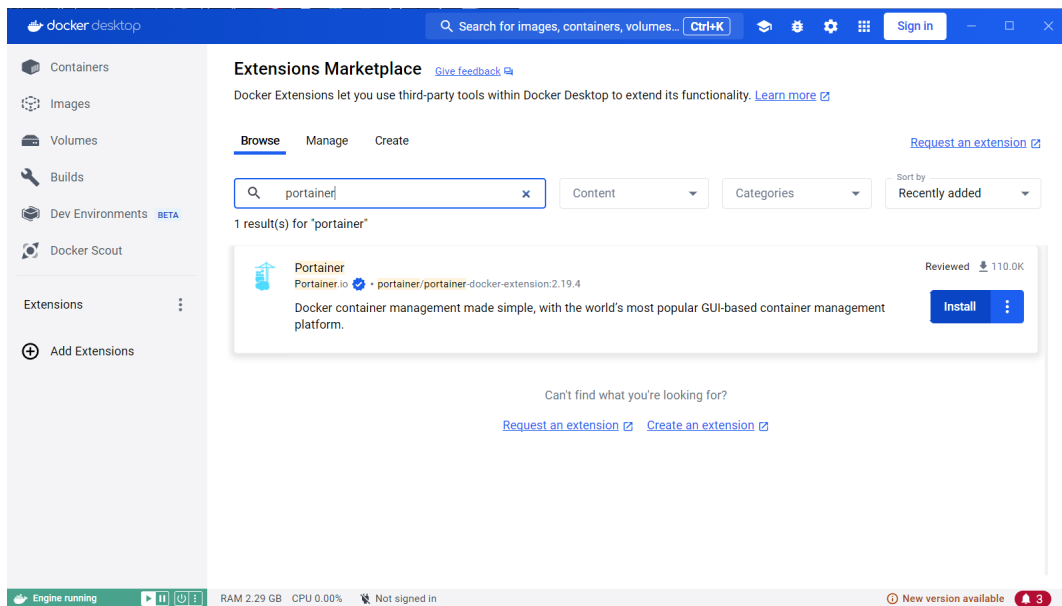


Figura 30: Instalación de Portainer como una extensión de Docker Desktop.

4.1.2.2. Configuración inicial

En el caso de que hayamos realizado la instalación en Linux, la interfaz de Portainer se encuentra disponible en la dirección `https://<DOCKER_HOST_IP>:9443`, siendo `<DOCKER_HOST_IP>` la dirección IP del equipo que está ejecutando el núcleo de red del sistema 5G.

Si la instalación ha sido como una extensión de Docker Desktop en Windows, podremos acceder a la interfaz de Portainer desde la propia aplicación de Docker Desktop.

Tras acceder, lo primero que deberemos configurar es el usuario administrador, aunque si se trata de la extensión de Docker Desktop este paso es innecesario ya que la Portainer no crea ningún usuario al ser la única forma de acceso desde la aplicación.



▼ New Portainer installation

Please create the initial administrator user.

Username

Password

Confirm password ☒

⚠ The password must be at least 12 characters long. ✓

[Create user](#)

☒ Allow collection of anonymous statistics. You can find more information about this in our [privacy policy](#).

[> Restore Portainer from backup](#)

Figura 31: Configuración inicial de Portainer, creación de usuario administrador.

Una vez configurado nos aparecerá un asistente para conectar Portainer al entorno Docker que deseemos. En nuestro caso, pulsaremos sobre *Get Started* que simplemente cargará el entorno en el que se está ejecutando Portainer.

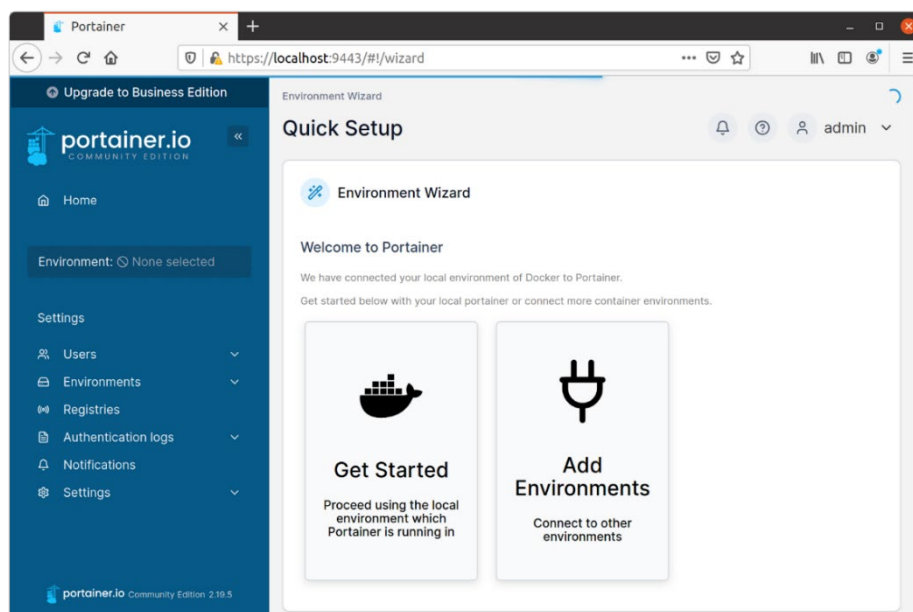


Figura 32: Configuración inicial de Portainer, 'Environment Wizard'.

4.1.2.3. Uso

Nada más iniciar sesión con la cuenta de administrador creada anteriormente, nos aparecerá esta pantalla. Para poder ver los contenedores que estamos ejecutando en Docker, pulsamos en el *environment* local.

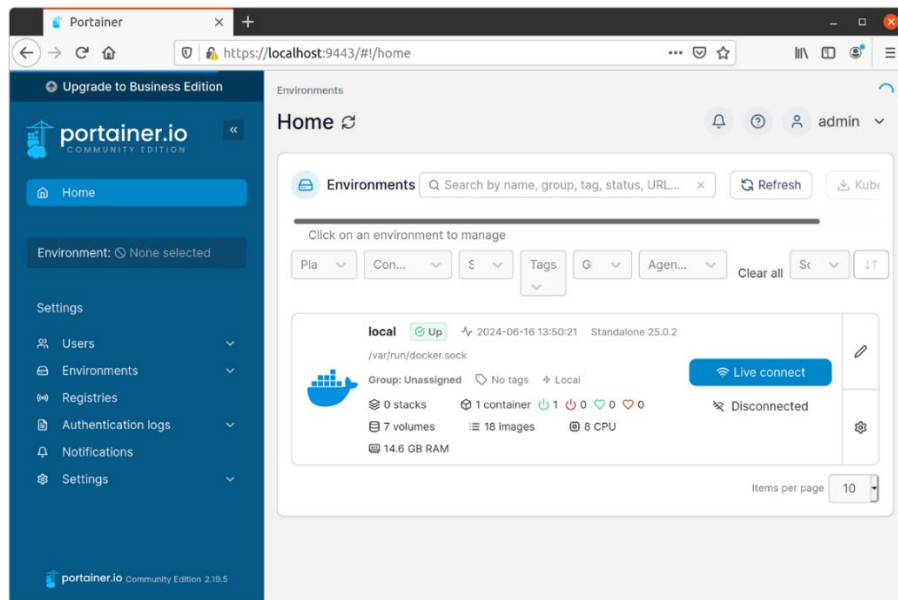


Figura 33: Pantalla principal de Portainer, selección de entorno Docker.

Una vez conectados al entorno Docker, podremos explorar los distintos *stacks*, contenedores, imágenes, volúmenes y redes que existan.

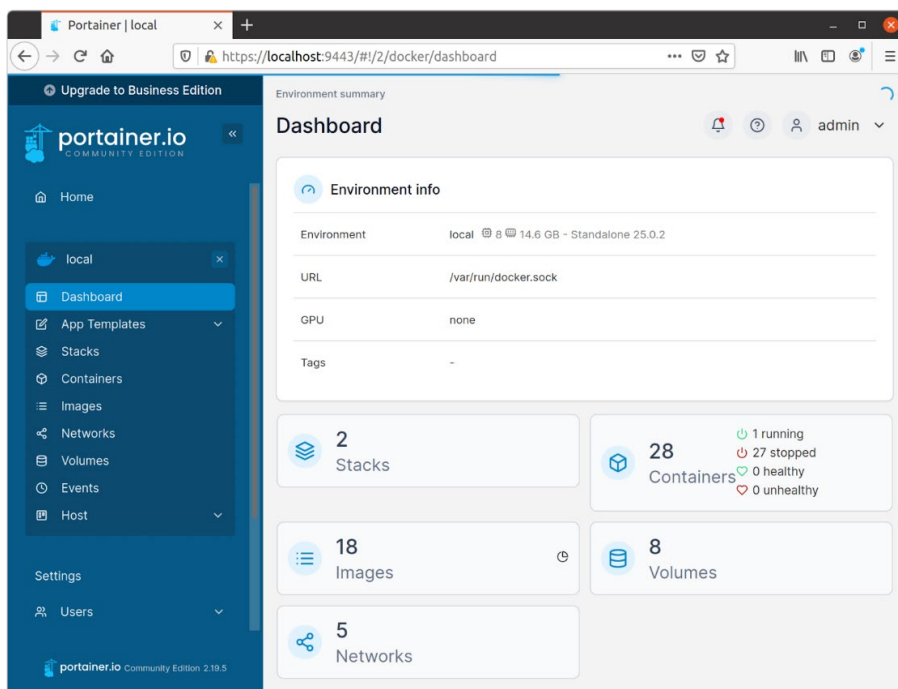


Figura 34: Información general sobre el entorno Docker al que nos hemos conectado con Portainer.

En nuestro caso, lo más interesante sería explorar la lista de contenedores que forman el *stack* “docker_open5gs”, ya que podemos comprobar de un vistazo que todo funciona correctamente, los puertos expuestos al exterior de cada contenedor y el *log* de cada uno de ellos.

Name	State	Quick Actions	Stack	Image	Created	IP Address	Published Ports	Ownership
amf	running	[Icons]	docker_open5gs	juanluss31/docker_open5gs:latest	2024-06-16 13:54:39	172.22.0.10	-	administrators
ausf	running	[Icons]	docker_open5gs	juanluss31/docker_open5gs:latest	2024-06-16 13:54:38	172.22.0.11	-	administrators
bsf	running	[Icons]	docker_open5gs	juanluss31/docker_open5gs:latest	2024-06-16 13:54:39	172.22.0.29	-	administrators
dns	running	[Icons]	docker_open5gs	juanluss31/docker_dns:latest	2024-06-16 13:54:38	172.22.0.15	-	administrators
grafana	running	[Icons]	docker_open5gs	grafana/grafana-oss:10.1.5	2024-06-16 13:54:39	172.22.0.39	3300:3000	administrators
icscf	running	[Icons]	docker_open5gs	juanluss31/docker_icscf:latest	2024-06-16 13:54:39	172.22.0.19	-	administrators
influxdb	running	[Icons]	docker_open5gs	influxdb:2.7	2024-06-16 13:54:38	172.22.0.38	-	administrators
metrics	running	[Icons]	docker_open5gs	juanluss31/docker_metrics:latest	2024-06-16 13:54:38	172.22.0.36	9090:9090	administrators
metrics_server	running	[Icons]	docker_open5gs	juanluss31/srsran_metrics_server:latest	2024-06-16 13:54:38	172.22.0.40	55555:55555	administrators
mongo	running	[Icons]	docker_open5gs	mongo:6.0	2024-06-16 13:54:38	172.22.0.2	-	administrators

Figura 35: Información de los contenedores que existen en el entorno Docker.

4.2. Herramientas para la realización de prácticas con la interfaz radio

4.2.1. Análisis de la interfaz radio

En el caso de la realización de prácticas con la interfaz radio, se ha procedido a la incorporación de un panel de visualización en tiempo real de estadísticas proporcionadas por la plataforma srsRAN.

Para ello, se han añadido al archivo “docker-compose.yml” tres nuevos contenedores:

- **Grafana.** Se trata de un software libre que permite la visualización y el formato de datos métricos. Permite crear cuadros de mando y gráficos a partir de múltiples fuentes, incluidas bases de datos de series de tiempo como Graphite, InfluxDB y OpenTSDB (40).
- **InfluxDB.** Se trata de un base de datos de series temporales (TSDB en inglés) de código abierto. Se utiliza para el almacenamiento y recuperación de datos de series temporales en campos como monitoreo de operaciones, métricas de aplicaciones, datos de sensores IoT, y análisis en tiempo real (41).

- **Metrics Server.** Se trata de un servidor creado por srsRAN para recibir los datos de las métricas proporcionados por un gNodeB de srsRAN Project y enviarlos a una base de datos InfluxDB.



Figura 36: Logotipo de Grafana.

Además, se ha añadido al archivo “.env” los siguientes parámetros de configuración necesarios para el despliegue de estos contenedores:

```
# Srsran Metrics:
## InfluxDB

DOCKER_INFLUXDB_VERSION=2.7

DOCKER_INFLUXDB_INIT_MODE=setup
DOCKER_INFLUXDB_INIT_USERNAME=admin
DOCKER_INFLUXDB_INIT_PASSWORD=admin1234
DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=605bc59413b7d5457d181ccf20f9fda15693f81b
068d70396cc183081b264f3b
DOCKER_INFLUXDB_INIT_ORG=srs
DOCKER_INFLUXDB_INIT_BUCKET=srsran
DOCKER_INFLUXDB_INIT_RETENTION=4d

DOCKER_INFLUXDB_INIT_PORT=8086
DOCKER_INFLUXDB_INIT_HOST=influxdb

INFLUXD_LOG_LEVEL=error

INFLUXDB_IP=172.22.0.38

## Grafana
```

```
GRAFANA_VERSION=10.1.5
GRAFANA_PORT=3000

GF_AUTH_ANONYMOUS_ENABLED=true
GF_LOG_LEVEL=warning

GF_DASHBOARDS_DEFAULT_HOME_DASHBOARD_PATH=/etc/dashboards/home.json
GF_DASHBOARDS_MIN_REFRESH_INTERVAL=1s

GRAFANA_DATASOURCE_INFLUXDB_UID=JOSE3g9KVz
GRAFANA_DATASOURCE_PROMETHEUS_UID=b0b4ba4e-1c4f-4c1a-9168-37458f0e1fbd

GRAFANA_IP=172.22.0.39
```

Por último, se ha modificado el panel principal de Grafana para mostrar las distintas métricas que proporciona el gNodeB. En concreto, la información que nos proporciona este panel es:

- **Active UEs.** Informa del número de UE conectados al gNodeB.
- **Active UEs by APN.** Informa del número de UE conectados al gNodeB diferenciando el APN al que están conectados.
- **RI (Rank Indicator).** Informa del número de antenas que se están utilizando en un entorno MIMO.
- **Maximum Total Downlink Bitrate.** Informa de la tasa de bits máxima alcanzada por el enlace descendente del gNodeB.
- **Current Total Downlink Bitrate.** Informa de la tasa de bits del enlace descendente del gNodeB.
- **Current Total Uplink Bitrate.** Informa de la tasa de bits del enlace ascendente del gNodeB.
- **Maximum Total Uplink Bitrate.** Informa de la tasa de bits máxima alcanzada por el enlace ascendente del gNodeB.
- **Quality metrics:**
 - **CQI.** Se trata del *Channel Quality Indicator*, un parámetro que informa de la calidad del canal radio desde el punto de vista del UE.
 - **Uplink SNR.** Relación señal-ruido del enlace ascendente de cada UE medido desde el gNodeB.

- **Downlink MCS.** Se trata del *Modulation Coding Scheme*, un parámetro que informa de la modulación y codificación utilizadas para el enlace descendente.
- **Uplink MCS.** Se trata del *Modulation Coding Scheme*, un parámetro que informa de la modulación y codificación utilizadas para el enlace ascendente.
- **Bitrate metrics:**
 - **Downlink Bitrate.** Informa de la tasa de bits del enlace descendente de cada UE conectado al gNodeB.
 - **Uplink Bitrate.** Informa de la tasa de bits del enlace ascendente de cada UE conectado al gNodeB.
 - **Downlink BS.** Informa del tamaño del buffer del enlace descendente de cada UE conectado al gNodeB.
 - **BSR.** Informa del tamaño del buffer del enlace ascendente de cada UE conectado al gNodeB.
- **Lost Packets:**
 - **Downlink % Packets Not OK.** Informa del porcentaje de paquetes perdidos en el enlace descendente.
 - **Uplink % Packets Not OK.** Informa del porcentaje de paquetes perdidos en el enlace ascendente.
 - **Downlink Packets.** Informa del número de paquetes enviados correctamente e incorrectamente en el enlace descendente de cada UE conectado al gNodeB.
 - **Uplink Packets.** Informa del número de paquetes enviados correctamente e incorrectamente en el enlace ascendente de cada UE conectado al gNodeB.

Para acceder a este panel de control podemos hacerlo desde la dirección http://<DOCKER_HOST_IP>:3000, siendo <DOCKER_HOST_IP> la dirección IP del equipo que está ejecutando el núcleo de red del sistema 5G.

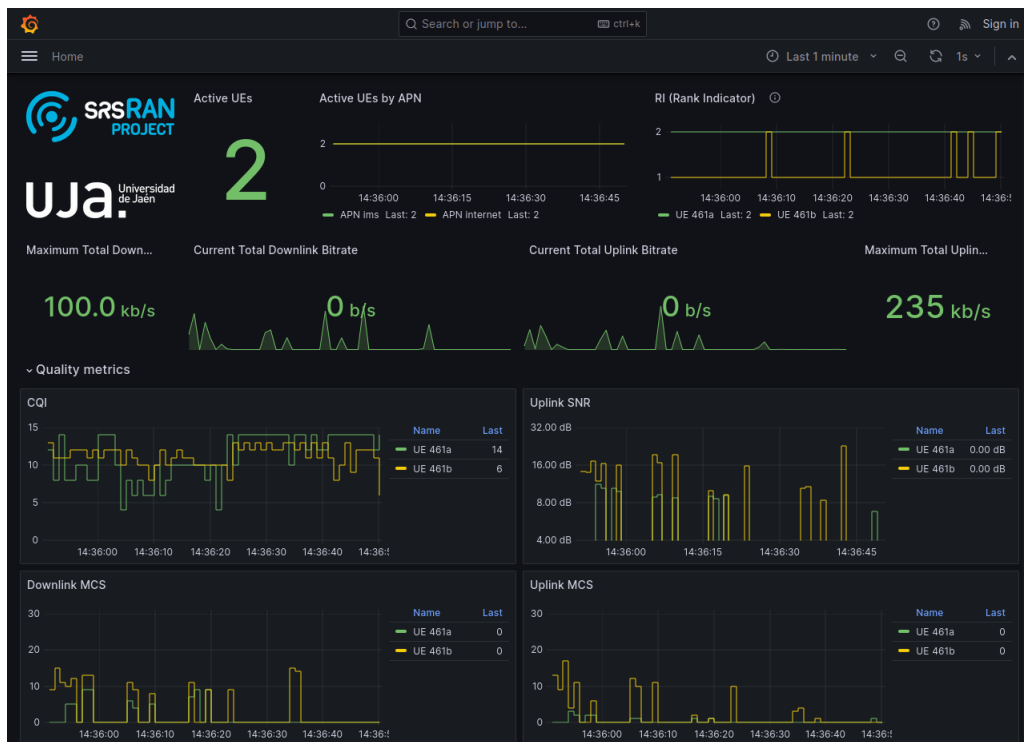


Figura 37: Panel principal de Grafana donde se muestran las métricas ofrecidas por srsRAN en tiempo real.

4.3. Herramientas para la realización de prácticas con el núcleo de red

4.3.1. Análisis de red

En cuanto a las prácticas relacionadas con el núcleo de red se propone realizar análisis de tráfico con la herramienta Wireshark. Sin embargo, al tratarse de una red virtualizada en Docker, Wireshark no siempre detecta correctamente estas interfaces virtuales de red.

Por ello se propone utilizar Edgeshark, una herramienta diseñada por Siemens que permite visualizar las conexiones de la red Docker y analizar el tráfico en ella. Para ello captura el tráfico de la red y lo envía remotamente a Wireshark. Esto permite realizar prácticas donde se encuentre un equipo del laboratorio que ejecute el núcleo de red 5G, y que los alumnos puedan analizar el tráfico de la red en tiempo real desde sus puestos de laboratorio.

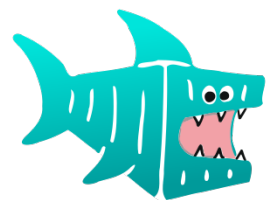


Figura 38: Logotipo de Edgeshark.



Figura 39: Logotipo de Ghostwire.

Para su implementación, se ha añadido al proyecto `docker_open5gs` un archivo Docker compose llamado `"edgeshark.yaml"`. Este archivo es el encargado de ejecutar los contenedores `"gostwire"` (encargado de ejecutar la interfaz web de

esta herramienta) y “edgeshark” (encargado del envío del tráfico de red a Wireshark).

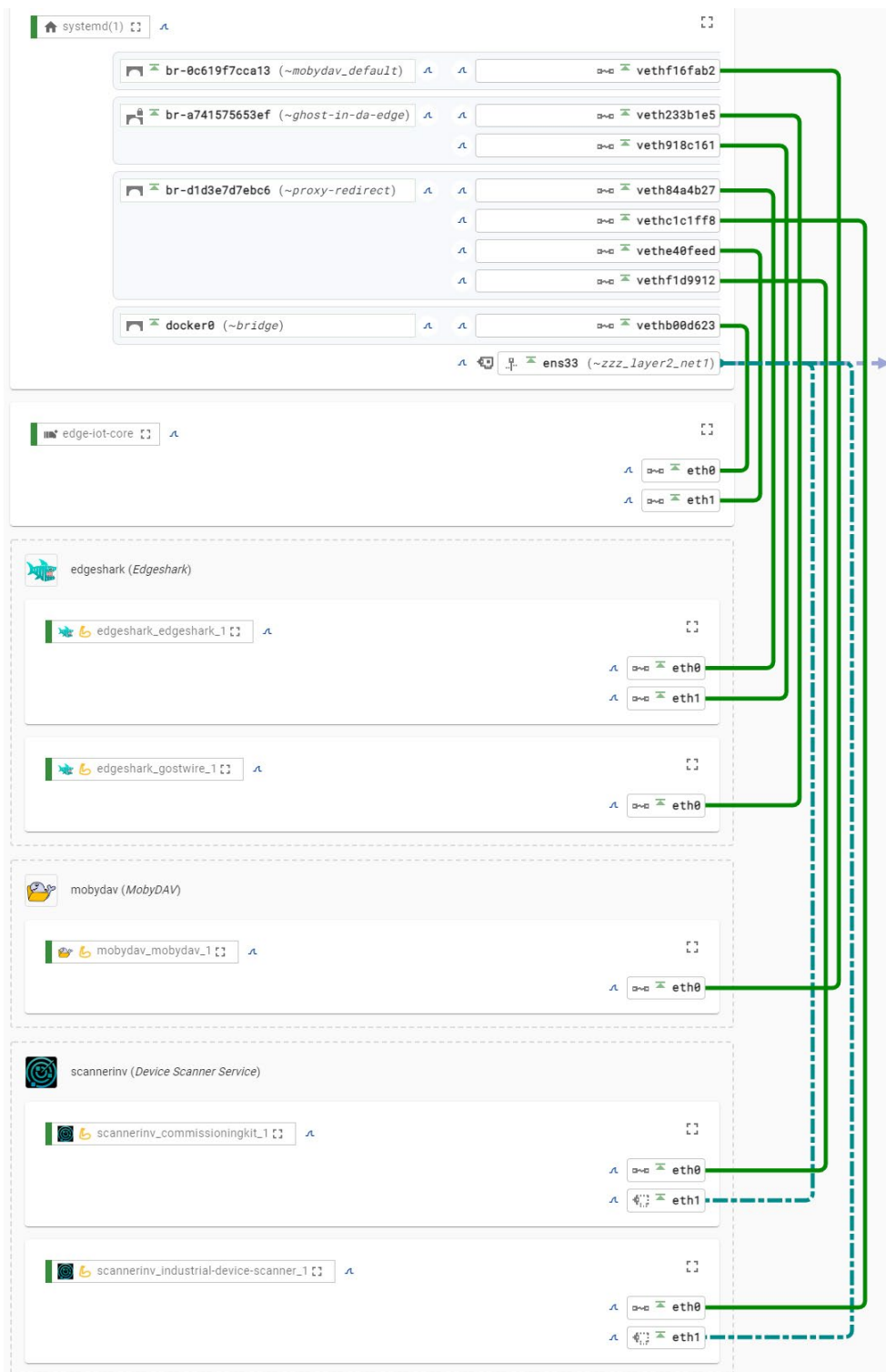


Figura 40: Visualización de conexiones de red con la herramienta Ghostwire.

En la figura anterior se muestra cómo aparece una red de ejemplo en la interfaz de Ghostwire. Aparecen todos los contenedores de la red con sus respectivas interfaces, a su vez estas interfaces se conectan a través de cables de color verde entre sí. Pulsando en el botón con forma de aleta de tiburón podemos capturar en tiempo real el tráfico de ese contenedor o interfaz.

Además, es necesario instalar el plugin cshargextcap (Containershark Extcap) para poder recibir el tráfico generado por Edgeshark en la aplicación Wireshark instalada en nuestro equipo.



Figura 41: Logotipo del plugin cshargextcap.

Para ello descargamos e instalamos la versión correspondiente al sistema operativo con el que estemos trabajando desde el repositorio de esta herramienta: <https://github.com/siemens/cshargextcap/releases/>.

Ahora, cada vez que queramos ver el tráfico de un contenedor o interfaz concreto nos aparecerá la siguiente ventana

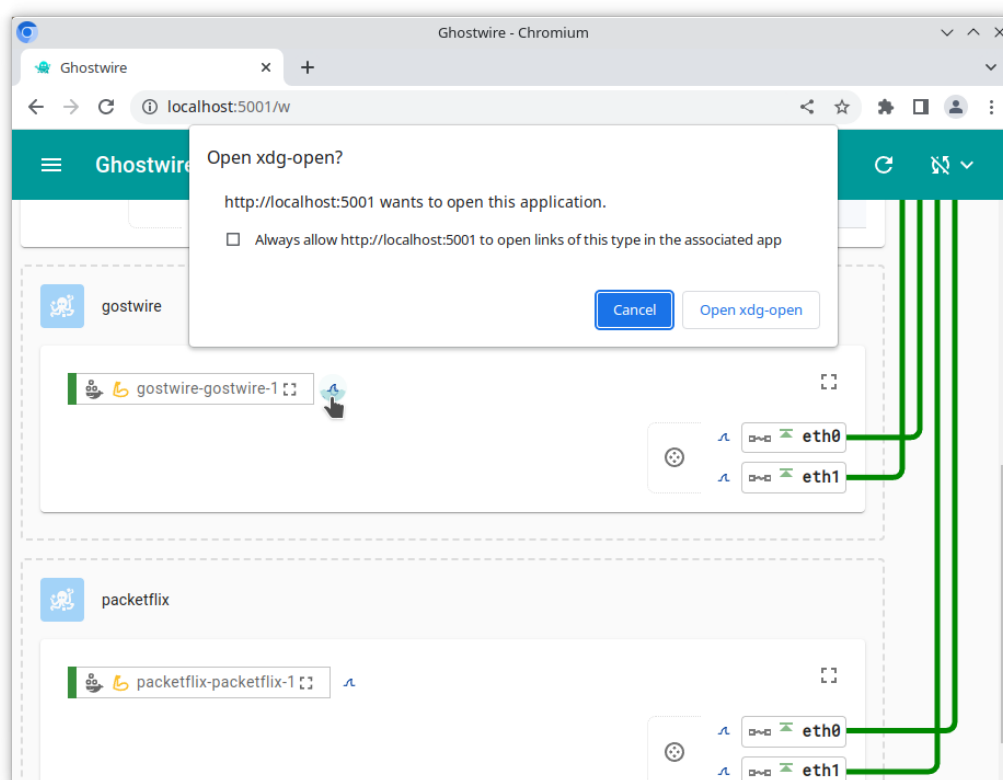


Figura 42: Detalle de la ventana de diálogo generada por la herramienta Ghostwire que abre la aplicación Wireshark para capturar el tráfico de un contenedor Docker.

La interfaz de Ghostwire se encuentra disponible en la dirección `http://<DOCKER_HOST_IP>:5001`, siendo `<DOCKER_HOST_IP>` la dirección IP del equipo que está ejecutando el núcleo de red del sistema 5G.

Otra forma de acceder al tráfico de un contenedor de la red Docker es utilizando la propia aplicación Wireshark sin necesidad de abrir la interfaz web de Gostwire. Para ello debemos pulsar en la rueda dentada que hay a la izquierda del nombre de la interfaz 'Docker host capture'. Después debemos introducir la url del contenedor Edgeshark (la

misma que la interfaz web de Ghostwire) y recargamos la lista de contenedores. Por último seleccionamos el contenedor donde queremos analizar el tráfico y pulsamos en comenzar.

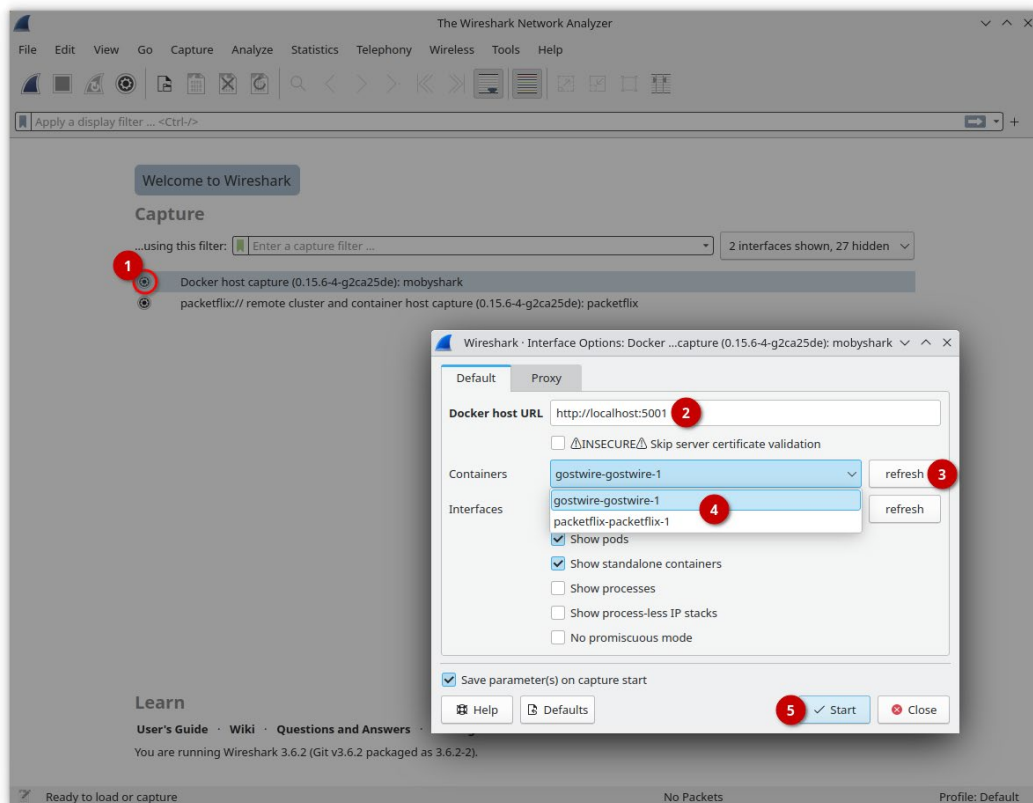


Figura 43: Pasos a seguir para capturar el tráfico de un contenedor Docker directamente desde Wireshark gracias al plugin cshargextcap.

4.3.2. Simulación de la interfaz radio

Para permitir realizar prácticas relacionadas con el núcleo de red sin necesidad de utilizar srsRAN para implementar la interfaz radio y utilizar terminales móviles reales, se opta por incluir una solución RAN virtualizada. Se trata de UERANSIM, que incluye un gNodeB virtual y un UE virtual.

Estos elementos se pueden desplegar ejecutando los archivos *Docker compose* “nr-gnb.yaml” y “nr-ue.yaml”. La configuración básica de UERANSIM se encuentra en el archivo “.env” situado en la raíz del repositorio.

```
# UERANSIM
NR_GNB_IP=172.22.0.23
NR_UE_IP=172.22.0.24

UE1_IMEI=356938035643803
UE1_IMEISV=4370816125816151
```

```
UE1_IMSI=001010000053537
UE1_KI=95B2689668A3C24443CF45CB7781F066
UE1_OP=577CB1FDAA2F8C39C4D7618D249C36E9
UE1_AMF=8000
```

En primer lugar desplegamos el gNB mediante el comando:

```
# UERANSIM NR-gNB (RF simulated)
## Linux Host
docker compose -f nr-gnb.yaml up -d && docker container attach nr_gnb

## Windows Host
docker compose -f nr-gnb.yaml up -d ; if($?) {docker container attach
nr_gnb}
```

Después desplegamos, en otra terminal, el UE mediante el comando:

```
# UERANSIM NR-UE (RF simulated)
## Linux Host
docker compose -f nr-ue.yaml up -d && docker container attach nr_ue

## Windows Host
docker compose -f nr-ue.yaml up -d ; if($?) {docker container attach
nr_ue}
```

En los mensajes de log de ambos elementos podemos ver información relacionada sobre el estado de registro en la red de ambos elementos.

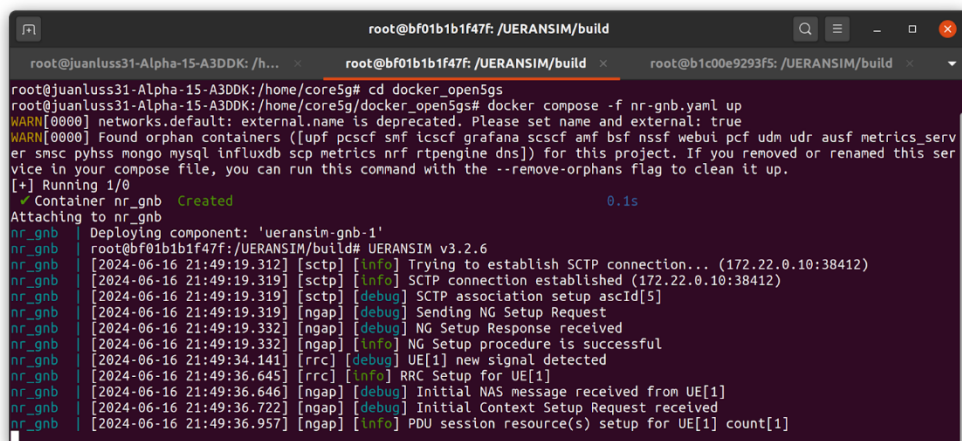


Figura 44: Log del gNB simulado con UERANSIM donde se muestra que está registrado en la red y que un UE se ha registrado en él.

```
root@b1c00e9293f5: /UERANSIM/build
root@juanluss31-Alpha-15-A3DDK: /home/core5g/dock...
root@b1c00e9293f5: /UERANSIM/build
root@b1c00e9293f5: /UERANSIM/build

core5g@juanluss31-Alpha-15-A3DDK:~$ sudo -s
[sudo] contraseña para core5g:
root@juanluss31-Alpha-15-A3DDK: /home/core5g# cd docker_open5gs
root@juanluss31-Alpha-15-A3DDK: /home/core5g/docker_open5gs# docker compose -f nr-ue.yaml up
WARN[0000] networks.default: external.name is deprecated. Please set name and external: true
WARN[0000] Found orphan containers ([nr_gnb upf pcsf scf lcsf grafana scscf anf bsf nssf webui pcf udm udr ausf metrics_server smsc pyhss mongo mysql influxdb
scp metrics nrf rtengine dns]) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans
flag to clean it up.
[+] Running 1/0
 * Container nr_ue Created                                0.1s
Attaching to nr_ue
nr_ue | Deploying component: 'ueransim-ue-1'
nr_ue | root@b1c00e9293f5: /UERANSIM/build# UERANSIM v3.2.6
nr_ue | [2024-06-16 21:49:34.141] [nas] [info] UE switches to state [MM-DEREGISTERED/PLMN-SEARCH]
nr_ue | [2024-06-16 21:49:34.141] [rrc] [debug] New signal detected for cell[1], total [1] cells in coverage
nr_ue | [2024-06-16 21:49:34.142] [nas] [info] Selected plmn[001/01]
nr_ue | [2024-06-16 21:49:36.641] [rrc] [info] Selected cell plmn[001/01] tac[1] category[SUITABLE]
nr_ue | [2024-06-16 21:49:36.641] [nas] [info] UE switches to state [MM-DEREGISTERED/PS]
nr_ue | [2024-06-16 21:49:36.641] [nas] [info] UE switches to state [MM-DEREGISTERED/NORMAL-SERVICE]
nr_ue | [2024-06-16 21:49:36.641] [nas] [debug] Initial registration required due to [MM-DEREG-NORMAL-SERVICE]
nr_ue | [2024-06-16 21:49:36.641] [nas] [debug] UAC access attempt is allowed for identity[0], category[M0_sig]
nr_ue | [2024-06-16 21:49:36.641] [nas] [debug] Sending Initial Registration
nr_ue | [2024-06-16 21:49:36.645] [nas] [info] UE switches to state [MM-REGISTER-INITIATED]
nr_ue | [2024-06-16 21:49:36.645] [rrc] [debug] Sending RRC Setup Request
nr_ue | [2024-06-16 21:49:36.646] [rrc] [info] RRC connection established
nr_ue | [2024-06-16 21:49:36.646] [nas] [info] UE switches to state [RRC-CONNECTED]
nr_ue | [2024-06-16 21:49:36.646] [nas] [info] UE switches to state [CM-CONNECTED]
nr_ue | [2024-06-16 21:49:36.673] [nas] [debug] Authentication Request received
nr_ue | [2024-06-16 21:49:36.684] [nas] [debug] Security Mode Command received
nr_ue | [2024-06-16 21:49:36.684] [nas] [debug] Selected integrity[2] ciphering[0]
nr_ue | [2024-06-16 21:49:36.723] [nas] [debug] Registration accept received
nr_ue | [2024-06-16 21:49:36.723] [nas] [info] UE switches to state [MM-REGISTERED/NORMAL-SERVICE]
nr_ue | [2024-06-16 21:49:36.723] [nas] [debug] Sending Registration Complete
nr_ue | [2024-06-16 21:49:36.723] [nas] [info] Initial Registration is successful
nr_ue | [2024-06-16 21:49:36.723] [nas] [debug] Sending PDU Session Establishment Request
nr_ue | [2024-06-16 21:49:36.723] [nas] [debug] UAC access attempt is allowed for identity[0], category[M0_sig]
nr_ue | [2024-06-16 21:49:36.926] [nas] [debug] Configuration Update Command received
nr_ue | [2024-06-16 21:49:36.958] [nas] [debug] PDU Session Establishment Accept received
nr_ue | [2024-06-16 21:49:36.958] [nas] [info] PDU Session establishment is successful PSI[1]
nr_ue | [2024-06-16 21:49:36.985] [app] [info] Connection setup for PDU session[1] is successful, TUN interface[uesimtun0, 192.168.100.2] is up.
```

Figura 45: Log del UE simulado con UERANSIM donde se muestra información acerca del intercambio de mensajes con el gNB hasta que se autentica correctamente y crea una conexión a la red proporcionada por el sistema 5G.

Una vez que ambos elementos han sido desplegados podemos hacer pruebas para comprobar que la conexión funciona correctamente. Por ejemplo, podemos hacer un *traceroute* a un servidor externo como “google.es” utilizando la interfaz que se acaba de crear con conexión al APN “internet”.

```
root@40ee07fb252c: /UERANSIM/build# traceroute -i uesimtun0 google.es
traceroute to google.es (142.250.184.163), 30 hops max, 60 byte packets
 1 192.168.100.1 (192.168.100.1) 1.218 ms 1.204 ms 1.242 ms
 2 juanluss31-Alpha-15-A3DDK (172.22.0.1) 1.390 ms 1.496 ms 1.561 ms
 3 www.adsl.vf (192.168.0.1) 5.744 ms 5.691 ms 5.654 ms
 4 * * *
 5 * * *
 6 * * *
 7 * * *
 8 212.166.147.46 (212.166.147.46) 22.911 ms 22.938 ms 22.898 ms
 9 142.251.53.181 (142.251.53.181) 22.858 ms 192.178.110.153 (192.178.110.153) 28.598 ms 28.599 ms
10 142.250.213.125 (142.250.213.125) 28.562 ms 142.250.213.127 (142.250.213.127) 28.560 ms 142.250.213.125 (142.250.213.125) 28.525 ms
11 mad07s23-in-f3.1e100.net (142.250.184.163) 28.508 ms 28.487 ms 28.008 ms
root@40ee07fb252c: /UERANSIM/build#
```

Figura 46: Traceroute a “google.es” ejecutado a través de la interfaz “uesimtun0” en el UE simulado por UERANSIM.

En la figura anterior podemos observar que la primera dirección IP se corresponde con la puerta de enlace del APN “internet” y la segunda dirección IP es la del equipo que ejecuta el sistema 5G, ya que es el proveedor de la conexión a Internet.

Sin embargo, si ejecutamos el mismo comando sin especificar la interfaz a utilizar observaremos que la ruta que siguen los paquetes hasta su destino es la misma que en el caso anterior pero saltándose el primer salto (el sistema 5G). Esto se debe a que el UE simulado es en realidad un contenedor Docker ejecutándose en la misma red que el sistema 5G.

```

root@40ee07fb252c:/UERANSIM/build# traceroute google.es
traceroute to google.es (142.250.184.163), 30 hops max, 60 byte packets
 1  juanluss31-Alpha-15-A3DDK (172.22.0.1)  0.383 ms  0.029 ms  0.021 ms
 2  www.adsl.vf (192.168.0.1)  4.996 ms  4.938 ms  4.883 ms
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  212.166.147.46 (212.166.147.46)  100.853 ms  212.166.147.222 (212.166.147.222)  95.539 ms  95.421 ms
 8  192.178.110.153 (192.178.110.153)  95.364 ms  16.221 ms  142.251.53.181 (142.251.53.181)  19.404 ms
 9  142.250.213.125 (142.250.213.125)  16.061 ms  142.250.213.127 (142.250.213.127)  16.040 ms  15.958 ms
10  mad07s23-in-f3.1e100.net (142.250.184.163)  15.923 ms  19.138 ms  19.122 ms
root@40ee07fb252c:/UERANSIM/build#

```

Figura 47: Traceroute a 'google.es' ejecutado en el UE simulado por UERANSIM.

4.3.2.1. Plugin para Wireshark

UERANSIM no implementa los protocolos de radio 5G por debajo de la capa RRC. Por tanto, las capas PHY, MAC, RLC y PDCP no están implementadas en UERANSIM. La radio 5G se simula parcialmente a través del protocolo UDP a través del puerto 4997, es por ello que es necesario utilizar un disector específico para poder analizar con Wireshark el enlace radio simulado que crea UERANSIM.

Para ello utilizaremos el disector creado por Louis Royer y Ali Güngör, RLS-wireshark-dissector, disponible en <https://github.com/nextmn/RLS-wireshark-dissector>. Simplemente debemos de clonar este repositorio en la carpeta “plugins” del programa Wireshark.

5. RESULTADOS

Finalmente, se ha desplegado un sistema 5G cuya arquitectura se puede resumir en el siguiente esquema.

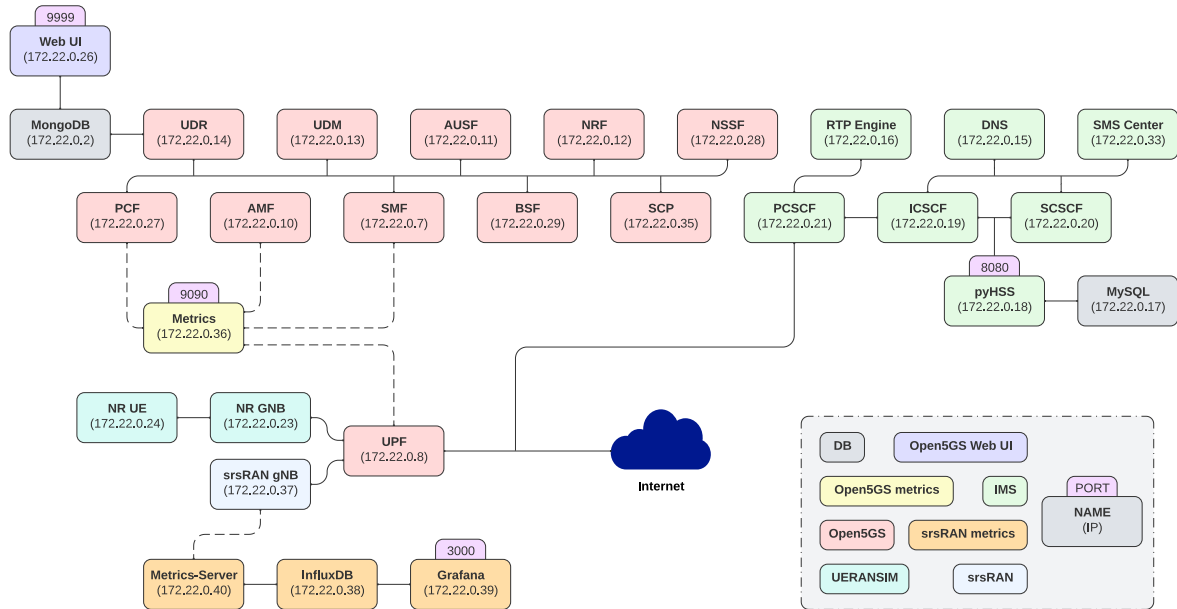


Figura 48: Arquitectura del sistema 5G desplegado.

El esquema anterior no muestra los contenedores de Edgeshark y Ghostwire ya que aunque estos se despliegan en el mismo equipo que el sistema 5G, no se encuentran dentro de la red virtual de este sistema.

5.1. Rendimiento de la red

A la hora de evaluar la interfaz radio, se ha realizado un test de velocidad de internet en uno de los terminales móviles. Para ello, el equipo que ejecuta el núcleo de red se ha conectado a un enlace a internet con una velocidad de 600 Mbps simétricos.

A la hora de escoger los parámetros radio adecuados, se ha realizado un estudio de la banda de frecuencia que se pretende a utilizar. En este estudio se descubre que actualmente la banda 1 está repartida entre cuatro operadores. Cada operador dispone de un ancho de banda de 15 MHz en ambos enlaces, ascendente y descendente.

En base a esta información se realiza un barrido de frecuencia de esta banda en la que encontramos que existe un hueco de 15 MHz sin utilizar en la banda 2140 – 2155 MHz, entre las señales 4G de Orange (2125 – 2140 MHz) y 3G de Movistar (2155 – 2170 MHz), por tanto se centra la frecuencia del enlace descendente en la frecuencia 2147.5 MHz.

Para ello, debemos calcular el ARFCN siguiendo la fórmula descrita en el capítulo 5.4.2.1 de la especificación técnica 38.104 de 3GPP, donde F_{REF} es la frecuencia de referencia en MHz y N_{REF} es el ARFCN buscado.

$$F_{REF} = F_{REF-Offs} + \Delta F_{Global}(N_{REF} - N_{REF-Offs}) \quad (8)$$

El resto de parámetros que aparecen en la fórmula dependen de la banda de frecuencia utilizada y se definen en la siguiente tabla:

Rango de frecuencias (MHz)	ΔF_{Global} (kHz)	$F_{REF-Offs}$ (MHz)	$N_{REF-Offs}$	Rango de N_{REF}
0 – 3000	5	0	0	0 – 599999
3000 – 24250	15	3000	600000	600000 – 2016666
24250 – 100000	60	24250.08	2016667	2016667 – 3279165

Por tanto, el valor de ARFCN vinculado a 2147.5 MHz es:

$$2147.5 = 0 + 5 \cdot 10^{-3}(N_{REF} - 0) \rightarrow N_{REF} = \mathbf{429500} \quad (9)$$

Finalmente, la interfaz radio, se ha configurado con los siguientes parámetros:

- Frecuencia de muestreo de la SDR: 30.72 MHz.
- Banda de frecuencia 5G: 1.
- ARFCN del enlace descendente: 429500.
- Ancho de banda del canal: 15 MHz.
- Número de antenas para el enlace ascendente: 2 antenas.
- Número de antenas para el enlace descendente: 2 antenas.
- Tabla MCS: qam64.

Con esta configuración, el terminal móvil nos arroja los siguientes resultados:



Figura 49: Test de velocidad realizado en el terminal móvil Xiaomi Redmi 12 5G.

Podemos comprobar, que los resultados son aceptables teniendo en cuenta la capacidad de un ordenador de laboratorio que está ejecutando simultáneamente el núcleo de red y la interfaz radio en contenedores Docker.

Además, analizando las estadísticas de la interfaz radio que podemos comprobar en el panel de control de Grafana, extraemos las siguientes conclusiones:

- El enlace radio ha utilizado la tecnología MIMO.
- El pico de velocidad de descarga ha sido de 95.6 Mbps.
- El pico de velocidad de subida ha sido de 12.3 Mbps.
- El valor de MCS en el enlace descendente ha sido en todo momento superior a 20, lo que significa que la modulación del enlace radio descendente ha sido 64QAM, mientras que para el enlace ascendente se ha utilizado la mayor parte del tiempo 4QAM ya que el valor de MCS para este enlace se ha mantenido en torno a 13. Esto se puede comprobar consultando la tabla 5.1.3.1-3 de la especificación técnica 38.214 de 3GPP.
- Podemos comprobar al principio de la subida de datos, que existía una alta pérdida de paquetes en el enlace ascendente, por lo que se redujo el número de símbolos de la modulación hasta que se minimizó el número de paquetes perdidos.
- Los problemas en el enlace ascendente podrían deberse a una baja sensibilidad por parte de la SDR en los canales de recepción. Para confirmarlo se puede probar con una antena con mayor ganancia.



Figura 50: Panel de Grafana donde se muestran las métricas del gNodeB relacionadas con el número de UE, uso de MIMO e índices de calidad de las conexiones descendente y ascendente

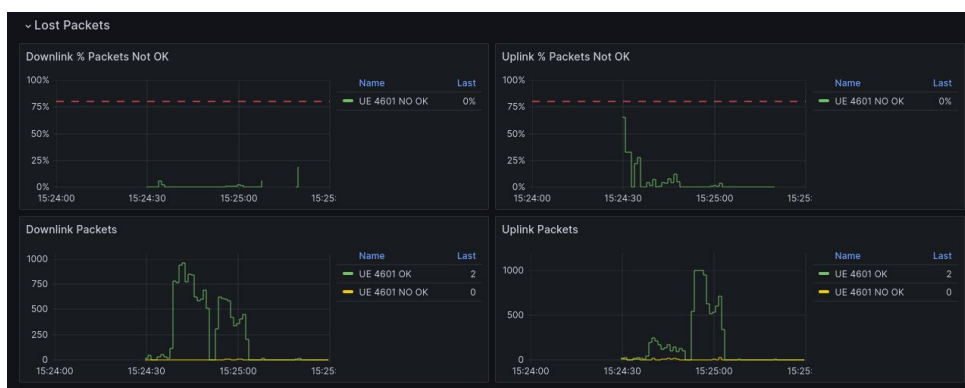


Figura 51: Panel de Grafana donde se muestran las métricas del gNodeB relacionadas con el número de paquetes enviados en los enlaces ascendente y descendente.

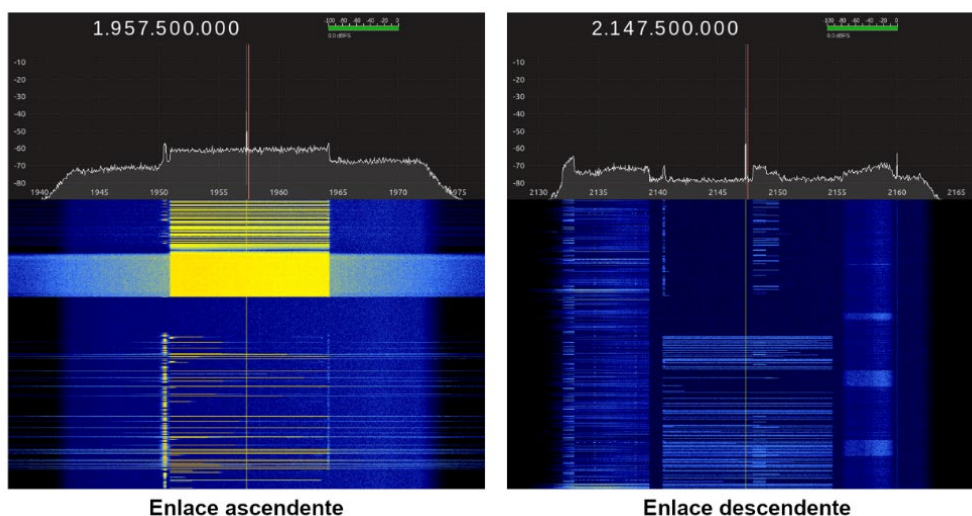


Figura 52: Espectro de los enlaces ascendente y descendente durante la prueba de velocidad.

5.2. Guiones de prácticas

A continuación se proponen dos guiones de prácticas relacionados con la interfaz radio y el núcleo de red del sistema 5G implementado.

Para el normal desarrollo de las prácticas propuestas, el sistema 5G desarrollado debe estar instalado y configurado en los equipos del laboratorio. Para ello se pueden seguir las instrucciones detalladas en el Anexo I: Puesta en marcha y configuración inicial del sistema 5G implementado.

Ambas prácticas están diseñadas para su desarrollo en un entorno Linux, aunque se recomienda que para las prácticas relacionadas con el núcleo de red, los estudiantes utilicen Windows para facilitar el uso de Wireshark.

Igualmente, en el caso de que la práctica relacionada con el núcleo de red utilice la configuración de gNB + UE virtualizados, se puede utilizar tanto Linux como Windows para desplegar el sistema 5G, lo que facilita también que los estudiantes puedan instalar el software desarrollado en su propio ordenador para la realización de prácticas en casa.

Se puede consultar de forma detallada la información acerca del sistema operativo que se puede utilizar en cada caso en el apartado “Requisitos previos” del Anexo I: Puesta en marcha y configuración inicial del sistema 5G implementado.

5.2.1. Práctica relacionada con la interfaz radio

El objetivo de esta práctica es proporcionar a los estudiantes una experiencia práctica en el despliegue de una red 5G SA en un entorno de laboratorio. Para ello, deberán realizar un estudio de las bandas de frecuencia compatibles con 5G y buscar espacios libres en el ancho de banda donde operará la estación base. Esta tarea implica no solo el conocimiento teórico de las especificaciones técnicas del 5G, sino también habilidades prácticas en la utilización de herramientas de análisis del espectro.

5.2.1.1. Selección de bandas de frecuencia.

El primer paso en este proceso es investigar las bandas de frecuencia que están designadas para uso en redes 5G. Las bandas de frecuencia se dividen en dos grandes categorías: FR1 (sub-6 GHz) y FR2 (ondas milimétricas). Cada una de estas bandas tiene características únicas en términos de cobertura y capacidad. El software instalado en los equipos del laboratorio solo permite realizar un despliegue bajo las bandas incluidas en la categoría FR1.

5.2.1.2. Análisis del espectro.

Una vez identificadas las bandas de frecuencia potenciales, los estudiantes deben realizar un análisis detallado del espectro para identificar espacios libres que puedan ser utilizados para desplegar la red 5G. Esto implica el uso de analizadores de espectro y la búsqueda online de información para detectar posibles interferencias y asegurar que el espectro seleccionado esté libre y sea adecuado para la operación de la estación base.

5.2.1.3. Configuración de la estación base.

Con los datos obtenidos del análisis del espectro, los estudiantes procederán a configurar la estación base. Deberán seleccionar los parámetros adecuados para minimizar posibles interferencias con otras redes o servicios.

Los parámetros configurables en la estación base son:

- **ARFCN del enlace descendente.** Para su cálculo se pueden utilizar calculadoras online.
- **Número de banda 5G.** El valor de ARFCN debe corresponder a una frecuencia ubicada en el interior de esta banda.
- **Ancho de banda.** Su valor depende de la banda utilizada.

5.2.1.4. Implementación y pruebas

Finalmente, los estudiantes implementarán su configuración en el entorno de laboratorio, utilizando el software disponible en los equipos del laboratorio. Realizarán una serie de pruebas para evaluar el rendimiento de la red en términos de cobertura, capacidad, latencia y calidad de servicio.

Para realizar el despliegue de la red se utilizará el software “docker_open5gs” disponible en los equipos del laboratorio. Este software se encuentra en la carpeta personal del usuario del equipo “/home/usuario/docker_open5gs”.

Para configurar la estación base se debe modificar con un editor de texto el archivo “gnb.yml” ubicado en la carpeta “srsran” (/home/usuario/docker_open5gs/srsran/gnb.yml).

Se deben configurar los siguientes parámetros:

- **Frecuencia de muestreo de la SDR.** Para ello, se puede cambiar el valor del parámetro “srate” dentro del bloque “ru_sdr”. El valor debe ser múltiplo de 1.92 MHz y mayor o igual al ancho de banda del canal.

- **Ganancia de la SDR.** Se puede modificar la ganancia de transmisión y recepción variando el valor de “tx_gain” entre 0 y 89.9 dB y “rx_gain” entre 0 y 76 dB. Estos parámetros se encuentran dentro del bloque “ru_sdr”.
- **ARFCN del enlace descendente.** Para ello se modifica el parámetro “dl_arfcn” ubicado en el bloque “cell_cfg”.
- **Banda.** Para establecer su valor se modifica el parámetro ‘band’ ubicado el bloque de opciones ‘cell_cfg’.
- **Ancho de banda.** Su valor puede establecerse modificando el parámetro “channel_bandwidth_MHz” ubicado en el bloque de opciones “cell_cfg”.
- **Número de antenas.** Se puede establecer modificando el valor de “nof_antennas_ul” y “nof_antennas_dl” ubicados en el bloque de opciones “cell_cfg”. Para utilizar la funcionalidad MIMO, su valor deberá ser “2” en ambos campos.
- **Parámetros de identificación de la celda.** Se puede establecer el valor de TAC y PCI retransmitidos modificando los parámetros “tac” y “pci” ubicados en el bloque de opciones “cell_cfg”.
- **Tabla de codificación.** Modificando el valor de “mcs_table” dentro de los bloques “pdsch” y “pusch” se puede establecer la tabla de codificación utilizada en los enlaces descendente y ascendente respectivamente. Su valor deberá ser “qam64”

Una vez configurados se guardará el archivo y se procederá al despliegue de la red 5G, para ello se debe conectar la SDR USRP B210 al equipo. Una vez conectada, se abrirá una terminal de comandos con poderes de administrador en la carpeta “docker_open5gs”.

Nota: para obtener permisos de administrador se puede escribir el comando:

```
sudo -s
```

Ahora se desplegará el núcleo de red con el siguiente comando:

```
docker compose -f sa-vonr-deploy.yaml up
```

Una vez desplegado el núcleo de red se abrirá otra ventana de terminal (se debe volver a escribir el comando para obtener permisos de administrador) y se introducirá el siguiente comando:

```
docker compose -f srsgnb.yaml up -d && docker container attach srsgnb
```

Con la red en funcionamiento podemos encender el teléfono móvil proporcionado. El terminal está equipado con una SIM compatible con el sistema 5G que se acaba de desplegar, por lo que no se debe realizar ninguna configuración extra para la conexión. Se puede acceder a una página web con estadísticas acerca de la estación radio accediendo a la dirección <http://127.0.0.1:3000>.

Para comprobar que se ha realizado correctamente el despliegue se propone navegar por internet y realizar un test de velocidad para evaluar las características de la red recién desplegada.

5.2.1.5. Cuestión teórica.

Con el software disponible en los equipos del laboratorio se ha realizado un despliegue 5G.

Tras realizar el estudio de las bandas de frecuencia libres en el laboratorio, se ha configurado la interfaz radio con los siguientes parámetros:

- **Frecuencia de muestreo de la SDR:** 30.72 MHz.
- **Banda de frecuencia 5G:** 1.
- **ARFCN del enlace descendente:** 429500.
- **Ancho de banda del canal:** 15 MHz.
- **Número de antenas para el enlace ascendente:** 2 antenas.
- **Número de antenas para el enlace descendente:** 2 antenas.
- **Tabla MCS:** qam64.

Con esta configuración, el terminal móvil nos arroja los siguientes resultados:



Figura 53: Práctica relacionada con el interfaz radio. Resultados prueba de velocidad en el terminal móvil.

Además, podemos extraer la siguiente información de la estación base en cuanto a la prueba de velocidad realizada.

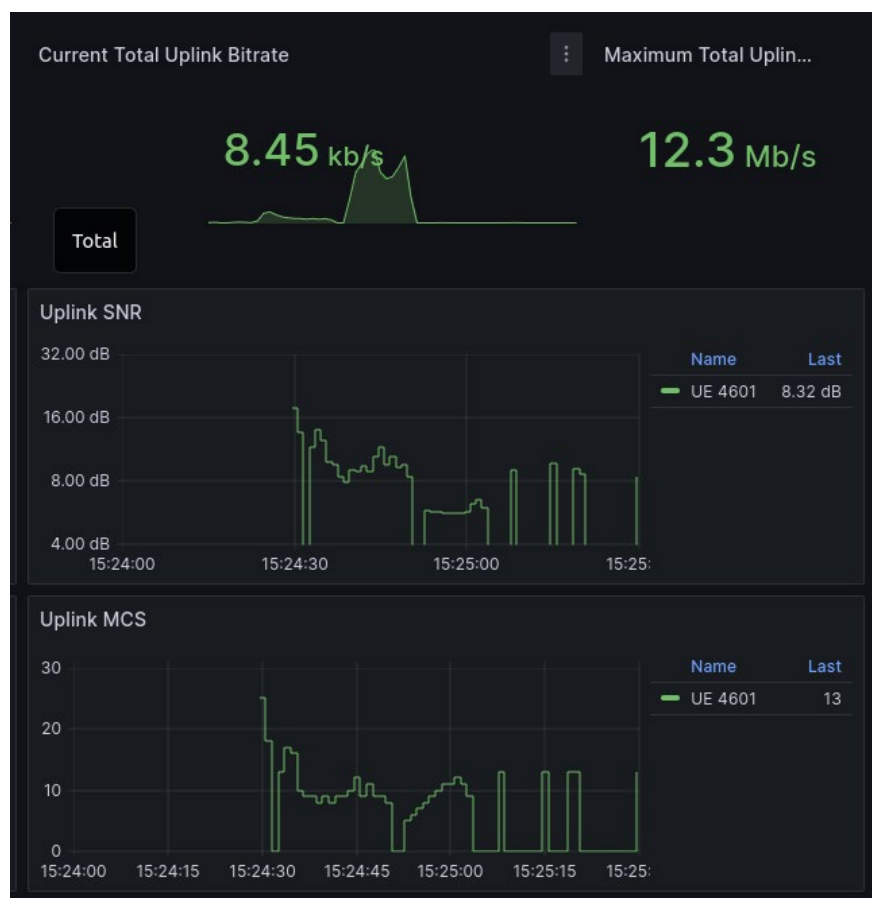


Figura 54: Práctica relacionada con el interfaz radio. Estadísticas del gNodeB relacionadas con el número de UE, uso de MIMO e índices de calidad de las conexiones descendente y ascendente

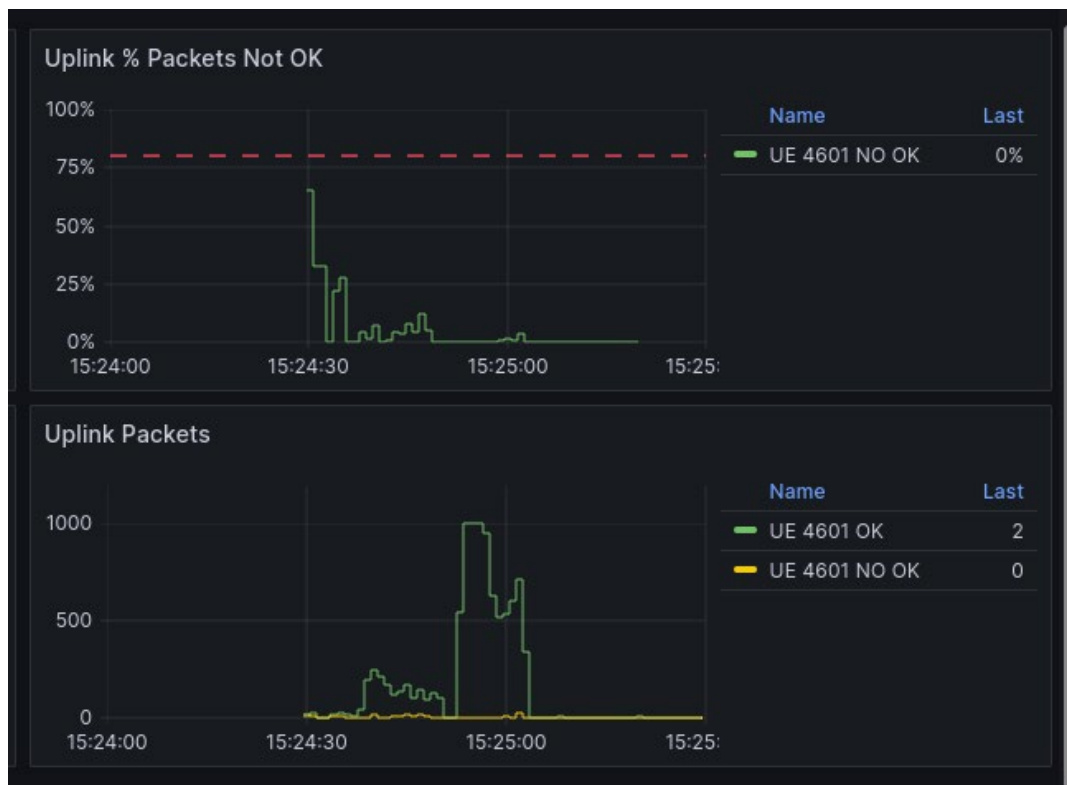


Figura 55: Práctica relacionada con el interfaz radio. Estadísticas del gNodeB relacionadas con el número de paquetes enviados en los enlaces ascendente y descendente.

En base a la información proporcionada en las figuras anteriores y la siguiente tabla, explique la relación existente entre el valor de paquetes erróneos, MCS y tasa de bits del enlace ascendente.

La siguiente tabla se utiliza única y exclusivamente cuando se configura la tabla “qam64” en los enlaces ascendente y descendente.

Índice MCS I_{MCS}	Orden de modulación Q_m	Tasa de código objetivo x [1024] R	Eficiencia espectral
0	2	30	0.0586
1	2	40	0.0781
2	2	50	0.0977
3	2	64	0.1250
4	2	78	0.1523
5	2	99	0.1934
6	2	120	0.2344
7	2	157	0.3066
8	2	193	0.3770
9	2	251	0.4902
10	2	308	0.6016
11	2	379	0.7402
12	2	449	0.8770
13	2	526	1.0273
14	2	602	1.1758
15	4	340	1.3281
16	4	378	1.4766
17	4	434	1.6953
18	4	490	1.9141
19	4	553	2.1602
20	4	616	2.4063
21	6	438	2.5664
22	6	466	2.7305
23	6	517	3.0293
24	6	567	3.3223
25	6	616	3.6094
26	6	666	3.9023
27	6	719	4.2129
28	6	772	4.5234
29	2	Reservado	
30	4	Reservado	
31	6	Reservado	

Tabla 10: Tabla 5.1.3.1-3 de la especificación técnica 38.214 del 3GPP.

5.2.2. Práctica relacionada con el núcleo de red

En el ámbito de las comunicaciones móviles, el proceso de registro de usuarios es un paso esencial que permite a los dispositivos acceder a la red y disfrutar de sus servicios. Este proceso conlleva un intercambio de mensajes entre el terminal móvil y distintos componentes del núcleo de red. El laboratorio cuenta con un software que permite desplegar una red 5G SA con servicios de internet y SMS.

El objetivo de esta práctica es que los alumnos identifiquen los equipos involucrados en el registro de un terminal móvil a la red y conozcan los mensajes intercambiados entre ellos para lograr este fin.

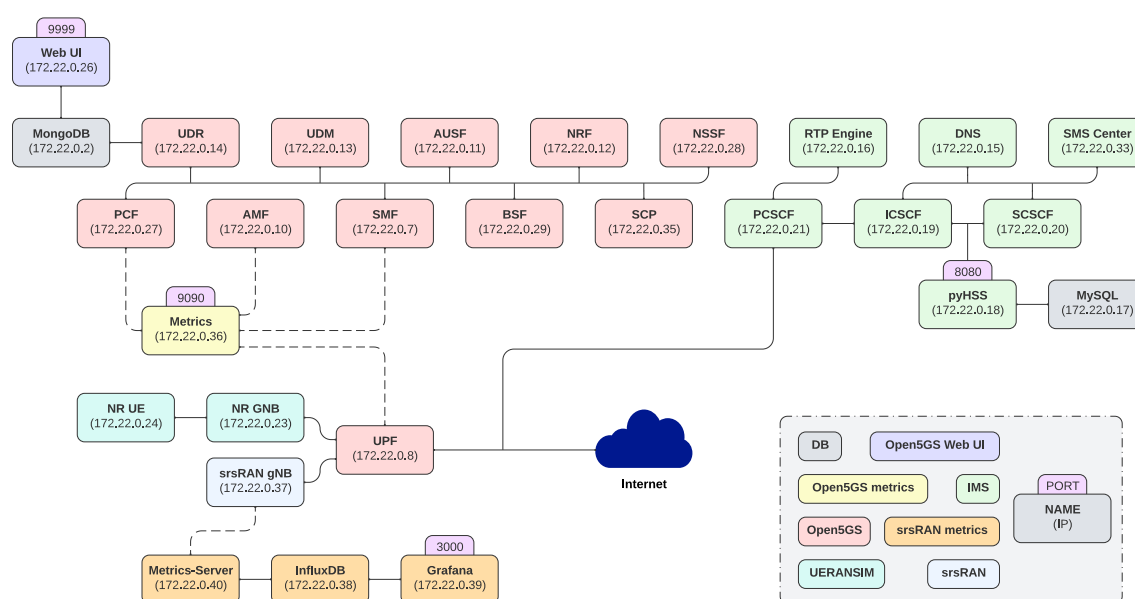


Figura 56: Práctica relacionada con el núcleo de red. Elementos que componen el sistema 5G desplegado por el software "docker_open5gs".

1. En primer lugar, determine el significado de las siglas SCP, GNB, IMSI, NGAP y AMF. Explique la función que cumple cada uno en el funcionamiento de una red 5G.

Ahora, el profesor desplegará el sistema 5G en un ordenador. El estudiante debe estar conectado a la misma red que el equipo del profesor para poder continuar.

2. Acceda al panel principal de Ghostwire. Se encuentra disponible en la dirección http://<5G_IP>:5001, siendo <5G_IP> la dirección IP del equipo que está ejecutando el núcleo de red del sistema 5G.
3. Localice el contenedor AMF y pulse en el icono con forma de aleta de tiburón para abrir Wireshark y comenzar con la captura.

Ahora el profesor conectará un terminal móvil a la red utilizando para ello una SDR que simulará una estación base. En el caso de no disponer de acceso a la SDR y los terminales móviles, se puede utilizar el simulador de estos elementos incluido en el software disponible en el laboratorio.

4. Filtre y localice los mensajes NGAP en la captura.
5. Realice una gráfica de flujo e identifique los equipos que realizan el intercambio de estos mensajes NGAP.

Cada UE cuenta con un identificador único, SUPI (Identidad Permanente de Suscripción), utilizado para diferenciarse de otros UE de la red. Durante el registro del UE en la red se intercambia este identificador, que puede ser, junto al IMSI, uno de los siguientes:

- SUCI (Identificador Oculto de Suscripción). Se trata de un identificador temporal para ocultar el SUPI. Ofrece características mejoradas de seguridad y privacidad en comparación con el IMSI.
 - 5G-GUTI (Identidad Temporal Global Única de 5G): Es un identificador temporal utilizado en redes 5G para identificar un dispositivo móvil y su suscripción asociada. Se usa en lugar del IMSI para proteger la identidad del dispositivo y proporcionar una mayor seguridad y privacidad.
 - IMEI (Identidad Internacional del Equipo Móvil). Se trata de un identificador único asignado a un dispositivo móvil por el fabricante. Es utilizado para identificar al dispositivo y se emplea para propósitos como bloquear el acceso a la red a dispositivos robados.
 - TMSI (Identidad Temporal del Suscriptor Móvil). Es un identificador temporal asignado a un dispositivo móvil por la red para proteger la identidad del dispositivo durante su operación normal. Se usa en lugar del IMSI para reducir la sobrecarga de señalización y mejorar la seguridad.
6. Localice el tipo de identificador utilizado durante el registro del UE en la red y su valor.
 7. Elimine los filtros y busque en la captura "IMSI". Para ello pulse CTRL+F, seleccione "Detalles de paquete" y "Cadena" y deshabilite "Mayúsculas y minúsculas" en las opciones de búsqueda.
 8. Identifique el equipo con el que se están intercambiando los paquetes que contienen el valor del IMSI.
 9. Especifique cuál es el IMEI y el número de teléfono del UE que se ha registrado en la red.

6. CONCLUSIONES

El desarrollo de este Trabajo Fin de Grado ha permitido el diseño y la implementación de prácticas de laboratorio orientadas al estudio del núcleo de red y la interfaz radio de un sistema 5G, partiendo de una infraestructura basada en contenedores Docker.

A través de las prácticas diseñadas, los estudiantes puede ahora explorar de manera interactiva los aspectos fundamentales de la tecnología 5G. La práctica relacionada con la interfaz radio les permite comprender la configuración y optimización de parámetros críticos, así como la gestión de recursos y análisis de rendimiento de la RAN. Por otro lado, la práctica centrada en el núcleo de red facilita la comprensión de los procesos de registro de usuarios y la captura y análisis de mensajes de señalización, proporcionando una visión detallada del funcionamiento interno de la red 5G. Además, gracias a las herramientas desarrolladas, se pueden realizar prácticas relacionadas con cualquier aspecto del sistema 5G.

El uso de contenedores Docker ha demostrado ser una herramienta eficaz y flexible para la simulación y despliegue de componentes de red, permitiendo la replicación de un entorno de red realista y escalable. Esto no solo facilita la implementación de las prácticas, sino que también ofrece a los estudiantes una experiencia valiosa en la configuración y gestión de entornos virtualizados, una habilidad cada vez más demandada en la industria de las telecomunicaciones.

En conclusión, este trabajo contribuye de manera significativa al fortalecimiento de la formación práctica en el área de telecomunicaciones de la Universidad de Jaén, proporcionando una herramienta educativa de gran valor para la preparación de futuros profesionales en tecnologías de redes móviles de última generación.

No obstante, se identifican algunas áreas para futuras mejoras como son:

- La posibilidad de realizar llamadas de voz utilizando la tecnología VoNR.
- Añadir la opción de exportar las estadísticas recogidas por la interfaz radio para el desarrollo de prácticas de laboratorio en las que no sea necesario desplegar el sistema 5G al completo.
- El diseño de antenas para su uso en bandas de frecuencia específicas compatibles con 5G.
- Añadir al panel de estadísticas de la interfaz radio varias gráficas relacionadas con la eficiencia espectral y el orden de modulación utilizado en cada momento.

7. ANEXOS

7.1. Anexo I: Puesta en marcha y configuración inicial del sistema 5G implementado.

7.1.1. Requisitos previos

Para implementar el sistema 5G completo es necesario contar con una máquina con un sistema operativo Linux compatible con Docker.

Plataforma	x86_64/amd64	arm64/aarch64	arm (32-bit)	ppc64le	s390x
CentOS	✓	✓		✓	
Debian	✓	✓	✓	✓	
Fedora	✓	✓		✓	
Raspberry Pi OS (32-bit)			✓		
RHEL	⚠	⚠			✓
SLES					✓
Ubuntu	✓	✓	✓	✓	✓
Binarios	✓	✓	✓		

⚠ = Experimental

Tabla 11: Plataformas compatibles con Docker (42).

Igualmente, sería posible implementar el sistema en una máquina Windows con Docker instalado a excepción del componente srsRAN, ya que no es posible la comunicación USB entre la SDR y la máquina virtual en Docker.

Además, tampoco es posible implementar el núcleo de red en un equipo Windows y la red de acceso en uno Linux por culpa del firewall de Windows. Aun desactivándolo, no es posible habilitar el puerto SCTP utilizado para la comunicación entre el núcleo de red y la RAN.

La compatibilidad de los componentes que forman el sistema de 5G implementado se resume en la siguiente tabla.

Componente	Ubuntu 22.04	Windows 11
Open5GS	✓	✓
srsRAN	✓	
UERANSIM	✓	✓

Tabla 12: Compatibilidad de componentes con los distintos sistemas operativos.

7.1.2. Instalación de Docker

7.1.2.1. Ubuntu

Para instalar Docker Engine, es necesario contar con la versión de 64 bits de alguna de estas versiones de Ubuntu:

- Ubuntu Noble 24.04 (LTS).
- Ubuntu Jammy 22.04 (LTS).
- Ubuntu Focal 20.04 (LTS).

Antes de instalar Docker, debemos de desinstalar las versiones antiguas que tengamos instaladas, ya que puede causar conflictos con la nueva instalación.

Para eliminar todos los paquetes que pueden causar un conflicto ejecute el siguiente comando:

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do sudo apt-get remove $pkg; done
```

Ahora debemos de instalar el repositorio desde el que descargaremos la última versión de Docker. Para ello ejecutamos los siguientes comandos:

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
# Add the repository to Apt sources:
echo \
    "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu
\
    $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
    sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Por último, para instalar la última versión de Docker ejecutamos:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-
plugin docker-compose-plugin
```

7.1.2.2. Windows

Para instalar Docker Desktop en Windows es necesario descargar el instalador desde la página oficial de Docker.

Una vez descargado, lo ejecutamos y seguimos las instrucciones del instalador:

En la primera pantalla que nos aparece, hay que asegurarse de seleccionar la opción **“Usar WSL 2 en lugar de Hyper-V”**.

Una vez instalado, para ejecutar Docker simplemente debemos buscar Docker Desktop en la barra de búsqueda del menú de inicio de Windows.

La primera vez que lo ejecutemos deberemos de aceptar los términos de uso. A continuación nos pedirá iniciar sesión, aunque podemos continuar sin hacerlo.

Después deberemos de rellenar una encuesta, aunque podemos saltarla pulsando en la opción para ello situada debajo.

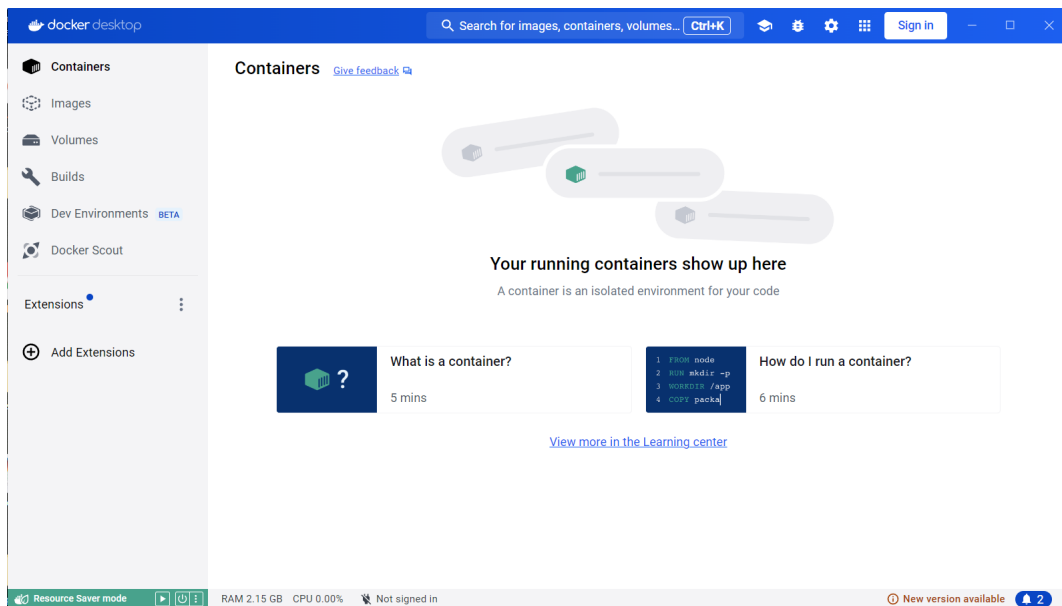


Figura 57: Interfaz de usuario de Docker Desktop.

Tras desplegar el sistema 5G, observaremos en la interfaz de Docker Desktop todos los contenedores que forman el sistema y además podremos realizar multitud de operaciones con cada uno de ellos. Entre estas operaciones se encuentran la consulta de *logs* o visualización de estadísticas de uso.

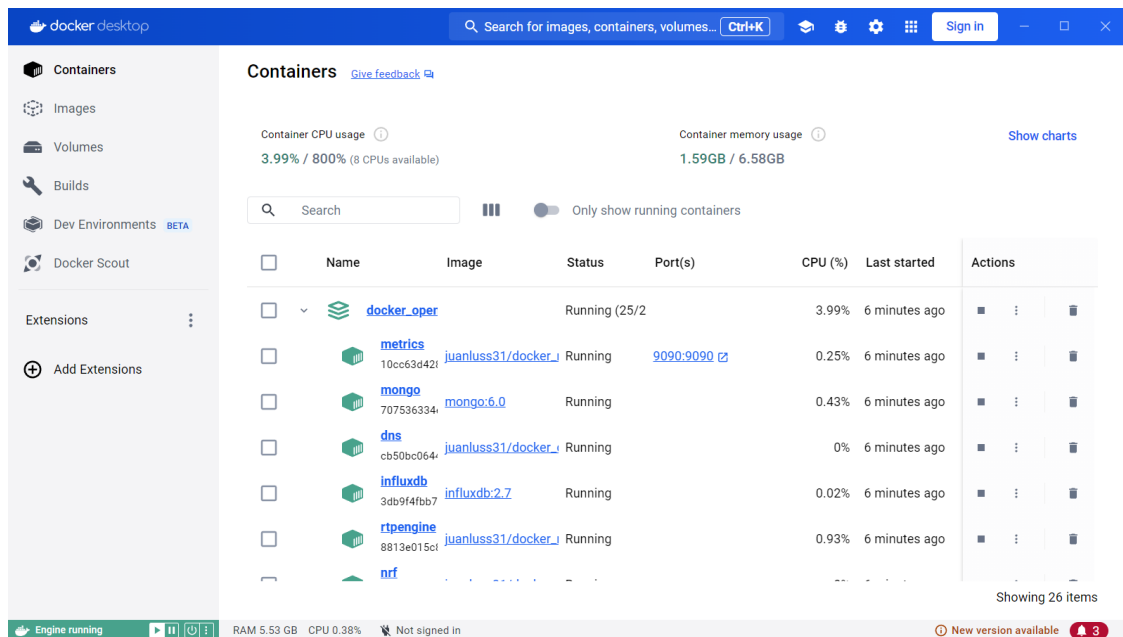


Figura 58: Detalle de los contenedores ejecutándose en Docker visto desde la aplicación Docker Desktop en Windows.

7.1.3. Descarga del repositorio docker_open5gs

Para descargar el repositorio podemos clonarlo utilizando la línea de comandos:

```
git clone https://github.com/juanluss31/docker_open5gs
```

En el caso de Windows, es preferible descargar el repositorio como archivo zip desde la web https://github.com/juanluss31/docker_open5gs para evitar problemas con los saltos de línea en la ejecución.

Windows y Linux utilizan saltos de línea distintos en los documentos de texto, CRLF y LF respectivamente. A la hora de ejecutar el sistema 5G, los distintos componentes leen su configuración desde varios ficheros de texto. Al encontrarse todos los componentes virtualizados en componentes basados en imágenes Linux, si los ficheros de configuración utiliza saltos de línea de tipo CRLF (Windows), no son capaces de interpretar correctamente la información, ya que esperan ficheros con saltos de línea del tipo LF (Linux).

7.1.4. Configuración básica del sistema 5G SA con IMS

El despliegue se puede realizar principalmente de dos formas distintas según el escenario:

- Configuración con un único equipo que incluya núcleo (5GC + IMS) y gNB.
- Configuración con varios equipos donde el núcleo (5GC + IMS) se despliegue en un equipo distinto al gNB.

En ambos casos, se deberán indicar los servidores DNS de la red a la que está conectado el 5GC para que los UE tengan conexión a Internet. Para ello, basta con establecer los servidores DNS en el archivo “named.conf”, localizado en el interior de la carpeta “dns”, y en el archivo “.env” modificando los valores “SMF_DNS1” y “SMF_DNS2”.

7.1.4.1. Configuración con un único equipo.

En este caso únicamente hay que modificar los siguientes parámetros en el archivo “.env”:

```
NETWORKNAME
MCC
MNC
DOCKER_HOST_IP --> Esta es la dirección IP del equipo que ejecuta Docker
UE_IPV4_INTERNET --> Cambiar al rango IP deseado para la red con el APN
internet
UE_IPV4_IMS --> Cambiar al rango IP deseado para la red con el APN ims
```

7.1.4.2. Configuración con varios equipos.

En este caso debemos de configurar cada equipo por separado.

- Equipo con el núcleo de red:

```
NETWORKNAME
MCC
MNC
DOCKER_HOST_IP --> Esta es la dirección IP del equipo que ejecuta el 5GC
UPF_ADVERTISE_IP --> Cambiar por el valor que haya en DOCKER_HOST_IP
UE_IPV4_INTERNET --> Cambiar al rango IP deseado para la red con el APN
internet
UE_IPV4_IMS --> Cambiar al rango IP deseado para la red con el APN ims
```

- Equipo con el gNB:

```
MCC
MNC
DOCKER_HOST_IP --> Esta es la dirección IP del equipo que ejecuta el gNB
AMF_IP --> Cambiar por la dirección IP del equipo que ejecuta el 5GC
SRS_GNB_IP --> Cambiar por la dirección IP del equipo que ejecuta el gNB
METRICS_SERVER_IP --> Cambiar por la dirección IP del equipo que ejecuta
el 5GC
```

7.1.4.3. Configuración de la interfaz radio con srsRAN

En el caso de la interfaz radio, podemos configurar cada gNB modificando el archivo “gnb.yml” ubicado dentro de la carpeta “srsran”.

En este archivo podemos modificar los siguientes parámetros:

- **Frecuencia de muestreo de la SDR.** Para ello, se puede cambiar el valor del parámetro “srate” dentro del bloque “ru_sdr”. El valor debe ser múltiplo de 1.92 MHz y mayor o igual al ancho de banda del canal.
- **Ganancia de la SDR.** Se puede modificar la ganancia de transmisión y recepción variando el valor de “tx_gain” entre 0 y 89.9 dB y “rx_gain” entre 0 y 76 dB. Estos parámetros se encuentran dentro del bloque “ru_sdr”.
- **ARFCN del enlace descendente.** Para ello se modifica el parámetro “dl_arfcn” ubicado en el bloque “cell_cfg”.

- **Banda.** Se puede escoger la banda de NR utilizada. El valor de ARFCN debe corresponder a una frecuencia ubicada en el interior de esta banda. Para establecer su valor se modifica el parámetro “band” ubicado el bloque de opciones “cell_cfg”.
- **Ancho de banda.** Su valor depende de la banda utilizada. Puede establecerse modificando el parámetro “channel_bandwidth_MHz” ubicado en el bloque de opciones “cell_cfg”.
- **Número de antenas.** Se puede establecer el valor de canales utilizados en los enlaces ascendente y descendente modificando el valor de “nof_antennas_ul” y “nof_antennas_dl” ubicados en el bloque de opciones “cell_cfg”.
- **Parámetros de identificación de la celda.** Se puede establecer el valor de TAC y PCI retransmitidos modificando los parámetros “tac” y “pci” ubicados en el bloque de opciones “cell_cfg”.
- **Tabla de codificación.** Modificando el valor de “mcs_table” dentro de los bloques “pdsch” y “pusch” se puede establecer la tabla de codificación utilizada en los enlaces descendente y ascendente respectivamente.

Cada vez que se cambie algún valor en el documento, este no se hará efectivo hasta que se vuelva a lanzar la interfaz radio mediante el comando:

```
docker compose -f srsgnb.yaml up -d && docker container attach srsgnb
```

7.1.5. Despliegue del sistema 5G SA con IMS

Para poner en funcionamiento el sistema es necesario ejecutar los siguientes comandos desde dentro de la carpeta “docker_open5gs”:

- Configuración con un equipo

```
# 5G Core Network
docker compose -f sa-vonr-deploy.yaml up

# srsRAN gNB using SDR (OTA)
docker compose -f srsgnb.yaml up -d && docker container attach srsgnb
docker compose -f srsgnb.yaml up -d ; if($?) {docker container attach
srsgnb}    # Windows Host
```

```
# UERANSIM nr-gNB (RF simulated)
docker compose -f nr-gnb.yaml up -d && docker container attach nr_gnb
docker compose -f nr-gnb.yaml up -d ; if($?) {docker container attach
nr_gnb}      # Windows Host

# UERANSIM NR-UE (RF simulated)
docker compose -f nr-ue.yaml up -d && docker container attach nr_ue
docker compose -f nr-ue.yaml up -d ; if($?) {docker container attach
nr_ue}      # Windows Host
```

- Configuración con varios equipos

```
# 5G Core Network
docker compose -f sa-vonr-deploy-multihost.yaml up

# srsRAN gNB using SDR (OTA)
docker compose -f srsgnb_multihost.yaml up -d && docker container attach
srsgnb
docker compose -f srsgnb_multihost.yaml up -d ; if($?) {docker container
attach srsgnb}      # Windows Host
```

7.1.6. Aprovisionamiento de abonados al 5GC e IMS

Una vez el sistema 5G ha sido desplegado aún no es capaz de dar servicio a ningún abonado ya que no hay ninguno en su base de datos.

El sistema 5G implementado cuenta con dos bases de datos de usuario. Una base de datos MongoDB que contiene los datos de subscripción de todos los abonados a la red, y una base de datos SQL que contiene los datos necesarios para la subscripción de los abonados al IMS.

Para añadir abonados a estas bases de datos podemos utilizar un script preparado para cargar en el sistema la información necesaria para que las diez tarjetas SIM del laboratorio tengan servicio, o podemos añadir la información a ambas bases de datos manualmente.

7.1.6.1. Aprovisionamiento automático de abonados

Para facilitar el proceso de aprovisionamiento de abonados al sistema 5G se ha creado una serie de scripts que permiten añadirlos automáticamente desde la línea de comandos.

Para evitar problemas con la instalación de dependencias para que los scripts funcionen correctamente, se ha creado una imagen Docker con todo lo necesario.

Ejecutando el comando que aparece a continuación, se crea un contenedor que se une a la red virtual donde se encuentran todos los equipos del sistema 5G y ejecuta los scripts diseñados para cargar los usuarios a la base de datos.

```
# 5G Core Network
docker compose -f sa-vonr-deploy-config.yaml up      # Run only once
```

Finalmente contaremos con los siguientes abonados en el sistema:

IMSI	MSISDN
001010000053530	0001110
001010000053531	0001111
001010000053532	0001112
001010000053533	0001113
001010000053534	0001114
001010000053535	0001115
001010000053536	0001116
001010000053537	0001117
001010000053538	0001118
001010000053539	0001119

Tabla 13: Abonados al sistema 5G

7.1.6.2. Aprovisionamiento manual de abonados

Para añadir abonados a las distintas bases de datos del sistema 5G podemos hacerlo a través del navegador web.

- Núcleo de red (Open5GS)

Abrimos en un navegador la dirección `http://<DOCKER_HOST_IP>:9999`, siendo `<DOCKER_HOST_IP>` la dirección IP del equipo que está ejecutando el núcleo de red del sistema 5G. Accedemos con las siguientes credenciales:

Username : admin

Password : 1423

Utilizando la interfaz web podemos añadir fácilmente un nuevo abonado.

La información que debemos introducir por cada abonado es:

- IMSI
- MSISDN
- Ki
- OPc

Create Subscriber

Subscriber Configuration

IMSI*

* is required

+

Subscriber Key (K)*

465B5CE8 B199B49F AA5F0A2E E238A6BC

Authentication Management Field (AMF)*

8000

USIM Type

OPc

Operator Key (OPc/OP)*

E8ED289D EBA952E4 283B54E8 8E6183CA

UE-AMBR Downlink*

1

Unit

Gbps

UE-AMBR Uplink*

1

Unit

Gbps

CANCEL SAVE

Figura 59: Interfaz para añadir abonados a la base de datos de Open5GS

Además, debemos configurar las APN a las que tiene acceso el abonado.

Debemos configurarlas con la siguiente información:

APN	Type	QCI	ARP	Capability	Vulnerability	MBR DL/UL (Kbps)	GBR DL/UL (Kbps)
internet	IPv4	9	8	Disabled	Disabled	unlimited/unlimited	
ims	IPv4	5	1	Disabled	Disabled	3850/1530	
		1	2	Enabled	Enabled	128/128	128/128
		2	4	Enabled	Enabled	128/128	128/128

Tabla 14: Configuración APN de cada abonado en la base de datos de Open5GS

001010000053530

Subscriber Configuration

0001110 --- MSISDN
1B7F04D495107F31E8DF29E5514F28CA
---K
DD71BC1880AF9367E4A52ECABAC326FD
---OPc
8000 --- AMF

SERVICE_GRANTED (0) ---
Subscriber Status (TS 29.272
7.3.29)
0 --- Operator Determined Barring
(TS 29.272 7.3.30)

1 Gbps --- DL
1 Gbps --- UL

SST:1 (Default S-NSSAI)

DNN/APN	Type	5QI/QCI	ARP	Capability	Vulnerability	MBR DL/UL	GBR DL/UL
internet	IPv4	9	8	Disabled	Disabled	1 Gbps / 1 Gbps	
ims	IPv4	5	1	Disabled	Disabled	3850 Kbps / 1530 Kbps	
		1	2	Enabled	Enabled	128 Kbps / 128 Kbps	128 Kbps / 128 Kbps
		2	4	Enabled	Enabled	128 Kbps / 128 Kbps	128 Kbps / 128 Kbps

Figura 60: Datos de un abonado a la base de datos de Open5GS

- Subsistema multimedia IMS (pyHSS)

En el caso de la base de datos del subsistema multimedia, la forma de añadir abonados es algo más complicada, por ello se han numerado los pasos a seguir. Nota: **el paso número 2 solo hay que realizarlo una vez**, ya que consiste en la creación de las APN.

1. En primer lugar abrimos en un navegador la dirección `http://<DOCKER_HOST_IP>:8080/docs/`.
2. Seleccionamos **apn** → **Create new APN** → Pulsamos en **Try it out**. Después, en la sección del *payload* escribimos el siguiente JSON y pulsamos **Execute**.

```
{
  "apn": "internet",
  "apn_ambr_dl": 0,
  "apn_ambr_ul": 0
}
```

Tomamos nota del **apn_id** especificado en el cuerpo de mensaje de respuesta (**Response body**) bajo la respuesta del servidor (**Server response**). Este será el identificador del APN **internet**.

Repetimos este paso con el siguiente *payload*:

```
{
  "apn": "ims",
  "apn_ambr_dl": 0,
  "apn_ambr_ul": 0
}
```

Tomamos nota del identificador, **apn_id**, del APN **ims**.

3. Ahora, seleccionamos **auc** → **Create new AUC** → Pulsamos en **Try it out**. Después, en la sección del *payload* escribimos el siguiente JSON de ejemplo con el **IMSI** y las claves **Ki** y **OPc** de nuestra tarjeta SIM y después pulsamos **Execute**.

```
{
  "ki": "8baf473f2f8fd09487cccdbd7097c6862",
  "opc": "8E27B6AF0E692E750F32667A3B14605D",
  "amf": "8000",
  "sqn": 0,
  "imsi": "001010123456790"
}
```

Tomamos nota del **auc_id** especificado en el cuerpo del mensaje de respuesta.

4. Después, seleccionamos **subscriber** → **Create new SUBSCRIBER** → Pulsamos en **Try it out**. Entonces, en la sección del *payload* escribimos el siguiente JSON de ejemplo con los datos de nuestra tarjeta SIM y después pulsamos **Execute**.

```
{
  "imsi": "001010123456790",
  "enabled": true,
  "auc_id": 1,
  "default_apn": 1,
  "apn_list": "1,2",
  "msisdn": "9076543210",
  "ue_ambr_dl": 0,
  "ue_ambr_ul": 0
}
```

- **auc_id** es el ID del **AUC** creado en el paso 3 para esa SIM.
 - **default_apn** es el ID del APN **internet** creado en el paso 2.
 - **apn_list** es una lista separada por comas de los IDs de las APN que tiene acceso el abonado las apn **internet** e **ims** creadas anteriormente.
5. Finalmente, seleccionamos **ims_subscriber** → **Create new IMS SUBSCRIBER** → Pulsamos en **Try it out**. Entonces, en la sección del **payload** escribimos el siguiente JSON de ejemplo con el los datos de nuestra tarjeta SIM y después pulsamos **Execute**.

```
{
  "imsi": "001010123456790",
  "msisdn": "9076543210",
  "sh_profile": "string",
  "scscf_peer": "scscf.ims.mnc001.mcc001.3gppnetwork.org",
  "msisdn_list": "[9076543210]",
  "ifc_path": "default_ifc.xml",
  "scscf": "sip:scscf.ims.mnc001.mcc001.3gppnetwork.org:6060",
  "scscf_realm": "ims.mnc001.mcc001.3gppnetwork.org"
}
```

7.1.7. Programación de las tarjetas SIM

La programación de las tarjetas SIM debe realizarse desde un equipo Linux.

En primer lugar, debemos instalar las dependencias necesarias:

```
sudo apt-get install pcscd pcsc-tools libccid libpcsclite-dev python3-pyusb
```

A continuación, debemos conectar el lector de tarjetas al equipo e insertar una tarjeta SIM programable en el lector.

Podemos comprobar el estado de la conexión ejecutando el comando **pcsc_scan**:

```
$ pcsc_scan
PC/SC device scanner
V 1.5.2 (c) 2001-2017, Ludovic Rousseau ludovic.rousseau@free.fr
Using reader plug'n play mechanism
```

```
Scanning present readers...
0: HID Global OMNIKEY 3x21 Smart Card Reader [OMNIKEY 3x21 Smart Card
Reader] 00

Sun May 26 14:26:12 2019
Reader 0: HID Global OMNIKEY 3x21 Smart Card Reader [OMNIKEY 3x21 Smart
Card Re
Card state: Card inserted,
ATR: 3B 9F 96 80 1F C7 80 31 A0 73 BE 21 13 67 43 20 07 18 00 00 01 A5
...
```

Si todo funciona correctamente podremos leer “Card inserted”.

Ahora, descargamos el repositorio PySIM:

```
git clone https://gitea.osmocom.org/sim-card/pysim.git
```

Instalamos las dependencias necesarias para el correcto funcionamiento de PySIM:

```
sudo apt-get install --no-install-recommends \
    pcscd libpcsc-lite-dev \
    python3 \
    python3-setuptools \
    python3-pycryptodome \
    python3-pyserial \
    python3-pip
pip3 install --user -r requirements.txt
```

Ya podemos leer la información de nuestra tarjeta SIM:

```
./pySim-read.py -p0
```

Por último, programamos cada tarjeta SIM utilizando para ello el script “sim-program” ubicado en la carpeta “sim”:

```
$ ./sim-program
pySIM and docker_open5gs must be in the same directory
Use: ./sim-program [options]
Options:
```

card [IMSI last number (0-9)]	
coims install [IMSI last number (0-9)]	Install certificates to use CoIMS app
coims list [IMSI last number (0-9)]	List of installed certificates on the SIM
scan	Execute pcsc_scan
help	Shows pySim help

El script en cuestión contiene la información necesaria para programar las diez tarjetas sysmolSIM-SJA2 disponibles en el Departamento de Ingeniería de Telecomunicación de la Universidad de Jaén.

En el caso de que se necesite programar una tarjeta SIM distinta a las facilitadas por el departamento sería necesario ejecutar el siguiente comando con la información oportuna:

```
./pySim-prog.py -p0 \
-a $PIN_ADM \
-n $OPERATOR \
-c $COUNTRY_CODE \
-x $MCC \
-y $MNC \
-m $SMSC \
-s $ICCID \
-i ${MCC}${MNC}${ID} \
--msisdn=$MSISDN \
-k $KI \
-o $OPC \
--acc=$ACC
```

Por ejemplo:

```
./pySim-prog.py -p0 \
-a 85903454\
-n UJA5G \
-c 0 \
-x 001 \
-y 01 \
-m 0005555 \
```

```
-s 8988211000000535305 \  
-i 001010000053530 \  
--msisdn=0001110 \  
-k 1B7F04D495107F31E8DF29E5514F28CA \  
-o DD71BC1880AF9367E4A52ECABAC326FD \  
--acc=0001
```

Nota: el IMSI 999700000053530 está formado por:

MCC	999
MNC	70
MSIN	0000053530

Tabla 15: Detalle del número IMSI

Por lo que el IMSI tiene que ser modificado según el MCC y el MNC utilizados en la implementación.

Por ejemplo, en este trabajo se ha utilizado el MCC: 001 y el MNC: 01.

Por tanto, el IMSI anterior debería ser: 001010000053530.

7.2. Anexo II: Archivos de configuración del sistema 5G

7.2.1. .env

El principal archivo de configuración del sistema 5G implementado es el archivo .env. Éste cuenta con todos los valores de entorno que utilizan todos los archivos que forman el sistema. En él podemos modificar la dirección IP de todos los equipos, así como el nombre de la red o el código de país (MCC) y operador (MNC) del sistema.

Este fichero cuenta también con la información de la tarjeta SIM ficticia del simulador UERANSIM.

Por otro lado, el fichero también contiene la configuración básica para el correcto funcionamiento de los servicios de métricas de la srsRAN, utilizados para monitorizar la señal y otros parámetros de la interfaz radio.

A continuación, se encuentra el fichero en su totalidad.

```
1 # Set proper timezone to sync times between docker host and containers
2 #TZ=Europe/Berlin
3
4 NETWORKNAME=EPSL5G
5
6 MCC=001
7 MNC=01
8
9 TEST_NETWORK=172.22.0.0/24
10 DOCKER_HOST_IP=192.168.10.1 # Multihost Deployment: IP adress of the host running 5GC or
   gNB (this PC) (change always)
11
12 # AMF
13 AMF_IP=172.22.0.10 # Multihost Deployment: IP adress of the host running 5GC or
   172.22.0.10 (change in gNB's PC)
14
15 # SRSRAN GNB
16 SRS_GNB_IP=172.22.0.37 # Multihost Deployment: IP adress of the host running gNB or
   172.22.0.37 (change in gNB's PC)
17
18 # srsRAN Metrics Server
19 METRICS_SERVER_PORT=55555
20 METRICS_SERVER_IP=172.22.0.40 # Multihost Deployment: IP adress of the host running 5GC or
   172.22.0.40 (change in gNB's PC)
21
22 # UPF
23 UPF_IP=172.22.0.8
24 UPF_ADVERTISE_IP=172.22.0.8 # Multihost Deployment: IP adress of DOCKER_HOST_IP or
   172.22.0.8 (change in 5GC's PC)
25
26 # MONGODB
27 MONGO_IP=172.22.0.2
28
29 # HSS - open5gs
30 HSS_IP=172.22.0.3
31
32 # PCRF
33 PCRF_IP=172.22.0.4
34 PCRF_BIND_PORT=3873
35
```

```

36 # SGW
37 SGWC_IP=172.22.0.5
38 SGWU_IP=172.22.0.6
39 SGWU_ADVERTISE_IP=172.22.0.6
40
41 # SMF
42 SMF_IP=172.22.0.7
43 SMF_DNS1=150.214.170.15
44 SMF_DNS2=150.214.170.21
45
46 # MME
47 MME_IP=172.22.0.9
48
49 # AUSF
50 AUSF_IP=172.22.0.11
51
52 # NRF
53 NRF_IP=172.22.0.12
54
55 # UDM
56 UDM_IP=172.22.0.13
57
58 # UDR
59 UDR_IP=172.22.0.14
60
61 # IMS DNS
62 DNS_IP=172.22.0.15
63
64 # RTPENGINE
65 RTPENGINE_IP=172.22.0.16
66
67 # MYSQL
68 MYSQL_IP=172.22.0.17
69
70 # PYHSS
71 PYHSS_IP=172.22.0.18
72 PYHSS_BIND_PORT=3875
73
74 # ICSCF
75 ICSCF_IP=172.22.0.19
76 ICSCF_BIND_PORT=3869
77
78 # SCSCF
79 SCSCF_IP=172.22.0.20
80 SCSCF_BIND_PORT=3870
81
82 # PCSCF
83 PCSCF_IP=172.22.0.21
84 PCSCF_BIND_PORT=3871
85
86 # SRSLTE ENB
87 SRS_ENB_IP=172.22.0.22
88
89 # UERANSIM
90 NR_GNB_IP=172.22.0.23
91 NR_UE_IP=172.22.0.24
92
93 UE1_IMEI=356938035643803
94 UE1_IMEISV=4370816125816151
95 UE1_IMSI=001010000053537
96 UE1_KI=95B2689668A3C24443CF45CB7781F066
97 UE1_OP=577CB1FDAA2F8C39C4D7618D249C36E9
98 UE1_AMF=8000
99
100 # OAI ENB
101 OAI_ENB_IP=172.22.0.25
102

```

```

103 # OPENS5GS WEBUI
104 WEBUI_IP=172.22.0.26
105
106 # PCF
107 PCF_IP=172.22.0.27
108
109 # NSSF
110 NSSF_IP=172.22.0.28
111
112 # BSF
113 BSF_IP=172.22.0.29
114
115 # ENTITLEMENT SERVER
116 ENTITLEMENT_SERVER_IP=172.22.0.30
117
118 # OSMOMSC
119 OSMOMSC_IP=172.22.0.31
120
121 # OSMOHLR
122 OSMOHLR_IP=172.22.0.32
123
124 # SMSC
125 SMSC_IP=172.22.0.33
126
127 # SRSLTE UE
128 SRS_UE_IP=172.22.0.34
129
130 # SCP
131 SCP_IP=172.22.0.35
132
133 # METRICS
134 METRICS_IP=172.22.0.36
135
136 # UE IPv4 Subnet Range for APN=internet
137 UE_IPV4_INTERNET=192.168.100.0/24
138
139 # UE IPv4 Subnet Range for APN=ims
140 UE_IPV4_IMS=192.168.101.0/24
141
142 # Maximum Number of UEs
143 MAX_NUM_UE=1024
144
145 # Srsran Metrics:
146 ## InfluxDB
147
148 DOCKER_INFLUXDB_VERSION=2.7
149
150 DOCKER_INFLUXDB_INIT_MODE=setup
151 DOCKER_INFLUXDB_INIT_USERNAME=admin
152 DOCKER_INFLUXDB_INIT_PASSWORD=admin1234
153 DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=605bc59413b7d5457d181ccf20f9fda15693f81b068d70396cc183081b264f3b
154 DOCKER_INFLUXDB_INIT_ORG=srs
155 DOCKER_INFLUXDB_INIT_BUCKET=srsran
156 DOCKER_INFLUXDB_INIT_RETENTION=4d
157
158 DOCKER_INFLUXDB_INIT_PORT=8086
159 DOCKER_INFLUXDB_INIT_HOST=influxdb
160
161 INFLUXD_LOG_LEVEL=error
162
163 INFLUXDB_IP=172.22.0.38
164

```

```

165 ## Grafana
166
167 GRAFANA_VERSION=10.1.5
168 GRAFANA_PORT=3000
169
170 GF_AUTH_ANONYMOUS_ENABLED=true
171 GF_LOG_LEVEL=warning
172
173 GF_DASHBOARDS_DEFAULT_HOME_DASHBOARD_PATH=/etc/dashboards/home.json
174 GF_DASHBOARDS_MIN_REFRESH_INTERVAL=1s
175
176 GRAFANA_DATASOURCE_INFLUXDB_UID=JOSE3g9KVz
177 GRAFANA_DATASOURCE_PROMETHEUS_UID=b0b4ba4e-1c4f-4c1a-9168-37458f0e1fbd
178
179 GRAFANA_IP=172.22.0.39
180
181 # FIRSTRUN
182 FIRSTRUN_IP=172.22.0.41
183

```

7.2.2. *named.conf*

Este fichero, localizado en el interior de la carpeta “dns”, cuenta con la configuración de las DNS del subsistema multimedia. Es importante modificar este documento con los DNS de la red que proporciona internet al sistema 5G.

En el caso de la Universidad de Jaén, debemos configurar este documento con sus propios DNS:

- 150.214.170.15
- 150.214.170.21
- 150.214.170.22

El fichero de configuración quedaría como se muestra a continuación.

```

1 options {
2     directory "/var/cache/bind";
3
4     // If there is a firewall between you and nameservers you want
5     // to talk to, you may need to fix the firewall to allow multiple
6     // ports to talk.  See http://www.kb.cert.org/vuls/id/800113
7
8     // If your ISP provided one or more IP addresses for stable
9     // nameservers, you probably want to use them as forwarders.
10    // Uncomment the following block, and insert the addresses replacing
11    // the all-0's placeholder.
12
13    forwarders {
14        150.214.170.15;
15        150.214.170.21;
16        150.214.170.22;
17    };
18
19    //=====
20    // If BIND logs error messages about the root key being expired,
21    // you will need to update your keys.  See https://www.isc.org/bind-keys
22    //=====
23    dnssec-validation no;
24    allow-query { any; };

```

```

25
26     auth-nxdomain no;    # conform to RFC1035
27     listen-on-v6 { any; };
28 };
29
30 //
31 // Do any local configuration here
32 //
33
34 // Consider adding the 1918 zones here, if they are not used in your
35 // organization
36 //include "/etc/bind/zones.rfc1918";
37 zone "IMS_DOMAIN" {
38     type master;
39     file "/etc/bind/ims_zone";
40 };
41
42 zone "EPC_DOMAIN" {
43     type master;
44     file "/etc/bind/epc_zone";
45 };
46
47 zone "PUB_3GPP_DOMAIN" {
48     type master;
49     file "/etc/bind/pub_3gpp_zone";
50 };
51
52 zone "e164.arpa" {
53     type master;
54     file "/etc/bind/e164.arpa";
55 };
56

```

7.2.3. gnb.yml

Este fichero, localizado dentro de la carpeta “srsran” cuenta con la configuración del gNodeB. Aquí podemos definir parámetros como la banda y frecuencia de operación, ancho de banda de canal o las tablas de codificación de canal a utilizar. También podemos modificar parámetros relacionados con la SDR, como son la ganancia de transmisión y recepción, o la frecuencia de muestreo.

```

1  amf:
2      addr: AMF_IP                                # The address or hostname of the AMF.
3      bind_addr: SRS_GNB_IP                        # A local IP that the gNB binds to for
      traffic from the AMF.
4
5  ru_sdr:
6      device_driver: uhd                           # The RF driver name.
7      device_args: type=b200                       # Optionally pass arguments to the
      selected RF driver.
8      clock:                                         # Specify the clock source used by the
      RF.
9      sync:                                         # Specify the sync source used by the
      RF.
10     srate: 40.32                                  # [x 1.92 Mhz] RF sample rate might
      need to be adjusted according to selected bandwidth.
11     otw_format: sc12
12     tx_gain: 89                                    # [0 - 89.8 dB] Transmit gain of the
      RF might need to adjusted to the given situation.
13     rx_gain: 76                                    # [0 - 76 dB] Receive gain of the RF
      might need to adjusted to the given situation.
14
15  cell_cfg:
16     dl_arfcn: 432000                              # ARFCN of the downlink carrier
      (center frequency).
17     band: 1                                         # The NR band.
18     channel_bandwidth_MHz: 20                     # Bandwidth in MHz. Number of PRBs will
      be automatically derived.

```

```

19   common_scs: 15                                # Subcarrier spacing in kHz used for
data.                                                # data.
20   plmn: "PLMN"                                    # PLMN broadcasted by the gNB.
21   tac: 1                                           # Tracking area code (needs to match
the core configuration).
22   pci: 1                                           # Physical cell ID.
23   nof_antennas_ul: 2                               # MIMO UL Antennas.
24   nof_antennas_dl: 2                              # MIMO DL Antennas.
25
26   pdsch:
27     mcs_table: qam64                               # Sets the MCS table to use for PDSCH.
Supported: [qam64, qam256]
28
29   pusch:
30     mcs_table: qam64                               # Sets the MCS table to use for PUSCH.
Supported: [qam64, qam256]
31
32   log:
33     filename: /mnt/srsran/gnb.log                  # Path of the log file.
34     all_level: warning                             # Logging level applied to all layers.
35
36   pcap:
37     mac_enable: false                             # Set to true to enable MAC-layer
PCAPs.
38     mac_filename: /mnt/srsran/gnb_mac.pcap         # Path where the MAC PCAP is stored.
39     ngap_enable: false                             # Set to true to enable NGAP PCAPs.
40     ngap_filename: /mnt/srsran/gnb_ngap.pcap       # Path where the NGAP PCAP is stored.
41     f1ap_enable: false                             # Set to true to enable F1AP PCAPs.
42     f1ap_filename: /mnt/srsran/gnb_f1ap.pcap       # Path where the F1AP PCAP is stored.
43     e1ap_enable: false                             # Set to true to enable E1AP PCAPs.
44     e1ap_filename: /mnt/srsran/gnb_e1ap.pcap       # Path where the E1AP PCAP is stored.
45     e2ap_enable: false                             # Set to true to enable E2AP PCAPs.
46     e2ap_filename: /mnt/srsran/gnb_e2ap.pcap       # Path where the E2AP PCAP is stored.
47     gtpu_enable: false                             # Set to true to enable GTP-U PCAPs.
48     gtpu_filename: /mnt/srsran/gnb_gtpu.pcap      # Path where the GTP-U PCAP is stored.
49
50   metrics:
51     enable_json_metrics: true                     # Enable reporting metrics in JSON
format
52     addr: METRICS_SERVER_IP                       # Metrics-server IP
53     port: 55555                                    # Metrics-server Port

```

7.2.4. qos.yml

Este fichero, localizado dentro de la carpeta “srsran”, cuenta con la configuración necesaria para la calidad de servicio de la interfaz radio según el tipo de tráfico cursado.

```

1  # Quality of Service (QoS) example configurations for 5QI 1, 2, 5, 7 and 9
2  # Based on 5QI characteristics in TS 23.501 table 5.7.4-1
3
4  # This is a supplementary configuration to modify the RLC and PDCP radio bearers on
5  # a per 5QI basis.
6
7  qos:
8    -
9      five_qi: 1 # E.g. Conversational Voice
10     rlc:
11       mode: um-bidir
12       um-bidir:
13         tx:
14           sn: 12
15           queue-size: 4096
16         rx:
17           sn: 12
18           t-reassembly: 50

```

```

19 pdcp:
20   integrity_required: false
21   tx:
22     sn: 12
23     discard_timer: -1
24     status_report_required: false
25   rx:
26     sn: 12
27     t_reordering: 80
28     out_of_order_delivery: false
29 fl_u_du:
30   backoff_timer: 10
31 fl_u_cu_up:
32   backoff_timer: 10
33 mac:
34   lc_priority: 4
35   lc_group_id: 1
36   bucket_size_duration_ms: 5
37   prioritized_bit_rate_kBps: 65537
38 -
39 five_qi: 2 # E.g. Conversational Video
40 rlc:
41   mode: um-bidir
42   um-bidir:
43     tx:
44       sn: 12
45       queue-size: 4096
46     rx:
47       sn: 12
48       t_reassembly: 50
49 pdcp:
50   integrity_required: false
51   tx:
52     sn: 12
53     discard_timer: -1
54     status_report_required: false
55   rx:
56     sn: 12
57     t_reordering: 80
58     out_of_order_delivery: false
59 fl_u_du:
60   backoff_timer: 10
61 fl_u_cu_up:
62   backoff_timer: 10
63 mac:
64   lc_priority: 4
65   lc_group_id: 1
66   bucket_size_duration_ms: 5
67   prioritized_bit_rate_kBps: 65537
68 -
69 five_qi: 5 # E.g. IMS signaling
70 rlc:
71   mode: am
72   am:
73     tx:
74       sn: 12
75       t_poll_retransmit: 80
76       max_retx_threshold: 4
77       poll_pdu: 64
78       poll_byte: 125
79       queue-size: 4096
80     rx:
81       sn: 12
82       t_reassembly: 80
83       t_status_prohibit: 10
84 pdcp:
85   integrity_required: false
86   tx:
87     sn: 12
88     discard_timer: -1
89     status_report_required: false

```

```

90     rx:
91         sn: 12
92         t_reordering: 80
93         out_of_order_delivery: false
94     f1u_du:
95         backoff_timer: 10
96     f1u_cu_up:
97         backoff_timer: 10
98     mac:
99         lc_priority: 5
100        lc_group_id: 2
101        bucket_size_duration_ms: 5
102        prioritized_bit_rate_kBps: 65537
103 -
104 five_qi: 7 # E.g. Voice, Video (live streaming)
105 rlc:
106     mode: um-bidir
107     um-bidir:
108         tx:
109             sn: 12
110             queue-size: 4096
111         rx:
112             sn: 12
113             t-reassembly: 50
114     pdcp:
115         integrity_required: false
116         tx:
117             sn: 12
118             discard_timer: -1
119             status_report_required: false
120         rx:
121             sn: 12
122             t_reordering: 80
123             out_of_order_delivery: false
124     f1u_du:
125         backoff_timer: 10
126     f1u_cu_up:
127         backoff_timer: 10
128     mac:
129         lc_priority: 4
130         lc_group_id: 1
131         bucket_size_duration_ms: 5
132         prioritized_bit_rate_kBps: 65537
133 -
134 five_qi: 9 # E.g. Buffered video streaming, TCP-based traffic
135 rlc:
136     mode: am
137     am:
138         tx:
139             sn: 12
140             t-poll-retransmit: 80
141             max-retx-threshold: 4
142             poll-pdu: 64
143             poll-byte: 125
144             queue-size: 4096
145         rx:
146             sn: 12
147             t-reassembly: 80
148             t-status-prohibit: 10
149     pdcp:
150         integrity_required: false
151         tx:
152             sn: 12
153             discard_timer: -1
154             status_report_required: false
155         rx:
156             sn: 12
157             t_reordering: 80
158             out_of_order_delivery: false

```

```
159     f1u_du:
160         backoff_timer: 10
161     f1u_cu_up:
162         backoff_timer: 10
163     mac:
164         lc_priority: 5
165         lc_group_id: 2
166         bucket_size_duration_ms: 5
167         prioritized_bit_rate_kBps: 65537
168
```

7.3. Anexo III: Scripts elaborados para la configuración del sistema

A continuación, se exponen los distintos scripts que se han elaborado para configurar distintos aspectos del sistema 5G.

7.3.1. *sim-program*

Se trata de un programa que permite programar fácilmente las distintas tarjetas SIM disponibles en el laboratorio.

```
1  #!/bin/bash
2
3  help() {
4      echo "pySIM and docker_open5gs must be in the same directory"
5
6      echo "Use: ./sim-program [options]"
7      echo "Options:"
8      echo "    card [IMSI last number (0-9)]"
9      echo "    coims install [IMSI last number (0-9)]    Install certificates to use CoIMS app"
10     echo "    coims list [IMSI last number (0-9)]       List of installed certificates on the SIM"
11     echo "    scan                                     Execute pcsc_scan"
12     echo "    help                                     Shows pySim help"
13 }
14
15 gp="java -jar $PWD/gp.jar"
16
17 OPERATOR=UJA5G
18 COUNTRY_CODE=0
19 MCC=001
20 MNC=01
21 SMSC=0005555
22 ACC=0001
23 [ ${#MNC} == 3 ] && IMS_HDOMAIN="ims.mnc${MNC}.mcc${MCC}.3gppnetwork.org" || IMS_HDOMAIN="
ims.mnc0${MNC}.mcc${MCC}.3gppnetwork.org"
24
25 PIN_ADM=(85903454 42970400 99089163 95466357 67215927 26323548 98936148 83173246 59879048 50381267)
26 ICCID=(8988211000000535305 8988211000000535313 8988211000000535321 8988211000000535339
8988211000000535347 8988211000000535354 8988211000000535362 8988211000000535370 8988211000000535388
8988211000000535396)
27 MSISDN=(0001110 0001111 0001112 0001113 0001114 0001115 0001116 0001117 0001118 0001119)
28 KI=(1B7F04D495107F31E8DF29E5514F28CA 58D702BE6603FA099D468D1296BC4072 EFF1295624972C632B870D7863A4FC9F
C581839E3F6AB86EE564F74150BF5D28 804786B423891CF2A8FA8FBD45E9D237 D1DACEF6CFDEFB4A7B5F7C3225C4D676
32364C8E8C9162E0BFC4FBD7AEDB4AE 95B2689668A3C24443CF45CB7781F066 09498A36425BCE313DC932DEAC3D63CB
E6E0652EA80C26E70F6F343BAB533C12)
29 OPC=(DD71BC1880AF9367E4A52ECABAC326FD 46FB1EA4A7A571F9DCF3E2ABC58FC9FD D97E5A24512DFAEDFE4EA3035AB8CFDE
02A9C1F2B246834BB64F023AA9595FDF F006BC07EC19D737CE0D042F7BBE79AC 8A9EED3E44D4C835121142641E7F836C
45F1BCC5F57668DEEC17C1D7B09C4811 577CB1FDAA2F8C39C4D7618D249C36E9 B8E26F33567C588E103D1BF5FE1F002D
C7D61C44E23706765D0457A21A1B803A)
30 ID=(0000053530 0000053531 0000053532 0000053533 0000053534 0000053535 0000053536 0000053537 0000053538
0000053539)
31
32 KIC1=(941DC7B74BEDC136A643CA3621D3772C 808129832B805C59799E82D712DB6421
68BC0B326B792E603B06C3FDCDD4C9BD E688922CA449BE8FCC0272FD4CC2796A 974C94E3B17F8F896FAB46DF48B8DF75
8E53A004B71769FDA6577B65460A3E48 A5F3AB6E6BF4E362479168B7D73E3D2B ADC6B83308091044D3D3513995A28653
4A4C99642BA8D7F0D8F3E5FCF902F997 D4FB40C3C732B1AECFCE3A7591B937D8)
33
34 KID1=(D163691A26AEA5D3D4F11EFC27886FA1 A3B6ECE3A4D94FDB7EA3A91FC6690E48
55410BC33C39F6EA8B2917EAF5F43E5E F8B637F54D631935C14F59964D7B9FD8 DD0CC133D2F7AAF17943D7B6E2FFBDC5
F10D4FEDE8FB5DAD05D0ADEC7E575F8F 03A643214A897F250C5A75B453254C0D B59BC0F9FAC7BAFBE1136B8CF5DBF082
EDD330669BA48B7128991319EA186320 ABC49A1ADE23711AFFDAE2ED468D3CB0)
35
36 KIK1=(8ACB9DFF5A09D156D62107016C1B5E96 D5EA389093F96E1E0D7FEE5E301FD199
373C3632484966647220A280A697498D 05D1DC6F288727565E499FB0E11C6144 AAC5518556C098A2465DDF5D8AD01D61
36085AD97808EF2D2B0CF64431DCA9EC 1BE4B4D9F1C31A4758796A87B31B6F5F 9F88AD5F2916CB7A867F828E0DF422BF
7872ED7FF23D084C7D365CF632CC60CA 94C2B712C386714032CC0DF141967277)
37
```

```

38 if [ "$1" == "card" ]; then
39     if [[ -n $2 && 0 -le $2 && $2 -le 9 ]]; then
40         echo "../pysim/pySim-prog.py -p0 \
41             -a ${PIN_ADM[$2]} \
42             -n $OPERATOR \
43             -c $COUNTRY_CODE \
44             -x $MCC \
45             -y $MNC \
46             -m $SMSC \
47             -s ${ICCID[$2]} \
48             -i ${MCC}${MNC}${ID[$2]} \
49             --msisdn=${MSISDN[$2]} \
50             -k ${KI[$2]} \
51
52             -o ${OPC[$2]} \
53             --acc=$ACC"
54         # --write-csv=prueba.csv \
55         # --ims-hdomain=$IMS_HDOMAIN \
56         # --impi=${MCC}${MNC}${ID[$2]}@$IMS_HDOMAIN \
57         # --impu="sip:${MCC}${MNC}${ID[$2]}@$IMS_HDOMAIN" \
58
59         exit 1
60     fi
61 fi
62
63 if [ "$1" == "coims" ]; then
64     if [ "$2" == "install" ]; then
65         if [[ -n $3 && 0 -le $3 && $3 -le 9 ]]; then
66             alias gp="java -jar $PWD/gp.jar"
67
68             $gp --key-enc ${KIC1[$3]} --key-mac ${KID1[$3]} --key-dek ${KIK1[$3]} --install applet.cap
69
70             $gp --key-enc ${KIC1[$3]} --key-mac ${KID1[$3]} --key-dek ${KIK1[$3]} -a
71             00A4040009A00000015141434C0000 -a 80E2900033F031E22FE11E4F06FFFFFFFFFFFFC1
72             14E46872F28B350B7E1F140DE535C2A8D5804F0BE3E30DD00101DB08000000000000001
73
74             exit 1
75         fi
76     fi
77
78     if [ "$2" == "list" ]; then
79         if [[ -n $3 && 0 -le $3 && $3 -le 9 ]]; then
80             $gp --key-enc ${KIC1[$3]} --key-mac ${KID1[$3]} --key-dek ${KIK1[$3]} --acr-list-aram
81
82             exit 1
83         fi
84     fi
85 fi
86
87 if [ "$1" == "scan" ]; then
88     pcsc_scan
89
90     exit 1
91 fi
92
93 if [ "$1" == "help" ]; then
94     ../pysim/pySim-prog.py -help
95
96     exit 1
97 fi
98
99 help
100
101 exit 1
102
103

```

7.3.2. open5gs-dbconf

Esta herramienta de configuración, basada a su vez en la herramienta open5gs-dbctl proporcionada con el código fuente de Open5GS, nos permite añadir abonados a la base de datos de Open5GS desde la línea de comandos.

```
1  #!/bin/bash
2
3  display_help() {
4      echo "open5gs-dbconf: Open5GS Database Configuration Tool"
5      echo "FLAGS: --db_uri=mongodb://localhost"
6      echo "COMMANDS: " >&2
7      echo "  remove {imsi}: removes a user from the database"
8      echo "  reset: WIPES OUT the database and restores it to an empty default"
9      echo "  help: displays this message and exits"
10     echo "  add [0-9/all]: adds a user to the database, {0-9} means SIM card IMSI last number"
11     echo "  showall: shows the list of subscriber in the db"
12     echo "  showpretty: shows the list of subscriber in the db in a pretty json tree format"
13     echo "  showfiltered: shows {imsi key opc apn ip} information of subscriber"
14     echo "  install: installs mongosh"
15
16 }
17
18 MSISDN=(0001110 0001111 0001112 0001113 0001114 0001115 0001116 0001117 0001118 0001119)
19 KI=(1B7F04D495107F31E8DF29E5514F28CA 58D702BE6603FA099D468D1296BC4072 EFF1295624972C632BB70D7863A4FC9F
C581839E3F6AB86EE564F74150BF5D28 804786B423891CF2A8FA8FBD45E9D237 D1DACEF6CFDEFB4A7B5F7C3225C4D676
32364C8E8C9162E0BFC4FBD7AEADB4AE 95B2689668A3C24443CF45CB7781F066 09498A36425BCE313DC932DEAC3D63CB
E6E0652EA80C26E70F6F343BAB533C12)
20 OPC=(DD71BC1880AF9367E4A52ECABAC326FD 46FB1EA4A7A571F9DCF3E2ABC58FC9FD D97E5A24512DFAEDFE4EA3035AB8CFDE
02A9C1F2B246834BB64F023AA9595FDF F006BC07EC19D737CE0D042F7BBE79AC 8A9EED3E44D4C835121142641E7F836C
45F1BCC5F57668DEEC17C1D7B09C4811 577CB1FDAA2F8C39C4D7618D249C36E9 B8E26F33567C588E103D1BF5FE1F002D
C7D61C44E23706765D0457A21A1B803A)
21 ID=(0000053530 0000053531 0000053532 0000053533 0000053534 0000053535 0000053536 0000053537 0000053538
0000053539)
22 MCC=001
23 MNC=01
24 APN1="internet"
25 APN2="ims"
26
27 addSubscriber() {
28     mongosh --eval "db.subscribers.insertOne(
29         {
30             \"_id\": new ObjectId(),
31             \"schema_version\": NumberInt(1),
32             \"imsi\": \"\${MCC}\${MNC}\${IDS}\",
33             \"msisdn\": [\"$\{MSISDNs}\",],
34             \"imeisv\": [],
35             \"mme_host\": [],
36             \"mm_realm\": [],
37             \"purge_flag\": [],
38             \"slice\":[
39                 {
40                     \"sst\": NumberInt(1),
41                     \"default_indicator\": true,
42                     \"session\": [
43                         {
44                             \"name\" : \"\${APN1}\",
45                             \"type\" : NumberInt(1),
46                             \"qos\" :
47                             { \"index\": NumberInt(9),
48                               \"arp\":
49                               {
50                                   \"priority_level\" : NumberInt(8),
51                                   \"pre_emption_capability\": NumberInt(1),
52                                   \"pre_emption_vulnerability\": NumberInt(1)
53                               }
54                             },
55                             \"ambr\":
56                             {
57                                 \"downlink\":
58                                 {
59                                     \"value\": NumberInt(1),
```

```

60         \unit\": NumberInt(3)
61     },
62     \uplink\":
63     {
64         \value\": NumberInt(1),
65         \unit\": NumberInt(3)
66     }
67 },
68 \pcc_rule\": [],
69 \_id\": new ObjectId(),
70 },
71 {
72     \name\": \ "$APN2\",
73     \type\": NumberInt(1),
74     \qos\":
75     { \index\": NumberInt(5),
76       \arp\":
77       {
78         \priority_level\": NumberInt(1),
79         \pre_emption_capability\": NumberInt(1),
80         \pre_emption_vulnerability\": NumberInt(1)
81       }
82     },
83     \ambr\":
84     {
85         \downlink\":
86         {
87             \value\": NumberInt(3850),
88             \unit\": NumberInt(1)
89         },
90         \uplink\":
91         {
92             \value\": NumberInt(1530),
93             \unit\": NumberInt(1)
94         }
95     },
96     \pcc_rule\": [
97     {
98         \qos\": {
99             \arp\": {
100                 \priority_level\": NumberInt(2),
101                 \pre_emption_capability\": NumberInt(2),
102                 \pre_emption_vulnerability\": NumberInt(2)
103             },
104             \mbr\": {
105                 \downlink\": {
106                     \value\": NumberInt(128),
107                     \unit\": NumberInt(1)
108                 },
109                 \uplink\": {
110                     \value\": NumberInt(128),
111                     \unit\": NumberInt(1)
112                 }
113             },
114             \gbr\": {
115                 \downlink\": {
116                     \value\": NumberInt(128),
117                     \unit\": NumberInt(1)
118                 },
119                 \uplink\": {
120                     \value\": NumberInt(128),
121                     \unit\": NumberInt(1)
122                 }
123             }
124         }
125     }
126 ],

```

```

124         \index\": NumberInt(1),
125     },
126     \_id\": new ObjectId(),
127     \Flow\": []
128 },
129 {
130     \qos\": {
131         \arp\": {
132             \priority_level\": NumberInt(4),
133             \pre_emption_capability\": NumberInt(2),
134             \pre_emption_vulnerability\": NumberInt(2)
135         },
136         \mbr\": {
137             \downlink\": {
138                 \value\": NumberInt(128),
139                 \unit\": NumberInt(1)
140             },
141             \uplink\": {
142                 \value\": NumberInt(128),
143                 \unit\": NumberInt(1)
144             }
145         },
146         \gbr\": {
147             \downlink\": {
148                 \value\": NumberInt(128),
149                 \unit\": NumberInt(1)
150             },
151             \uplink\": {
152                 \value\": NumberInt(128),
153                 \unit\": NumberInt(1)
154             }
155         },
156         \index\": NumberInt(2),
157     },
158     \_id\": new ObjectId(),
159     \Flow\": []
160 }
161 ],
162 \_id\": new ObjectId(),
163 }],
164 \_id\": new ObjectId(),
165 }],
166 \security\":
167 {
168     \k\": \${KIs}\",
169     \op\": null,
170     \opc\": \${OPCs}\",
171     \amf\": \"8000\",
172 },
173 \ambr\":
174 {
175     \downlink\": { \value\": NumberInt(1), \unit\": NumberInt(3)},
176     \uplink\": { \value\": NumberInt(1), \unit\": NumberInt(3)}
177 },
178 \access_restriction_data\": 32,
179 \network_access_mode\": 0,
180 \subscriber_status\": 0,
181 \operator_determined_barring\": 0,
182 \subscribed_rau_tau_timer\": 12,
183 \__v\": 0
184 }
185 );" $DB_URI
186 }
187

```

```

188 while test $# -gt 0; do
189     case "$1" in
190         --db_uri*)
191             DB_URI=`echo $1 | sed -e 's/^[^=]*=//g`
192             shift
193             ;;
194         *)
195             break
196             ;;
197     esac
198 done
199
200 DB_URI="${DB_URI:-mongodb://172.22.0.2/open5gs}"
201
202 if [ "$#" -lt 1 ]; then
203     display_help
204     exit 1
205 fi
206
207 if [ "$1" = "help" ]; then
208     display_help
209     exit 1
210 fi
211
212 if [ "$1" = "remove" ]; then
213     if [ "$#" -ne 2 ]; then
214         echo "open5gs-dbconf: incorrect number of args, format is \"open5gs-dbconf remove imsi\""
215         exit 1
216     fi
217
218     IMSI=$2
219     mongosh --eval "db.subscribers.deleteOne({'imsi': \"${IMSI}\"});" $DB_URI
220     exit $?
221 fi
222
223 if [ "$1" = "reset" ]; then
224     if [ "$#" -ne 1 ]; then
225         echo "open5gs-dbconf: incorrect number of args, format is \"open5gs-dbconf reset\""
226         exit 1
227     fi
228
229     mongosh --eval "db.subscribers.deleteMany({});" $DB_URI
230     exit $?
231 fi
232
233 if [ "$1" = "add" ]; then
234     if [ "$#" -eq 2 ]; then
235
236         if [ "$2" = "all" ]; then
237
238             for i in {0..9}
239             do
240                 IDs=${ID[$i]}
241                 MSISDNs=${MSISDN[$i]}
242                 KIs=${KI[$i]}
243                 OPCs=${OPC[$i]}
244
245                 echo "Adding subscriber:"
246                 echo "  IMSI: ${MCC}${MNC}${IDs}"
247                 echo "  MSISDN: ${MSISDNs}"
248                 echo "  KI: ${KIs}"
249                 echo "  OPC: ${OPCs}"
250                 echo ""
251             done

```

```

252         addSubscriber
253     done
254     exit 1
255 fi
256
257 if [[ -n $2 && 0 -le $2 && $2 -le 9 ]]; then
258
259     IDs=${ID[$2]}
260     MSISDNs=${MSISDN[$2]}
261     KIs=${KI[$2]}
262     OPCs=${OPC[$2]}
263
264     echo "Adding subscriber:"
265     echo "  IMSI: ${MCC}${MNC}${IDs}"
266     echo "  MSISDN: ${MSISDNs}"
267     echo "  KI: ${KIs}"
268     echo "  OPC: ${OPCs}"
269
270     addSubscriber
271     exit $?
272
273 fi
274
275 echo "open5gs-dbconf: incorrect number of args, format is \"open5gs-dbconf add [0-9/all]\""
276 exit 1
277 fi
278
279 echo "open5gs-dbconf: incorrect number of args, format is \"open5gs-dbconf add [0-9/all]\""
280 exit 1
281 fi
282
283 if [ "$1" = "showall" ]; then
284     mongosh --eval "db.subscribers.find()" $DB_URI
285     exit $?
286 fi
287
288 if [ "$1" = "showpretty" ]; then
289     mongosh --eval "db.subscribers.find().pretty()" $DB_URI
290     exit $?
291 fi
292
293 if [ "$1" = "showfiltered" ]; then
294     mongosh --eval "db.subscribers.find({},{'_id':0,'imsi':1,'security.k':1, 'security.opc':1, '
slice.session.name':1,'slice.session.ue.ipv4':1})" $DB_URI
295     exit $?
296 fi
297
298 if [ "$1" = "install" ]; then
299
300     wget https://downloads.mongodb.com/compass/mongodb-mongosh_2.1.4_amd64.deb
301     sudo dpkg -i mongodb-mongosh_2.1.4_amd64.deb
302     rm mongodb-mongosh_2.1.4_amd64.deb
303
304     exit 1
305 fi
306
307 display_help
308

```

7.3.3. pyhss-dbconf

Esta herramienta de configuración, nos permite añadir abonados a la base de datos del IMS desde la línea de comandos.

```
1  #!/bin/bash
2
3  MCC=001
4  MNC=01
5  KI=(1B7F04D495107F31E8DF29E5514F28CA 58D702BE6603FA099D468D12968C4072 EFF1295624972C632BB70D7863A4FC9F
   C581839E3F6AB86EE564F74150BF5D28 804786B423891CF2A8FA8FBD45E9D237 D1DACEF6CFDEFB4A7B5F7C3225C4D676
   32364C8E8C9162E0BFC4FBD7AEDB4AE 95B2689668A3C24443CF45CB7781F066 09498A36425BCE313DC932DEAC3D63CB
   E6E0652EA80C26E70F6F343BAB533C12)
6  OPC=(DD71BC1880AF9367E4A52ECABAC326FD 46FB1EA4A7A571F9DCF3E2ABC58FC9FD D97E5A24512DFAEDFE4EA3035AB8CFDE
   02A9C1F2B246834BB64F023AA9595FDF F006BC07EC19D737CE0D042F7BBE79AC 8A9EED3E44D4C835121142641E7F836C
   45F1BCC5F57668DEEC17C1D7B09C4811 577CB1FDAA2F8C39C4D7618D249C36E9 B8E26F33567C588E103D1BF5FE1F002D
   C7D61C44E23706765D0457A21A1B803A)
7  ID=(0000053530 0000053531 0000053532 0000053533 0000053534 0000053535 0000053536 0000053537 0000053538
   0000053539)
8  MSISDN=(0001110 0001111 0001112 0001113 0001114 0001115 0001116 0001117 0001118 0001119)
9
10 display_help() {
11     echo "pyhss-dbconf: PyHSS Database Configuration Tool"
12     echo "INSTRUCTIONS:"
13     echo "  First, add APN \"internet\" and \"ims\" (only once).\"
14     echo "  Next, add AUC for each SIM card.\"
15     echo "  Next, add subscriber for each SIM card.\"
16     echo "  Finally, add ims subscriber for each SIM card.\"
17     echo "FLAGS: --db_ip=127.0.0.1"
18     echo "COMMANDS: >&2"
19     echo "  apn"
20     echo "    apn put [APN]: Create new APN"
21     echo "    apn list: List all APNs"
22     echo "    apn get: Get all APNs info"
23     echo "    apn get [APN_id]: Get data for specified APN"
24     echo "    apn delete [APN_id]: Delete specified APN"
25     echo "  auc"
26     echo "    auc put [IMSI last number (0-9)]: Create new AUC"
27     echo "    auc list: List all AUCs"
28     echo "    auc delete [AUC_id]: Delete specified AUC"
29     echo "    auc get [AUC_id]: Get data for specified AUC"
30     echo "  subscriber"
31     echo "    subscriber put [IMSI last number (0-9)]: Create new subscriber"
32     echo "    subscriber list: List all subscribers"
33     echo "    subscriber get: Get all subscribers info"
34     echo "    subscriber get [subscriber_id]: Get data for specified subscriber"
35     echo "    subscriber delete [subscriber_id]: Delete specified subscriber"
36     echo "  ims_subscriber"
37     echo "    ims_subscriber put [IMSI last number (0-9)]: Create new ims_subscriber"
38     echo "    ims_subscriber list: List all ims_subscribers"
39     echo "    ims_subscriber get: Get all ims_subscribers info"
40     echo "    ims_subscriber get [ims_subscriber_id]: Get data for specified ims_subscriber"
41     echo "    ims_subscriber delete [ims_subscriber_id]: Delete specified ims_subscriber"
42     echo "  install: installs jq (needed to process JSON)"
43 }
44
45 JSON='[{"ip_version": 0, "apn_id": 6, "sgw_address": null, "apn_ambr_dl": 0, "qci": 9, "
   arp_preemption_capability": false, "arp_preemption_vulnerability": true, "last_modified": "2024-02-
   15T11:58:00Z", "pgw_address": null, "apn": "pru", "charging_characteristics": "0800", "apn_ambr_ul": 0,
   "arp_priority": 4, "charging_rule_list": null}]'
46
47 while test $# -gt 0; do
48     case "$1" in
49         --db_ip*)
50             DB_IP=`echo $1 | sed -e 's/^[^=]*=//g`
51             shift
52             ;;
53         *)
54             break
55         ;;
56     esac
57
```

```

58 done
59
60 DB_IP="{DB_IP:-127.0.0.1}"
61
62 if [ "$1" = "apn" ]; then
63
64     if [ "$2" = "get" ]; then
65         if [ "$#" -ne 2 ]; then
66             curl -s -X 'GET' \
67                 "http://$DB_IP:8080/apn/$3" \
68                 -H 'accept: application/json' | jq .
69             exit $?
70         fi
71         curl -s -X 'GET' \
72             "http://$DB_IP:8080/apn/list" \
73             -H 'accept: application/json' | jq .
74         exit $?
75     fi
76
77     if [ "$2" = "list" ]; then
78         curl -s -X 'GET' \
79             "http://$DB_IP:8080/apn/list" \
80             -H 'accept: application/json' | jq ".[] | {apn, apn_id}"
81
82         exit 1
83     fi
84
85     if [ "$2" = "put" ]; then
86         if [[ -n $3 ]]; then
87             APNID=$(curl -s -X 'PUT' \
88                 "http://$DB_IP:8080/apn/" \
89                 -H 'accept: application/json' \
90                 -H 'Content-Type: application/json' \
91                 -d "{ \
92                     \"apn\": \"$3\", \
93                     \"apn_ambr_dl\": 0, \
94                     \"apn_ambr_ul\": 0 \
95                 }" | jq .apn_id)
96             echo "APN $3 has id: $APNID"
97             exit $?
98         fi
99         echo "      apn put [APN]: Create new APN"
100        exit $?
101    fi
102
103    if [ "$2" = "delete" ]; then
104        if [[ -n $3 ]]; then
105            curl -s -X 'DELETE' \
106                "http://$DB_IP:8080/apn/$3" \
107                -H 'accept: application/json' | jq .
108            exit $?
109        fi
110        echo "      apn delete [APN_id]: Delete specified APN"
111        exit $?
112    fi
113
114    display_help
115    exit 1
116 fi
117
118 if [ "$1" = "auc" ]; then
119     if [ "$2" = "get" ]; then
120         curl -s -X 'GET' \
121             "http://$DB_IP:8080/auc/$3" \

```

```

122     -H 'accept: application/json' | jq .
123     exit $?
124 fi
125
126 if [ "$2" = "list" ]; then
127     curl -s -X 'GET' \
128         "http://$DB_IP:8080/auc/list" \
129         -H 'accept: application/json' | jq ".[] | {imsi, auc_id}"
130     exit 1
131 fi
132
133 if [ "$2" = "put" ]; then
134     if [[ -n $3 && 0 -le $3 && $3 -le 9 ]]; then
135         AUCID=$(curl -s -X 'PUT' \
136             "http://$DB_IP:8080/auc/" \
137             -H 'accept: application/json' \
138             -H 'Content-Type: application/json' \
139             -d "{ \
140                 \"ki\": \"${KI[$3]}\", \
141                 \"opc\": \"${OPC[$3]}\", \
142                 \"amf\": \"8000\", \
143                 \"sqn\": 0, \
144                 \"imsi\": \"${MCC}${MNC}${ID[$3]}\" \
145             }" | jq .auc_id)
146         if [[ $AUCID != null ]]; then
147             echo "IMSI ${MCC}${MNC}${ID[$3]} has id: $AUCID"
148             exit $?
149         fi
150         echo "IMSI ${MCC}${MNC}${ID[$3]} has been already registered"
151         exit $?
152     fi
153     echo "      auc put [IMSI last number (0-9)]: Create new AUC"
154     exit $?
155 fi
156
157 if [ "$2" = "delete" ]; then
158     if [[ -n $3 ]]; then
159         curl -s -X 'DELETE' \
160             "http://$DB_IP:8080/auc/$3" \
161             -H 'accept: application/json' | jq .
162         exit $?
163     fi
164     echo "      auc delete [AUC_id]: Delete specified AUC"
165     exit $?
166 fi
167
168 display_help
169 exit 1
170 fi
171
172 if [ "$1" = "subscriber" ]; then
173
174     if [ "$2" = "get" ]; then
175         if [ "$#" -ne 2 ]; then
176             curl -s -X 'GET' \
177                 "http://$DB_IP:8080/subscriber/$3" \
178                 -H 'accept: application/json' | jq .
179             exit $?
180         fi
181         curl -s -X 'GET' \
182             "http://$DB_IP:8080/subscriber/list" \
183             -H 'accept: application/json' | jq .
184         exit $?
185     fi

```

```

186
187 if [ "$2" = "list" ]; then
188     curl -s -X 'GET' \
189         "http://$DB_IP:8080/subscriber/list" \
190         -H 'accept: application/json' | jq ".[] | {imsi, msisdn, subscriber_id}"
191     exit 1
192 fi
193
194 if [ "$2" = "put" ]; then
195     if [[ -n $3 && 0 -le $3 && $3 -le 9 ]]; then
196
197         AUC_id=$(curl -s -X 'GET' \
198             "http://$DB_IP:8080/auc/list" \
199             -H 'accept: application/json' | jq "map(select(any(.imsi; contains(\"${MCC}${MNC}
200 ${ID[$3]}\")))|.auc_id)[0]")
201
202         INTERNET=$(curl -s -X 'GET' \
203             "http://$DB_IP:8080/apn/list" \
204             -H 'accept: application/json' | jq "map(select(any(.apn; contains(\"internet\")))
205             [.apn_id][0]")
206
207         IMS=$(curl -s -X 'GET' \
208             "http://$DB_IP:8080/apn/list" \
209             -H 'accept: application/json' | jq "map(select(any(.apn; contains(\"ims\")))|.apn_id
210             [0]")
211
212         if [[ $AUC_id != null ]] && [[ $INTERNET != null ]] && [[ $IMS != null ]]; then
213             SUBSCRIBER_ID=$(curl -s -X 'PUT' \
214                 "http://$DB_IP:8080/subscriber/" \
215                 -H 'accept: application/json' \
216                 -H 'Content-Type: application/json' \
217                 -d "{ \
218                     \"imsi\": \"${MCC}${MNC}${ID[$3]}\", \
219                     \"enabled\": true, \
220                     \"auc_id\": ${AUC_id}, \
221                     \"default_apn\": ${INTERNET}, \
222                     \"apn_list\": \"${INTERNET},${IMS}\", \
223                     \"msisdn\": \"${MSISDN[$3]}\", \
224                     \"ue_ambr_dl\": 0, \
225                     \"ue_ambr_ul\": 0 \
226                 }" | jq .subscriber_id)
227             if [[ $SUBSCRIBER_ID != null ]]; then
228                 echo "Subscriber ${MCC}${MNC}${ID[$3]} has id: $SUBSCRIBER_ID"
229                 exit $?
230             fi
231             echo "Subscriber ${MCC}${MNC}${ID[$3]} has been already registered"
232             exit $?
233         fi
234         echo "Subscriber ${MCC}${MNC}${ID[$3]} does not have auc_id or APNs Internet and IMS have
235         not been added"
236         exit $?
237     fi
238
239     if [ "$2" = "delete" ]; then
240         if [[ -n $3 ]]; then
241             curl -s -X 'DELETE' \
242                 "http://$DB_IP:8080/subscriber/$3" \
243                 -H 'accept: application/json' | jq .
244             exit $?
245         fi
246         echo "subscriber delete [subscriber_id]: Delete specified subscriber"
247         exit $?

```

```

247     fi
248
249     display_help
250     exit 1
251 fi
252
253 if [ "$1" = "ims_subscriber" ]; then
254
255     if [ "$2" = "get" ]; then
256         if [ "$#" -ne 2 ]; then
257             curl -s -X 'GET' \
258                 "http://$DB_IP:8080/ims_subscriber/$3" \
259                 -H 'accept: application/json' | jq .
260             exit $?
261         fi
262         curl -s -X 'GET' \
263             "http://$DB_IP:8080/ims_subscriber/list" \
264             -H 'accept: application/json' | jq .
265         exit $?
266     fi
267
268     if [ "$2" = "list" ]; then
269         curl -s -X 'GET' \
270             "http://$DB_IP:8080/ims_subscriber/list" \
271             -H 'accept: application/json' | jq ".[] | {imsi, msisdn, ims_subscriber_id}"
272         exit 1
273     fi
274
275     if [ "$2" = "put" ]; then
276         if [[ -n $3 && 0 -le $3 && $3 -le 9 ]]; then
277             IMS_SUBSCRIBER_ID=$(curl -s -X 'PUT' \
278                 "http://$DB_IP:8080/ims_subscriber/" \
279                 -H 'accept: application/json' \
280                 -H 'Content-Type: application/json' \
281                 -d "{ \
282                     \"imsi\": \"${MCC}${MNC}${ID[$3]}\", \
283                     \"msisdn\": \"${MSISDN[$3]}\", \
284                     \"sh_profile\": \"string\", \
285                     \"scscf_peer\": \"scscf.ims.mnc001.mcc001.3gppnetwork.org\", \
286                     \"msisdn_list\": \"[${MSISDN[$3]}]\", \
287                     \"ifc_path\": \"default_ifc.xml\", \
288                     \"scscf\": \"sip:scscf.ims.mnc001.mcc001.3gppnetwork.org:6060\", \
289                     \"scscf_realm\": \"ims.mnc001.mcc001.3gppnetwork.org\" \
290                 }" | jq .ims_subscriber_id)
291             if [[ $IMS_SUBSCRIBER_ID != null ]]; then
292                 echo "IMS subscriber ${MCC}${MNC}${ID[$3]} has id: $IMS_SUBSCRIBER_ID"
293                 exit $?
294             fi
295             echo "IMS subscriber ${MCC}${MNC}${ID[$3]} has been already registered"
296             exit $?
297         fi
298         echo "ims_subscriber put [IMSI last number (0-9)]: Create new ims_subscriber"
299         exit $?
300     fi
301
302     if [ "$2" = "delete" ]; then
303         if [[ -n $3 ]]; then
304             curl -s -X 'DELETE' \
305                 "http://$DB_IP:8080/ims_subscriber/$3" \
306                 -H 'accept: application/json' | jq .
307             exit $?
308         fi
309         echo "ims_subscriber delete [ims_subscriber_id]: Delete specified ims_subscriber"
310         exit $?

```

```

311     fi
312
313     display_help
314     exit 1
315 fi
316
317 if [ "$1" = "install" ]; then
318     sudo apt-get install -y jq
319     exit 1
320 fi
321
322 display_help
323
324

```

7.3.4. dbconf-all.sh

Este script de Bash simplemente ejecuta los programas “open5gs-dbconf” y “pyhss-dbconf” de forma secuencial para cargar todos los abonados a ambas bases de datos.

```

1  #!/bin/bash
2
3  printf "\033[32mAdding subscribers to Open5gs core DB\n\n\033[0m"
4  ./mnt/firstrun/open5gs-dbconf --db_uri=$MONGO_IP/open5gs add all
5
6  printf "\033[32mConfiguring pyHSS DB\n\n\033[0m"
7
8  printf "\033[35m\nAdding APN internet\n\033[0m"
9  ./mnt/firstrun/pyhss-dbconf --db_ip=$PYHSS_IP apn put internet
10 printf "\033[35m\nAdding APN ims\n\033[0m"
11 ./mnt/firstrun/pyhss-dbconf --db_ip=$PYHSS_IP apn put ims
12
13 for i in {0..9}
14 do
15     printf "\033[35m\nAdding auc for SIM $i\n\033[0m"
16     ./mnt/firstrun/pyhss-dbconf --db_ip=$PYHSS_IP auc put $i
17     printf "\033[35m\nAdding subscriber for SIM $i\n\033[0m"
18     ./mnt/firstrun/pyhss-dbconf --db_ip=$PYHSS_IP subscriber put $i
19     printf "\033[35m\nAdding ims_subscriber for SIM $i\n\033[0m"
20     ./mnt/firstrun/pyhss-dbconf --db_ip=$PYHSS_IP ims_subscriber put $i
21 done
22
23 printf "\033[1;32m\nDONE\n\n\033[0m"
24

```

8. BIBLIOGRAFÍA

1. **Emilio-blogger.** La evolución de la telefonía móvil, del "zapatófono" al 5G". [En línea] Movistar, 07 de julio de 2021. [Citado el: 20 de abril de 2024.] <https://comunidad.movistar.es/t5/Blog-Movisfera/La-evoluci%C3%B3n-de-la-telefon%C3%ADa-m%C3%B3vil-del-quot-zapat%C3%B3fono-quot-al/ba-p/4531705>.

2. **Wikipedia contributors.** 1G. [En línea] Wikipedia, The Free Encyclopedia, 07 de abril de 2024. [Citado el: 20 de abril de 2024.] <https://en.wikipedia.org/w/index.php?title=1G&oldid=1217677982>.

3. **Rodríguez, Emilio.** Historias de Telefónica: MoviLine. [En línea] Movistar, 28 de diciembre de 2023. <https://www.movistar.es/blog/home/historias-telefonica-moviline/>.

4. **Colaboradores de Wikipedia.** Sistema Telefónico Móvil Avanzado. [En línea] Wikipedia, La enciclopedia libre, 8 de agosto de 2023. [Citado el: 20 de abril de 2024.] https://es.wikipedia.org/w/index.php?title=Sistema_Telef%C3%B3nico_M%C3%B3vil_Avanzado&oldid=152943704.

5. **Wikipedia contributors.** 2G. [En línea] Wikipedia, The Free Encyclopedia, 3 de mayo de 2024. [Citado el: 4 de mayo de 2024.] <https://en.wikipedia.org/w/index.php?title=2G&oldid=1221958093>.

6. **Colaboradores de Wikipedia.** 3GPP. [En línea] Wikipedia, La enciclopedia libre, 8 de abril de 2023. [Citado el: 5 de mayo de 2024.] <https://es.wikipedia.org/w/index.php?title=3GPP&oldid=150415839>.

7. **Wikipedia contributors.** 3rd Generation Partnership Project 2. [En línea] Wikipedia, The Free Encyclopedia, 27 de marzo de 2024. [Citado el: 5 de mayo de 2024.] https://en.wikipedia.org/w/index.php?title=3rd_Generation_Partnership_Project_2&oldid=1215876500.

8. —. IMT-2000. [En línea] Wikipedia, The Free Encyclopedia, 23 de abril de 2024. [Citado el: 5 de mayo de 2024.] <https://en.wikipedia.org/w/index.php?title=IMT-2000&oldid=1220416635>.

9. **Colaboradores de Wikipedia.** Telefonía móvil 3G. [En línea] Wikipedia, La enciclopedia libre, 15 de enero de 2024. [Citado el: 5 de mayo de 2024.] https://es.wikipedia.org/w/index.php?title=Telefon%C3%ADa_m%C3%B3vil_3G&oldid=157212804.

10. **Wikipedia contributors.** IMT Advanced. [En línea] Wikipedia, The Free Encyclopedia, 26 de abril de 2024. [Citado el: 5 de mayo de 2024.] https://en.wikipedia.org/w/index.php?title=IMT_Advanced&oldid=1220817631.

11. —. 4G. [En línea] Wikipedia, The Free Encyclopedia, 24 de abril de 2024. [Citado el: 5 de mayo de 2024.] <https://en.wikipedia.org/w/index.php?title=4G&oldid=1220567681>.

12. **UIT-R.** *Recomendación UIT-R M.2083-0.* (09/2015). Concepción de las IMT - Marco y objetivos generales del futuro desarrollo de las IMT para 2020 y en adelante.

13. —. *Report ITU-R M.2410-0.* (11/2017). Minimum requirements related to technical performance for IMT-2020 radio interface(s).

14. **Wikipedia contributors.** 3GPP. [En línea] Wikipedia, The Free Encyclopedia, 25 de abril de 2024. [Citado el: 20 de mayo de 2024.] <https://en.wikipedia.org/w/index.php?title=3GPP&oldid=1220742111>.

15. *Ericsson Technology Review.* **Rahman Imadur, Sara Modarres Razavi, Olof Liberg, Christian Hoymann, Henning Wiemann, Claes Tidestav, Paul Schliwa-Bertling, Patrik Persson, Dirk Gerstenberger.** s.l. : Ericsson, 13 de octubre de 2021.

16. **Gaurav Gangwal, Kevin Gray.** *Dell Technologies.* [En línea] 17 de agosto de 2023. [Citado el: 3 de junio de 2024.] <https://infohub.delltechnologies.com/es-es/p/the-5g-core-network-demystified/>.

17. **Alain Sultan, MCC.** 5G System Overview. [En línea] 3GPP Portal, 8 de agosto de 2022. [Citado el: 5 de junio de 2024.] <https://www.3gpp.org/technologies/5g-system-overview>.

18. **Karirom.** RAN - Arquitectura de 5G. [En línea] Comunidad Huawei Enterprise, 31 de enero de 2024. [Citado el: 05 de junio de 2024.] <https://forum.huawei.com/enterprise/es/core-de-red-5g-arquitectura-de-5g/thread/752733987165384704-667212896691695616>.

19. **Hernando, José María, Vázquez, Juan Manuel y Lluch, Cayetano.** Modulación y Multiacceso en 5G-NR. [En línea] Portal de la 5G - ETSIT UPM. <http://www.grc.ssr.upm.es/Portal%205G/subsitio/Informes%20y%20Notas/Modulaci%C3%B3n%20y%20multiacceso%205GNR.pdf>.

20. **3GPP**. 5G; NR; Physical channels and modulation (3GPP TS 38.211 version 18.2.0 Release 18). [En línea] 27 de mayo de 2024. [Citado el: 20 de junio de 2024.] https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/18.02.00_60/ts_138211v180200p.pdf.
21. **OpenAirInterface.org**. OAI 5G RAN PROJECT GROUP. [En línea] 2024. [Citado el: 5 de junio de 2024.] <https://openairinterface.org/oai-5g-ran-project/>.
22. —. 5G CORE NETWORK. [En línea] 2024. [Citado el: 5 de junio de 2024.] <https://openairinterface.org/oai-5g-core-network-project/>.
23. **Open5GS**. Quickstart. [En línea] 5 de junio de 2024. [Citado el: 5 de junio de 2024.] <https://open5gs.org/open5gs/docs/guide/01-quickstart/>.
24. **free5GC**. [En línea] <https://free5gc.org/>.
25. —. free5GC Specification. [En línea] [Citado el: 5 de junio de 2024.] <https://free5gc.org/guide/features/>.
26. **GÜNGÖR, ALİ**. UERANSIM. [En línea] 8 de octubre de 2023. [Citado el: 11 de junio de 2024.] <https://github.com/aligungr/UERANSIM>.
27. —. Feature Set. [En línea] 1 de marzo de 2022. [Citado el: 11 de junio de 2024.] <https://github.com/aligungr/UERANSIM/wiki/Feature-Set>.
28. **Software Radio Systems**. srsRAN gNB with srsUE. [En línea] 2024 de mayo de 16. [Citado el: 11 de junio de 2024.] <https://docs.srsran.com/projects/project/en/latest/tutorials/source/srsUE/source/index.html>.
29. —. Features and Roadmap. [En línea] 16 de abril de 2024. [Citado el: 11 de junio de 2024.] https://docs.srsran.com/projects/project/en/latest/general/source/2_features_and_roadmap.html.
30. **Colaboradores de Wikipedia**. Radio definida por software. [En línea] Wikipedia, La enciclopedia libre, 26 de septiembre de 2023. [Citado el: 11 de junio de 2024.] https://es.wikipedia.org/w/index.php?title=Radio_definida_por_software&oldid=154048626.
31. **Ettus Research**. USRP B210 USB Software Defined Radio (SDR). [En línea] NI. [Citado el: 11 de junio de 2024.] <https://www.ettus.com/all-products/ub210-kit/>.

32. —. USRP X310 High Performance Software Defined Radio. [En línea] NI. [Citado el: 11 de junio de 2024.] <https://www.ettus.com/all-products/x310-kit/>.
33. —. USRP N310. [En línea] NI. [Citado el: 11 de junio de 2024.] <https://www.ettus.com/all-products/usrp-n310/>.
34. **Lime Microsystems Ltd.** LimeSDR. [En línea] [Citado el: 11 de junio de 2024.] <https://limemicro.com/products/boards/limesdr/>.
35. **WiMo.** LimeSDR. [En línea] [Citado el: 11 de junio de 2024.] <https://www.wimo.com/es/limesdr>.
36. **astroradio.** RTL-SDR Blog R860 RTL2832U V3. [En línea] [Citado el: 11 de junio de 2024.] https://www.astroradio.com/p/receptor_rtlsdr_blog_r820t2_rtl2832u/.
37. **Great Scott Gadgets.** HackRF One. [En línea] [Citado el: 11 de junio de 2024.] <https://greatscottgadgets.com/hackrf/one/>.
38. **Astoradio.** HackRF One. [En línea] [Citado el: 11 de junio de 2024.] <https://www.astroradio.com/p/hackrf-one/>.
39. **nuand.** bladeRF. [En línea] [Citado el: 11 de junio de 2024.] <https://www.nuand.com/bladerf-1/>.
40. **Colaboradores de Wikipedia.** Grafana. [En línea] Wikipedia, La enciclopedia libre, 13 de marzo de 2024. [Citado el: 13 de junio de 2024.] <https://es.wikipedia.org/w/index.php?title=Grafana&oldid=158783713>.
41. **Wikipedia contributors.** [En línea] Wikipedia, The Free Encyclopedia, 11 de febrero de 2024. [Citado el: 13 de junio de 2024.] <https://en.wikipedia.org/w/index.php?title=InfluxDB&oldid=1206253416>.
42. **Docker Inc.** Install Docker Engine. [En línea] [Citado el: 8 de junio de 2024.] <https://docs.docker.com/engine/install/>.