



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



PortaRAN: Implementación de un Demostrador Portable de Red Móvil 5G basado en srsRAN

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE
INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Facundo Pedreira, Bruno Tió

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA FINALIZACIÓN DE LA CARRERA DE
INGENIERÍA EN SISTEMAS DE COMUNICACIÓN.

TUTOR

Claudina Rattaro Universidad de la República
Leonardo Barboni..... Universidad de la República

TRIBUNAL

Álvaro Valdés..... Universidad de la República
Romina García Universidad de Pennsylvania, Universidad de la
República
Raúl Hartmam..... Universidad de la República
Jan García Morales Universidad del Rey Juan Carlos

Montevideo
viernes 8 agosto, 2025

PortaRAN: Implementación de un Demostrador Portable de Red Móvil 5G basado en srsRAN, Facundo Pedreira, Bruno Tió.

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.2).
Contiene un total de 135 páginas.
Compilada el viernes 8 agosto, 2025.
<http://iie.fing.edu.uy/>

Agradecimientos

Mediante estos párrafos queremos agradecer a todos quienes estuvieron involucrados, de un modo u otro, en el desarrollo de este proyecto. Fueron muchos meses de trabajo y estamos convencidos de que no hubiese sido posible lograrlo sin el apoyo de cada uno.

En primer lugar, expresar nuestra gratitud hacia nuestros tutores Claudina y Leonardo. Siendo guías y referentes, estuvieron siempre atentos y a completa disposición. Gracias por acercarnos este proyecto, que fue una gran oportunidad para adentrarnos en un mundo fascinante. Además, destacar que Claudina fue nuestra directora de carrera durante todo el transcurso de esta, siendo una mentora más allá del aula.

En segundo lugar, agradecer a nuestras familias, parejas y amigos. Su apoyo incondicional, paciencia y palabras de aliento en las etapas más desafiantes fueron el sostén que nos permitió seguir adelante. Gracias por ser nuestro refugio y motivación cuando el camino se hacía largo.

En tercer lugar, extendemos nuestro agradecimiento a la Universidad de la República, la Facultad de Ingeniería y, en especial, al Instituto de Ingeniería Eléctrica, nuestra casa de estudios y formación profesional. Gracias por brindarnos las herramientas académicas, los espacios de laboratorio y el entorno de aprendizaje que hicieron posible este proyecto. En particular, mencionar a Pablo Vázquez y Horacio Duffau, por su disposición y sus aportes.

Por último, a los varios foros y comunidades consultados, principalmente la de *srsRAN*, donde encontramos un grupo de personas dispuestas a ayudar y compartir información de forma abierta y desinteresada. A todos quienes ayudan a llevar a cabo tanto este, como muchos otros proyectos *open-source* que democratizan y extienden el acceso al conocimiento.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

El mundo de las redes móviles se encuentra en constante desarrollo y evolución. En los últimos años, el despliegue de redes de quinta generación (5G) se ha convertido en una prioridad global. En Uruguay, el primer despliegue comercial se concretó en 2023, aunque ya desde 2019 distintas compañías habían realizado pruebas piloto. Si bien hoy existen diversas implementaciones, contando con distintos tipos de *hardware* y *software*, muchas veces se trata de equipos de alto costo y propietarios de diferentes empresas. Esto dificulta el acceso a estas nuevas tecnologías a los equipos de investigación y al área educativa.

En la Universidad de la República y en particular en la Facultad de Ingeniería se han realizado diversas experiencias abarcadas por esta subárea de las telecomunicaciones, tomando en este caso como antecedente principal la realizada en el proyecto de grado titulado *Implementación de una Maqueta de Pruebas y Desarrollo de una Red 5G Stand Alone Completa* de W. Peña, W. Piastri y P. Vázquez. Durante ese trabajo se utilizó la solución de *software* *OpenAirInterface*, una de las principales opciones de código abierto que existen que permiten el despliegue de los elementos de la red de acceso de 5G. En este proyecto se tomó como objetivo utilizar la solución “competidora”, *srsRAN*, para implementar una maqueta de red 5G análoga, con el fin de contrastar ambas herramientas además de evaluar cuánto ha avanzado su desarrollo en el último tiempo.

Además de implementar una versión de la maqueta de red 5G utilizando el mismo *hardware* de alto costo del proyecto antecedente pero con otra solución base de *software*, se avanzó en la evaluación de distintas alternativas de *hardware* y *software* para el módulo del equipo de usuario (UE). Se incorporaron a la maqueta y se compararon tres variantes de UE: *srsUE*, *COTS UE* y *Hat Sixfab 5G* para *Raspberry Pi*. El primero es parte del paquete ofrecido por *srsRAN*, siendo una modificación de su versión 4G y no un UE 5G propio. Por otro lado, tanto el *COTS UE* utilizado como el *Hat Sixfab* sí poseen integración 5G nativa, diferencia que se evidenció en su desempeño.

Finalmente, como parte de la evaluación de la portabilidad de componentes en la red de acceso, este proyecto avanzó en la implementación de un concepto clave en arquitecturas 5G: la separación funcional de la estación base (gNB). En particular, se integró a la maqueta la capacidad de dividir la gNB en una Unidad Centralizada (CU) y una Unidad Distribuida (DU), siguiendo el *Split 2* definido por el estándar del 3GPP. A su vez, se trabajó con el *Split 8* entre la DU y la Unidad Remota (RU).

Se evaluaron dos soluciones para el enlace intermedio (*midhaul*) entre CU y DU: una conexión cableada mediante Ethernet y otra completamente inalámbrica.

ca utilizando Wi-Fi. De forma destacable, la DU fue desplegada exitosamente en un equipo portátil Intel NUC, logrando comunicación estable con la CU a través de Wi-Fi. También se evaluaron alternativas en el enlace frontal (*fronthaul*) utilizando diferentes equipos de *Software Defined Radio* como RU, con un enlace basado en 10GB Ethernet y uno basado en USB 3.0. Los resultados de este proyecto demuestran la viabilidad de estaciones base parcialmente inalámbricas, abriendo nuevas posibilidades para despliegues portables o de bajo costo en el ámbito académico, y estableciendo una base sólida para futuros estudios en este campo.

Tabla de contenidos

Agradecimientos	I
Resumen	III
1. Introducción	1
1.1. Contexto y motivación	1
1.2. Definición del problema	3
1.2.1. Objetivos generales	3
1.2.2. Objetivos específicos	4
1.2.3. Alcance y limitaciones	4
1.3. Antecedentes	5
1.3.1. Testbeds 5G	5
1.3.2. Otros antecedentes	7
1.4. Contribuciones	8
1.5. Estructura del documento	10
2. Marco teórico y soluciones de hardware y software utilizadas	11
2.1. Redes móviles 5G	11
2.1.1. Aspectos generales y casos de uso	11
2.1.2. Arquitectura de red 5G	14
2.1.3. Transmisión y acceso al medio	21
2.2. Software utilizado	23
2.2.1. El proyecto srsRAN	23
2.2.2. Open5GS	25
2.2.3. ZMQ	26
2.3. Software Defined Radio	26
2.3.1. USRP Ettus X310	27
2.3.2. USRP Ettus B200	28
3. Maqueta de red 5G	29
3.1. Estructura base de la maqueta	29
3.2. Procedimiento	33
3.2.1. Entorno virtualizado	33
3.2.2. Entorno físico	35
3.3. Caracterización de la maqueta	38

Tabla de contenidos

3.4.	Variantes en gNB	39
3.4.1.	Implementación de gNB en NUC	39
3.4.2.	Split CU/DU	41
3.5.	Variantes en UE	44
3.5.1.	Dispositivo COTS	45
3.5.2.	Dispositivo Sixfab	46
3.6.	Características y costos finales de la maqueta	47
4.	Dispositivos, pruebas y resultados preliminares	49
4.1.	Tests de desempeño	49
4.1.1.	Ganancias en antenas	52
4.1.2.	Ancho de banda	52
4.1.3.	Bandas del espectro	54
4.1.4.	Cobertura	55
4.1.5.	Variación de SDR	56
4.2.	Escenarios de prueba	58
4.2.1.	Tráfico concurrente	58
4.2.2.	Desempeño con Split 2	59
4.2.3.	Conexión a Internet	61
5.	Análisis de resultados	65
5.1.	Comparativa teórica con <i>OpenAirInterface</i>	65
5.2.	Establecimiento de conexión y análisis de desempeño	67
5.2.1.	Establecimiento de conexión inicial y rango de ganancias	67
5.2.2.	Evaluación de <i>throughput</i> por ancho de banda	68
5.3.	Comparativa práctica con <i>OAI</i>	69
5.3.1.	Análisis de anomalías	70
5.4.	Variación de parámetros en conexión física	71
5.4.1.	Impacto del ruido presente en la conexión	71
5.4.2.	Evaluación de diferentes bandas	72
5.4.3.	Impacto de la distancia física entre equipos	72
5.5.	Caracterización de Hardware	73
5.6.	Análisis de Configuraciones	75
5.6.1.	Variantes en UEs	75
5.6.2.	Rendimiento de splits	78
6.	Conclusiones y trabajo a futuro	81
6.1.	Cumplimiento de objetivos	81
6.2.	Conclusiones sobre la maqueta y sus variantes	83
6.3.	Trabajo a futuro	83
A.	Manual de instalación y configuraciones	85
B.	Evaluación de compra de dispositivos	93
C.	Kit de desarrollo Sixfab 5G	95

Tabla de contenidos

D. Reprogramación de SDR mediante puerto JTAG	99
E. Programación de SIMs mediante herramienta OpenCells	103
Glosario	105
Referencias	111
Índice de tablas	118
Índice de figuras	121

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 1

Introducción

1.1. Contexto y motivación

El mundo de las redes móviles está en constante evolución y el despliegue de redes de quinta generación (5G) ha sido el eje central en los últimos años a nivel mundial. Con las comunicaciones mejoradas de banda ancha móvil (eMBB), 5G soporta altas velocidades de transmisión de datos para aplicaciones como la realidad virtual y aumentada. Las comunicaciones ultra fiables y de baja latencia (URLLC) abren nuevas posibilidades para servicios críticos, incluyendo vehículos autónomos y telemedicina, mientras que las comunicaciones masivas tipo máquina (mMTC) facilitan el despliegue de ecosistemas de Internet de las Cosas (IoT) con millones de dispositivos conectados en simultáneo.

Para garantizar el cumplimiento de estos casos de uso existen varios desafíos tecnológicos que sin duda pueden extrapolarse a lo que serán las redes de sexta generación, entre los cuales están: la gestión eficiente e inteligente de los recursos para el cumplimiento de la calidad de servicio (QoS), despliegue de *Network Slicing* en los distintos segmentos de la red, interoperabilidad entre sistemas heterogéneos, abordaje de problemas de ciberseguridad, entre otros [1].

La evolución de las redes móviles ha estado históricamente marcada por *hardware* de propósito específico, con funciones de red estrechamente vinculadas a los fabricantes, lo que limita tanto la flexibilidad de despliegue como la accesibilidad a los equipos. Sin embargo, tecnologías como la Radio Definida por Software (SDR) [2] han transformado este panorama, permitiendo que las implementaciones de redes inalámbricas, y en particular las redes móviles, sean más accesibles para la investigación y la educación.

Un SDR es un equipo de comunicaciones que sustituye funciones tradicionalmente implementadas en *hardware* por soluciones en *software*, operando sobre dispositivos de radiofrecuencia genéricos. Esto permite crear equipos de comunicación capaces de interactuar con diversos sistemas, como la clásica radio AM y FM, televisión digital [3], comunicaciones aeronáuticas [4] [5], e incluso aplicaciones específicas como estaciones base o equipos de usuario en redes móviles.

A pesar de estos avances, muchas de las soluciones disponibles para realizar

Capítulo 1. Introducción

despliegues realistas de redes móviles aún dependen de *hardware* costoso y *software* propietario, lo que representa una barrera significativa para equipos de investigación con recursos limitados. A modo de respuesta, han emergido alternativas de código abierto, como *srsRAN* [6] y *OpenAirInterface* [7].

Ambas ofrecen herramientas robustas para la implementación de redes 5G NR y su predecesora 4G LTE (*4th Generation Long-Term Evolution*), de forma local y autónoma. Este tipo de soluciones amplían el acceso a las tecnologías de redes móviles, fomentando la innovación y abriendo nuevas posibilidades para la investigación y la educación en esta área.

Por un lado *OpenAirInterface* (OAI) es gestionado por la *OpenAirInterface Software Alliance* (OSA), un grupo internacional de empresas, universidades e instituciones, incluyendo grandes actores de telecomunicaciones como Nokia, Vodafone y Ericsson. Este grupo colabora regularmente con el 3rd Generation Partnership Project (3GPP), la organización que produce las especificaciones técnicas que impulsan las diferentes generaciones de tecnologías de telecomunicaciones.

Por otro lado, el proyecto *srsRAN* es desarrollado por la compañía *Software Radio Systems* (SRS), una empresa privada que provee tanto el proyecto de código abierto *srsRAN* como una versión paga orientada a empresas junto con soporte continuo. Su primer proyecto en este campo fue una implementación análoga de un *testbed* para redes 4G, pasando luego a desarrollar el proyecto *srsRAN* 5G utilizado en este trabajo.

Contar con un banco de pruebas (**testbed**) abierto y configurable para redes de quinta generación resulta fundamental para universidades, centros de investigación y operadores que buscan experimentar con funcionalidades avanzadas, realizar pruebas de interoperabilidad o explorar nuevos enfoques como la arquitectura O-RAN [8]. Este tipo de entornos permite validar configuraciones, analizar el rendimiento bajo diferentes condiciones de red y, especialmente, reducir la barrera de entrada a tecnologías que tradicionalmente han estado reservadas a grandes proveedores. Además, al utilizar herramientas de código abierto, se promueve un modelo de desarrollo colaborativo, adaptable a los requerimientos específicos de cada proyecto.

Este proyecto tiene un enfoque integrador, abordando en simultáneo todos los componentes principales de una red 5G (UE, RAN y Core). A diferencia de proyectos especializados que profundizan exclusivamente en alguno de estos componentes, esta investigación adopta una perspectiva más generalista, orientada a proporcionar una visión holística del despliegue. Este abordaje resulta especialmente útil como base para proyectos futuros que decidan especializarse en componentes específicos, facilitando la continuidad y profundización en investigaciones posteriores.

Una de las ventajas más relevantes de los *testbeds* basados en soluciones libres como *srsRAN* es su capacidad de ejecutarse sobre plataformas de bajo costo y consumo, como mini-PCs o sistemas embebidos con soporte para virtualización. Esto habilita el diseño de redes portátiles y escalables, que pueden desplegarse en escenarios diversos, desde laboratorios académicos hasta entornos de producción reales, abarcando diferentes industrias. En la Facultad de Ingeniería de la Universidad de la República se tiene experiencia en el uso de *OpenAirInterface*, donde el

1.2. Definición del problema

proyecto “Implementación de una Maqueta de Pruebas y Desarrollo de una Red 5G Stand Alone Completa” [9] realizó un despliegue completo de un *testbed* 5G basado en este *software*. De allí que surge la necesidad de explorar *srsRAN*, con el fin de realizar una comparativa y sentar bases para futuras investigaciones.

Por otra parte, *srsRAN* ofrece una base práctica para realizar pruebas de aspectos clave como el acceso por radio, la separación funcional de la estación base, el monitoreo de tráfico o la interacción con distintos tipos de equipos de usuario, todos abordados en el presente documento.

1.2. Definición del problema

En este contexto, el grupo de trabajo se propone diseñar, desplegar y documentar un *testbed* funcional de red 5G que sea reproducible, adaptable y compatible con equipamiento comercial de amplia disponibilidad, contribuyendo tanto al plano de la investigación como a la formación en tecnologías de nueva generación. A continuación se presentan los objetivos generales y específicos.

1.2.1. Objetivos generales

El objetivo principal es implementar una maqueta o *testbed* 5G basada en *srsRAN Project*, tomando como referencia un trabajo anterior [9] que utilizó *OpenAirInterface* sobre el mismo *hardware* de base, para posteriormente analizar las ventajas de cada caso. En adición, se buscará estudiar el comportamiento de la red bajo distintos escenarios, explorar funcionalidades avanzadas y analizar la interacción con diferentes tipos de UE (*User Equipment*, equipo de usuario), incluyendo dispositivos embebidos y equipos de usuario comerciales.

Dado que *srsRAN* está optimizado para ejecutarse en plataformas de bajo porte y consumo, como *NUCs* o *Raspberry Pi*, el trabajo se orienta además a aprovechar esta eficiencia para lograr una implementación portable, capaz de operar fuera de entornos de laboratorio y adaptarse a condiciones de uso más versátiles.

El propósito es que esta implementación no esté sujeta a las restricciones de movilidad e infraestructura de otras que previamente fueron realizadas en PCs de escritorio, sino a lograr un *testbed* que facilite su uso en contextos que requieran mayor versatilidad en la ubicación y forma de uso de los equipos. Se evaluará la portabilidad obtenida desde 3 aspectos diferentes, buscando lograr el cumplimiento total de al menos uno de ellos en uno o más elementos de la red:

1. En el aspecto espacial, se buscó utilizar equipos cuyo tamaño y peso sean lo suficientemente reducidos como para que estos resulten fáciles de mover y colocar para el usuario en varios contextos.
2. En cuanto a la conectividad, se apuntó a lograr una independencia tal que la comunicación se realice de forma inalámbrica, facilitando el uso de los equipos al no requerir manejo de conexiones por cable.

Capítulo 1. Introducción

3. En el ámbito energético, la independencia deseada es tal que el *hardware* sea capaz de funcionar con una fuente de energía móvil que permita su traslado de forma sencilla.

Este enfoque no solo busca validar técnicamente la viabilidad de una red 5G portable sobre *hardware* accesible, sino también aportar herramientas prácticas para futuras implementaciones educativas o de investigación, promoviendo la flexibilidad, escalabilidad y autonomía en el desarrollo de soluciones de este tipo.

1.2.2. Objetivos específicos

Para ordenar el trabajo realizado, se desgranaron las metas generales expresadas en la sección anterior en los siguientes objetivos puntuales, de forma de poder medir el grado de cumplimiento de cada uno de ellos:

- 1 - Implementar el *testbed* o maqueta *end-to-end* con *srsRAN*, primero en un entorno virtual y luego sobre *hardware* físico.
- 2 - Una vez alcanzado un *testbed* funcional análogo al de experiencias previas, diagramar y ejecutar pruebas de rendimiento, con el objetivo de comparar esta implementación con otras, tomando como referencia los resultados obtenidos en trabajos anteriores como [9].
- 3 - Aplicar cambios en la arquitectura del *testbed* basados en *splits* funcionales para distribuir el procesamiento requerido a *hardware* portable de menos recursos.
- 4 - Utilizar los distintos tipos de UE disponibles para comparar el rendimiento de las variantes portables frente al *testbed* original implementado en la primera etapa, permitiendo así evaluar los compromisos y beneficios asociados al proceso de portabilización.

1.2.3. Alcance y limitaciones

El alcance de este proyecto concierne principalmente al uso e investigación del *software open-source* *srsRAN*, así como los equipos y programas utilizados para su operación y monitorización. Esto significa que se contempló el trabajo de configurar o modificar el código cuando fuera necesario en casos específicos, pero no se consideró dentro del alcance la implementación de programas más complejos. De la misma manera, el manejo del *hardware* utilizado en las diferentes etapas se restringió al uso, configuración e integración de equipos existentes, no incluyendo entonces posibles tareas de modificación y/o desarrollo de nuevo *hardware*.

En el caso de este proyecto, el equipamiento se limitó principalmente al ya disponible en la Facultad de Ingeniería de la Universidad de la República, con un presupuesto adicional destinado a la adquisición del *Kit Sixfab 5G*, que se describirá en detalle más adelante. Experiencias posteriores a esta se podrán beneficiar de

1.3. Antecedentes

adquirir *hardware* adicional compatible con el proyecto *srsRAN* para ampliar el universo de pruebas posibles.

Otra razón principal por la que este proyecto se limitó a los equipos disponibles fue con el fin de comparar el rendimiento de la implementación de *srsRAN* con los resultados observados en el proyecto antecesor “Implementación de una Maqueta de Pruebas y Desarrollo de una Red 5G Stand Alone Completa” [9], dado que se trata de un estudio cercano en el tiempo realizado en la misma institución, pero evaluando el principal competidor en el mercado, *OpenAirInterface*. De esta manera, esta restricción de *hardware* buscó aproximar lo más posible las condiciones en que se dieron las pruebas en ambos casos. Se deja fuera del alcance repetir las pruebas realizadas con *OpenAirInterface*, tomando como base comparativa los resultados documentados en el proyecto mencionado.

1.3. Antecedentes

Dado que se buscó evaluar el rendimiento de una de las principales soluciones *open-source* del mercado para emulación de redes 5G, es necesario observar las experiencias de trabajos anteriores sobre esta misma plataforma. De este modo, se pretende conocer el panorama actual e identificar en qué aspectos es posible plantear experimentaciones innovadoras, y cómo este trabajo puede servir de base para futuras investigaciones.

1.3.1. Testbeds 5G

Interesa en primer lugar hacer un relevamiento general de investigaciones similares a esta experiencia, evaluando los *testbeds* implementados en otros proyectos en los que se releva el rendimiento de una conexión básica. Esto permite no solo conocer capacidades y resultados ya observados con el *software* y *hardware* a utilizar, sino también profundizar en puntos en que no se haya ahondado previamente. Con este fin, se compiló en las tablas 1.1, 1.2 y 1.3 un resumen de la comparación entre tres investigaciones previas que se consideraron altamente representativas del trabajo observado en este campo. Estas representan tanto *testbeds* implementados usando el proyecto *srsRAN* como maquetas en otras plataformas, todas oficiando de guía en cuanto a la metodología a seguir.

Capítulo 1. Introducción

Tabla 1.1: “Implementación de una Maqueta de Pruebas y Desarrollo de una Red 5G Stand Alone Completa” [9].

Metodología	Comparativa inicial teórica de OAI y srsRAN. Implementación de <i>testbed</i> completo en OAI.
Software / Hardware	OpenAirInterface sobre PCs convencionales y equipos SDR USRP X310.
Contribuciones	Evaluación detallada del rendimiento de la red. Desarrollo de un planificador optimizado para la asignación de recursos en una única portadora.
Resultados	Buen rendimiento en escenarios controlados, aunque se detecta un uso intensivo de recursos de <i>hardware</i> y limitaciones al aumentar la demanda de tráfico.
Observaciones	La metodología utilizada sirvió de referencia directa para las pruebas sobre srsRAN realizadas en este proyecto.

Tabla 1.2: “Experimental Comparison of 5G SDR Platforms: srsRAN x OpenAirInterface” [10].

Metodología	Comparativa directa entre srsRAN y OAI con un único UE, mediante mediciones de <i>throughput</i> .
Software / Hardware	srsRAN y OpenAirInterface, mini-PCs Dell, SDR USRP B210 para gNB y UE simulado, y COTS UE.
Contribuciones	Validación de maquetas funcionales con UEs comerciales, y obtención de métricas de rendimiento en ambas plataformas.
Resultados	El UE comercial (COTS) mostró mejor rendimiento que los UEs simulados de ambas plataformas. Se reportaron diferencias de rendimiento y facilidad de configuración.
Observaciones	Las observaciones sobre el rendimiento del COTS UE fueron clave para la interpretación de resultados en este proyecto. También aportaron información útil sobre las curvas de aprendizaje de cada plataforma.

Los ejemplos presentados en las tablas 1.1, 1.2 y 1.3 representan una visión general del estado de la investigación en este campo, además de servir de referencia comparativa para la implementación del *testbed* y el diseño de pruebas. Como se mencionó anteriormente, se partió del proceso de pruebas realizadas en el proyecto MAQ5G [9] para realizar pruebas análogas exhaustivas en *srsRAN* y obtener una comparación detallada entre ambos proyectos. En el caso del segundo y tercer ejemplo utilizados, se profundizó en las pruebas realizadas para cada factor que incidiera en el rendimiento de la maqueta obtenida, pudiendo usarlos como base

1.3. Antecedentes

Tabla 1.3: “Deploying a Stable 5G SA Testbed using srsRAN and Open5GS: UE Integration and Troubleshooting towards Network Slicing” [11].

Metodología	Implementación de <i>testbed</i> simple en srsRAN con integración de varios UEs comerciales.
Software / Hardware	srsRAN sobre PC y SDRs, con múltiples UEs COTS disponibles en el mercado.
Contribuciones	Presenta una guía detallada del proceso de instalación y configuración de srsRAN. Releva además el nivel de compatibilidad 5G de varios modelos de UEs.
Resultados	Se logra un funcionamiento estable y se evidencia el apoyo de la comunidad de srsRAN en comparación con otras plataformas.
Observaciones	Las recomendaciones brindadas fueron de gran utilidad como complemento práctico a la documentación oficial de srsRAN, especialmente frente a los obstáculos más comunes, también experimentados en este trabajo.

para generar un conjunto de pruebas más detalladas.

Otro antecedente [12] relevante realizó una comparación directa entre las implementaciones *open-source* más utilizadas del núcleo 5G: *Open5GS*, *OpenAirInterface* y *Free5GC*. Al hacer una comparativa del rendimiento en el plano de control, y basándose en pruebas centradas en el tiempo de registro de los UEs, el uso de CPU, memoria y otros indicadores clave, el estudio permite identificar diferencias concretas en eficiencia y escalabilidad entre las soluciones de *Open5Gs* y *OpenAirInterface*. En este se observa un mejor rendimiento de *Open5GS* en numerosas pruebas, sobre todo en el consumo de CPU y memoria, siendo auspicioso para el objetivo de la portabilidad planteado.

1.3.2. Otros antecedentes

Por otra parte, se buscaron experiencias que hubiesen comprendido escenarios similares a los planteados en la segunda etapa del proyecto. Un ejemplo es el artículo de Amini y Rosenberg (2024) [13], en el que se analizan exhaustivamente un *testbed* 5G-SA basado en *srsRAN* y *OpenAirInterface* en escenarios de múltiples UEs. Este concluye que la capacidad de compartir recursos en escenarios con múltiples UEs depende significativamente del software RAN utilizado. Por ejemplo, OAI-RAN mantiene un rendimiento más estable y equilibrado en términos de reparto de ancho de banda entre UEs cuando aumenta la cantidad de dispositivos conectados.

Además, en el artículo “*Multi-UE 5G RAN Measurements: A Gamut of Architectural Options*” [14], se presenta un antecedente muy valioso para el análisis de maquetas basadas en *srsRAN*, especialmente en contextos donde múltiples UEs están conectados simultáneamente. Este trabajo encontró que incluso cuando

Capítulo 1. Introducción

los UEs utilizados son idénticos en *hardware* y configuración, el rendimiento entre ellos puede variar significativamente. Esta irregularidad es importante como antecedente porque alerta sobre posibles limitaciones o inestabilidades inherentes a la solución elegida en casos multiusuario.

Por otro lado, no se encontraron investigaciones que hubiesen realizado un *split* CU/DU con *srsRAN*. En el artículo [15] se realiza un análisis de los *splits* funcionales posibles. Allí se presenta al *Split 2* como una opción atractiva para implementaciones prácticas de redes desagregadas, balanceando entre los exigentes requerimientos del *fronthaul* del *Split 7* o *8* y la limitada flexibilidad del *Split 1*. Se menciona particularmente que requiere enlaces de *fronthaul* con latencias bajas (alrededor de 6 ms o menos) para mantener la sincronización de los planos de control y usuario. El hecho de la escasa exploración de la implementación de *splits* en *srsRAN* resulta una motivación para su investigación, abriendo puertas a implementaciones innovadoras.

Por último, se presenta como antecedente un trabajo [16] que resultó de interés al equipo, en el que se usa *srsRAN* como base para el despliegue de una solución diseñada para mejorar la inmunidad de las redes 5G frente a interferencias y mejorar las capacidades de localización, utilizando técnicas MIMO. Este estudio reafirma la relevancia de *srsRAN* como herramienta de investigación y desarrollo, y muestra cómo puede escalarse hacia pruebas más complejas.

1.4. Contribuciones

Antes de pasar al núcleo de la documentación, se anticipan al lector las contribuciones logradas en este trabajo de fin de carrera. En particular, este proyecto logró documentar un *testbed* integrador, contemplando simultáneamente todos los elementos principales de una red 5G (UE, RAN y Core). Se generó una comparativa global con la plataforma competidora *OpenAirInterface*, abordando aspectos generales del rendimiento e interacción entre los diferentes componentes. Estos resultados preliminares establecen una base robusta para futuros trabajos que busquen especializarse en aspectos específicos del despliegue o rendimiento de redes 5G. Dichos resultados fueron presentados en las “Jornadas Uruguayas de Ciencias de la Computación 2024” [17] organizadas por el Instituto de Computación de la Facultad de Ingeniería.

A su vez, se logró implementar una radiobase utilizando *srsRAN* con una arquitectura modular aplicando diferentes *splits* funcionales. Esta permitió disminuir el procesamiento requerido de cada componentes de la radiobase, además de permitir la utilización de equipos de menor porte que habilitan su uso de forma portable.

El despliegue de la maqueta basada puramente en *srsRAN* presentó ciertas limitaciones, derivadas del uso de *srsUE*, el cual es una adaptación de la versión LTE y no un UE 5G propio. Si bien algunas fueron sorteadas posteriormente mediante el uso de otros dispositivos de usuario, es importante tenerlas presentes. Estas limitaciones serán explicitadas en capítulos posteriores, particularmente en los análisis de desempeño, donde se discute su impacto en las pruebas realizadas y en la portabilidad de la solución.

1.4. Contribuciones

Esto llevó a la incorporación de nuevos equipos de usuario, cuyos desempeños fueron evaluados de forma práctica, incluyendo tanto dispositivos comerciales (COTS UE) como plataformas embebidas. Particularmente, se logró integrar de forma exitosa el dispositivo *Sixfab Hat 5G*, una placa reciente basada en un módem Qualcomm diseñada para agregar conectividad 5G a computadoras como la *Raspberry Pi*. Hasta la fecha, no se han encontrado estudios que documenten el uso de este equipo en entornos experimentales con redes 5G SA. Esta inclusión aporta evidencia concreta sobre la viabilidad de utilizar dispositivos de bajo consumo y bajo costo en pruebas de conectividad 5G, especialmente en contextos de Internet de las Cosas (IoT) y despliegues portables.

Se generó también un repositorio [18] en la plataforma de *Gitlab* de la Facultad de Ingeniería que contiene un *fork* del proyecto *srsRAN 5G*, con las configuraciones realizadas. En las diferentes ramas de este proyecto se encuentran las implementaciones en los diferentes entornos estudiados. Incluye también *scripts* desarrollados por el equipo que facilitan la ejecución de la maqueta al configurar una variedad de aspectos de uso del *hardware* requeridos por *srsRAN*. A su vez, se realizó un *fork* similar del proyecto *srsRAN 4G* con el fin de utilizar el módulo *srsUE* en esta maqueta, disponible también en la plataforma *Gitlab* [19].

Por último, se generó y adjuntó como apéndice a este documento un manual de uso del *testbed* implementado, en que se explican los comandos a utilizar para lograr un despliegue rápido de la maqueta funcional a partir del repositorio generado.

1.5. Estructura del documento

Este documento se estructura de la siguiente manera:

- Capítulo 1: Introducción.** Capítulo actual, dedicado a presentar el trabajo realizado y la motivación, junto con los objetivos y las contribuciones.
- Capítulo 2: Marco teórico y soluciones de hardware y software utilizadas.** Contexto teórico en que se desarrolla el proyecto, introduciendo los principales conceptos involucrados.
- Capítulo 3: Maqueta de red 5G.** Planteo de los procedimientos realizados en cada etapa para implementar y sintonizar las diferentes variantes de *testbed*.
- Capítulo 4: Dispositivos, pruebas y resultados preliminares.** Detalle de las pruebas diagramadas en diferentes casos para evaluar el rendimiento de las soluciones implementadas en el capítulo 3 y presentación de los resultados obtenidos.
- Capítulo 5: Análisis de resultados.** Análisis de los resultados obtenidos para cada experiencia en particular y en el contexto del resto del proyecto.
- Capítulo 6: Conclusiones y trabajo a futuro.** Conclusiones alcanzadas en base a los resultados observados y las hipótesis y objetivos planteados inicialmente.
- Apéndices:** Presentación de experiencias adicionales de trabajo adyacente al proyecto que escapa al alcance de la documentación principal.

Capítulo 2

Marco teórico y soluciones de hardware y software utilizadas

Este capítulo tiene como objetivo introducir al lector en la tecnología de redes celulares de quinta generación (5G), abordando sus aspectos más relevantes para una mejor comprensión del presente trabajo. Además, se presentarán las principales características de las herramientas fundamentales utilizadas en el desarrollo. Dado que el proyecto implicó el uso de una amplia variedad de *hardware* y *software*, en este capítulo se profundizará específicamente en conceptos clave para el entendimiento del proyecto, y se complementa con referencias bibliográficas en cada sección, permitiendo así al lector profundizar en los temas que le sean de interés.

2.1. Redes móviles 5G

Las redes móviles 5G representan una evolución significativa con respecto a sus predecesoras, no solo en términos de rendimiento, sino también en la diversidad de aplicaciones que posibilitan. Para comprender su impacto y su relevancia, es fundamental analizar primero sus principales características y casos de uso. Por otra parte, es de suma importancia explorar la arquitectura de las redes móviles y, en particular, de una de quinta generación, detallando sus componentes esenciales.

2.1.1. Aspectos generales y casos de uso

La evolución de las redes celulares ha estado marcada por la búsqueda de mayor capacidad, velocidad y eficiencia espectral para adaptarse a las crecientes demandas de conectividad (ver Figura 2.1). Desde la primera generación de redes móviles (1G), que solo permitía comunicaciones de voz analógicas, cada avance ha traído mejoras clave en términos de rendimiento y nuevas aplicaciones. Un ejemplo es 4G LTE, desarrollada a partir de la *Release 8* de 3GPP [20], que representó un salto significativo al introducir una arquitectura simplificada basada en IP, el uso de Orthogonal Frequency-Division Multiplexing (OFDM) [21] en el enlace descendente y MIMO (*Multiple Input Multiple Output*) para mejorar la eficiencia

Capítulo 2. Marco teórico y soluciones de hardware y software utilizadas

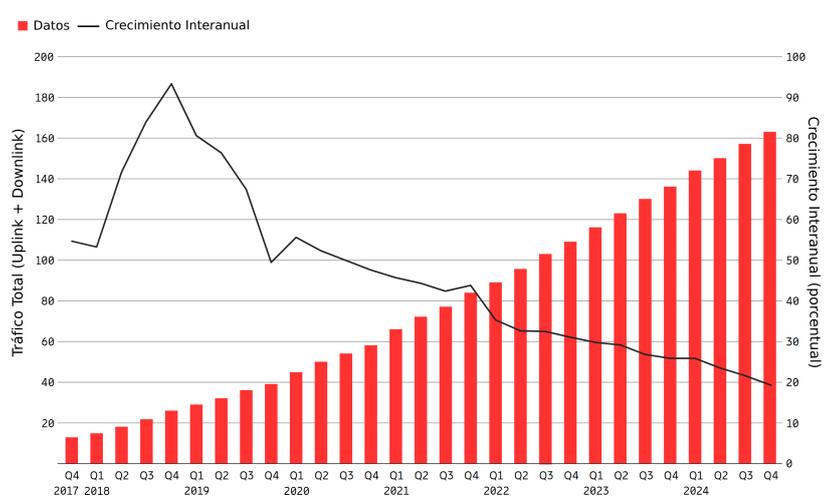


Figura 2.1: Tráfico de datos de la red móvil global de Ericsson y crecimiento interanual (EB por mes). Imagen adaptada de [22].

espectral. Estas innovaciones permitieron alcanzar velocidades de hasta 1 Gbps, abriendo paso a aplicaciones de alta demanda de datos, como el streaming en HD, las videoconferencias y los servicios en la nube.

A pesar de los avances de LTE y LTE-Advanced, las demandas de conectividad siguieron en aumento, particularmente con la proliferación del Internet de las Cosas (IoT), la búsqueda de baja latencia en aplicaciones críticas y el tráfico dispar de múltiples dispositivos. Ante estas innovaciones, surgió la necesidad de una nueva generación de redes móviles, lo que motivó el desarrollo del estándar 5G NR (New Radio). Diseñado como una evolución directa de LTE, 5G no solo mejora la capacidad y el *throughput*, sino que redefine la manera en que los recursos de red son gestionados y optimizados para soportar un ecosistema de servicios mucho más amplio y exigente. Para ello, fue fundamental el desarrollo de un nuevo núcleo de red *5G Core* [23], el cual mediante un nuevo enfoque tuviese la capacidad de manejar estos exigentes requerimientos.

Entre las mejoras técnicas de 5G se destaca la posibilidad de operar en una amplia gama de anchos de banda, desde 5 MHz hasta 100 MHz en el espectro por debajo de 6 GHz (FR1), y hasta 400 MHz en el rango de frecuencias más altas (FR2). Además, se incorporan modulaciones más eficientes como 256-QAM, y el uso de técnicas como *beamforming*, que permiten focalizar la señal hacia el usuario para mejorar la cobertura y la eficiencia espectral.

Por otra parte, generaciones anteriores de redes celulares no fueron diseñadas para manejar de forma eficiente tráfico heterogéneo, lo que obligaba a desplegar redes específicas para diferentes tipos de usuarios y aplicaciones. Una de las grandes innovaciones de 5G es su capacidad para integrar, dentro de una misma infraestructura, dispositivos con necesidades de servicio muy distintas, optimizando el uso de los recursos de red sin requerir soluciones dedicadas para cada caso. En este sentido, el estándar IMT-2020 de la *International Telecommunication Union (ITU)* establece tres pilares fundamentales [24]:

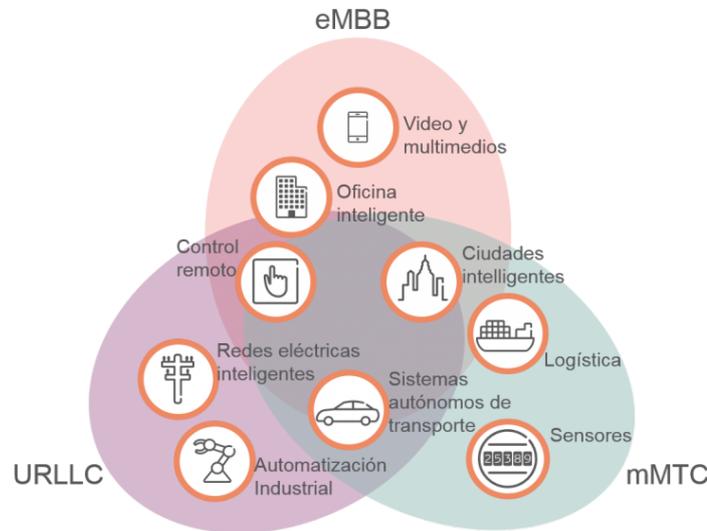


Figura 2.2: Áreas de aplicación de redes 5G y ejemplos de servicios. Imagen obtenida de [25].

- **Banda ancha móvil mejorada (eMBB):** Se busca ofrecer velocidades de datos significativamente más altas a las alcanzadas en generaciones anteriores, mejorando la velocidad de descarga de contenido de gran tamaño y tráfico concurrente, e incluso posibilitando aplicaciones de *streaming* de muy alta calidad.
- **Comunicaciones ultra confiables y de baja latencia (URLLC):** Se enfoca en aplicaciones críticas que requieren baja latencia y alta confiabilidad, como telemedicina, automatización industrial (Industria 4.0) y vehículos autónomos.
- **Comunicaciones masivas de tipo máquina (mMTC):** Se pretende abarcar la conexión simultánea de millones de dispositivos IoT, optimizando el uso del espectro para sensores, dispositivos inteligentes y ciudades conectadas.

Dentro de estos pilares, las redes 5G han habilitado diversos casos de uso que abarcan desde conectividad avanzada para usuarios hasta aplicaciones industriales y de investigación. Un ejemplo es la automatización industrial y las fábricas inteligentes, donde las capacidades de URLLC permiten la operación de maquinaria autónoma y la ejecución de procesos sincronizados en tiempo real. La posibilidad de implementar diferentes perfiles de QoS (*Quality of Service*) y el uso de técnicas como *Network Slicing* permiten la optimización de los procesos de producción y reducen la latencia en sistemas críticos [26]. En el sector del transporte, las redes V2X (*Vehicle-to-Everything*) facilitan la comunicación en milisegundos entre vehículos e infraestructura, mejorando la seguridad vial y la eficiencia en la gestión del tráfico. Son estos algunos de los muchos casos de uso posibles (ver Figura 2.2).

La flexibilidad de 5G se debe en gran parte a su arquitectura basada en la virtualización de funciones de red (NFV) y redes definidas por software (SDN),

Capítulo 2. Marco teórico y soluciones de hardware y software utilizadas

lo que permite una gestión dinámica y adaptable a cada aplicación. Además, su evolución hacia Open RAN (O-RAN) [8] ha impulsado la interdisciplina con redes ópticas, favoreciendo la integración con infraestructuras de *fronthaul* y *backhaul* basadas en fibra óptica. Este enfoque no solo optimiza la transmisión de datos en redes móviles, sino que también permite mejorar la eficiencia espectral y la escalabilidad de las implementaciones futuras.

Además del impacto técnico y funcional, el despliegue de redes 5G también plantea implicancias de orden social y económico. Desde el punto de vista productivo, la posibilidad de contar con conectividad robusta y de baja latencia habilita modelos de negocio basados en automatización, monitoreo y toma de decisiones basada en datos en tiempo real. A nivel social, el acceso a redes de alta capacidad puede reducir la brecha digital en zonas rurales mediante soluciones como el acceso inalámbrico fijo (FWA) [27], facilitando el acceso a servicios educativos, de salud y administrativos en línea. No obstante, estos beneficios dependerán en gran medida de las políticas de despliegue, los marcos regulatorios y la inversión en infraestructura, que determinarán cuán equitativo será el acceso a esta tecnología.

2.1.2. Arquitectura de red 5G

Una red móvil se compone de varios elementos que trabajan de forma coordinada para brindar servicios al usuario. Al igual que en generaciones anteriores, en una red de quinta generación es posible distinguir tres bloques principales: el equipo de usuario (UE), la red de acceso (*Next Generation - Random Access Network*, NG-RAN) y el núcleo de red de quinta generación (*Core 5G*, 5GC), interconectados tal como se puede ver en la Figura 2.3.

Esta división lógica —entre el terminal que accede a la red, el nodo de acceso y el sistema que gestiona el tráfico y los servicios— ha estado presente desde las primeras generaciones de redes móviles. Ya en 2G (GSM) se definían componentes equivalentes, como el MS (*Mobile Station*), el BTS (*Base Transceiver Station*) y el MSC (*Mobile Switching Center*). A lo largo de las sucesivas generaciones, esta estructura se ha mantenido, aunque las funciones de cada una han cambiado, volviéndose cada vez más complejas y adaptándose a nuevos requerimientos. Esta separación permite una mayor modularidad, facilita la escalabilidad de la red y permite evolucionar cada componente de forma independiente, lo cual es especial-

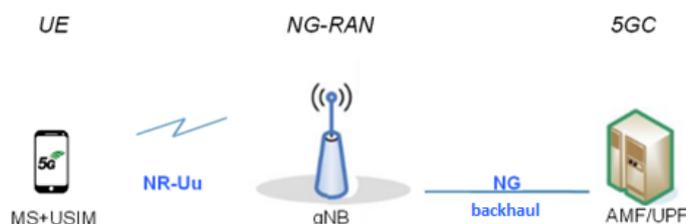


Figura 2.3: Visión general de arquitectura de red. Presenta componentes, interfaces y tipo de conexión. Obtenida de [28].

mente relevante en 5G, donde cada uno de estos bloques puede ser desplegado en forma virtualizada o distribuida.

Núcleo de Red

En la arquitectura 5G, el núcleo de red o *Core Network* es el nodo central en la gestión del acceso, la movilidad y el tráfico de datos entre el usuario y las redes externas. A diferencia de generaciones anteriores, en donde funcionaba como un bloque monolítico, el núcleo de quinta generación fue diseñado desde las bases con una arquitectura basada en servicios (*Service-Based Architecture*, SBA) [29], lo que permite una organización más flexible, modular y escalable.

Este se subdivide en funciones específicas, cada una cumpliendo un rol determinado en los planos de control y usuario. Esta separación se enmarca en otra de mayor abstracción, conocida como *Control and User Plane Separation* (CUPS). Si bien ya había sido introducida en las últimas versiones del núcleo LTE, en 5G se implementa de forma más explícita y flexible. CUPS permite que las funciones encargadas del plano de usuario (como la UPF) se desplieguen de manera independiente a las del plano de control (como AMF y SMF). Entre las funciones más relevantes para este proyecto se encuentran:

- **AMF (Access and Mobility Management Function):** gestiona el acceso del UE a la red, el proceso de registro, la autenticación y la movilidad entre celdas. Actúa como punto central de control del plano de acceso.
- **SMF (Session Management Function):** responsable del establecimiento, modificación y liberación de las sesiones de datos. También administra la asignación de direcciones IP y se comunica con la UPF para definir rutas de tráfico.
- **UPF (User Plane Function):** se ubica en el plano de usuario y se encarga del reenvío del tráfico de datos del usuario entre la RAN y las redes externas. Soporta funciones como control de tráfico, filtrado de paquetes, mediciones y aplicación de políticas.
- **UDM (Unified Data Management):** administra la información del abonado, incluyendo datos de autenticación y políticas de servicio, y es equivalente funcional al HSS en redes LTE.
- **PCF (Policy Control Function):** define y aplica políticas de control de acceso, priorización de tráfico y asignación de recursos. Se comunica con el SMF y otros nodos para autenticar a los usuarios, aplicar decisiones de QoS y cobro en las sesiones establecidas, e incluso aplicar políticas adicionales como *roaming*.
- **NSSF (Network Slice Selection Function):** determina a qué *slice* de red debe conectarse el dispositivo, en función del tipo de servicio y políticas definidas.

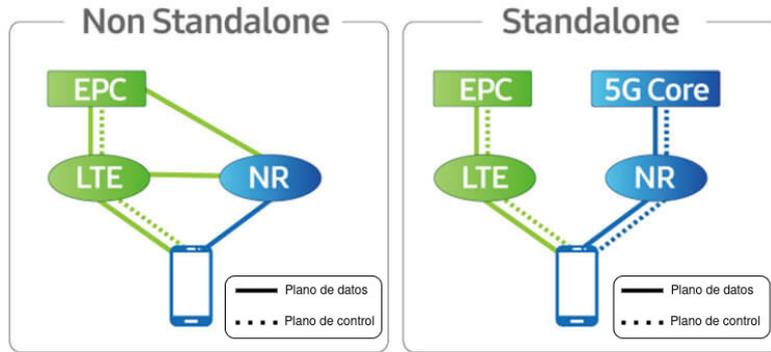


Figura 2.4: Implementación SA vs NSA. Imagen adaptada de [30].

Esta arquitectura orientada a servicios no solo mejora la escalabilidad y eficiencia, sino que también permite que múltiples instancias de cada función coexistan en paralelo. Además, es posible colocar nodos de funciones específicas más cerca del borde de la red (*edge computing*) para reducir la latencia, como es el caso del UPF en implementaciones URLLC. A su vez, la separación CUPS permite, por ejemplo, que el plano de usuario se distribuya geográficamente para mejorar la eficiencia del enrutamiento y reducir el uso del *backhaul*, mientras que el plano de control puede permanecer centralizado, optimizando el uso de recursos y la gestión de políticas. En este sentido, el núcleo 5G habilita conceptos como *network slicing*, redes privadas y servicios personalizados.

Una característica clave del despliegue de redes 5G es la posibilidad de implementarlas en dos modos distintos: *Non-Standalone* (NSA) y *Standalone* (SA). En el modo NSA, la red 5G se apoya sobre la infraestructura existente de 4G LTE, utilizando el *Evolved Packet Core* (EPC) de LTE como núcleo, y permitiendo que el control de la conexión siga gestionado por una eNodeB LTE. En este esquema, el gNB 5G actúa como nodo secundario (gNB-NR) y se encarga exclusivamente del tráfico de datos de alta velocidad, sin gestionar el plano de control (ver Figura 2.4). Esto facilita un despliegue inicial más rápido y económico, aunque con limitaciones en cuanto a latencia y flexibilidad de servicios avanzados.

Por el contrario, en la arquitectura SA, el gNB 5G asume completamente la gestión de la conexión, apoyado en un núcleo 5G (5GC) nativo y con arquitectura SBA. Esta modalidad permite aprovechar completamente las capacidades de la tecnología 5G, como la ultra baja latencia, la separación completa de planos de control y usuario (CUPS), y la implementación de *network slicing* [31]. El modo SA representa la arquitectura objetivo a largo plazo, y es la base para habilitar aplicaciones críticas como URLLC o redes privadas industriales.

A lo largo de este proyecto se trabajó con una arquitectura *Stand Alone*, operando tanto con un núcleo de red como con una radiobase de quinta generación, intentando así explorar las capacidades de esta nueva tecnología. De todos modos, dado que el foco no era el estudio del núcleo en específico, este se desplegó utilizando una solución *open-source* como es *Open5GS*, que provee un *core* funcional

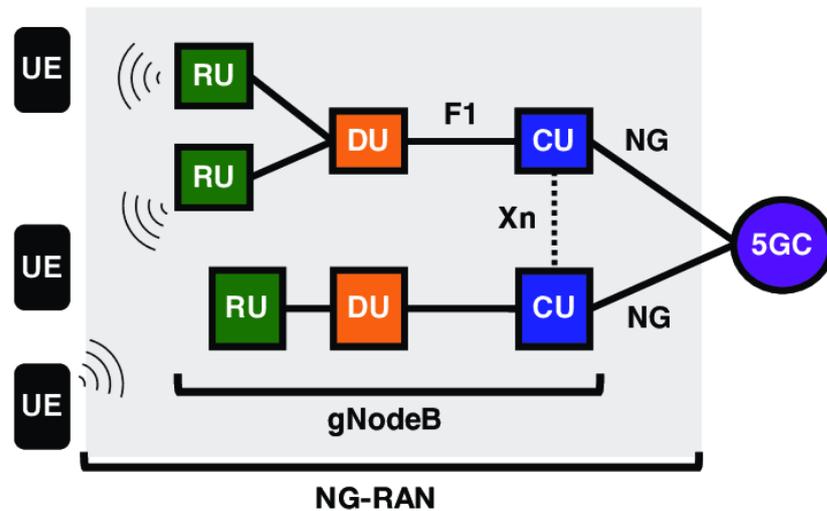


Figura 2.5: Secciones de una radiobase 5G (gNB). En gris se enmarcan las diferentes componentes. Imagen obtenida de [36].

listo para el despliegue. En su página oficial [32] se presentan en detalle sus componentes y funcionalidades.

Radiobase

En redes móviles, la radiobase es un componente esencial que establece y gestiona la comunicación inalámbrica entre los dispositivos de usuario y el núcleo de red. Su denominación ha cambiado con el avance de las generaciones, así como su rol y funciones. En particular, en 5G estas estaciones se denominan gNodeB (gNB) y representan una evolución significativa respecto a las eNodeB de las redes 4G. La gNB no solo maneja las funciones tradicionales de transmisión y recepción de señales de radio, sino que también incorpora tecnologías clave como codificación y modulación adaptativa (AMC) [33], múltiples configuraciones de subportadora (*Subcarrier Spacing*) en el acceso OFDM [34], soporte para *beamforming* digital y analógico, y la operación simultánea sobre múltiples portadoras (*Carrier Aggregation*) [35], lo que permite cumplir con los exigentes requisitos de latencia, eficiencia espectral y densidad de dispositivos propios de las redes 5G.

La evolución hacia redes de acceso por radio más flexibles y eficientes ha llevado a la introducción de arquitecturas desagregadas en 5G, donde la radiobase tradicional se divide en tres: una Unidad Central (CU), una Unidad Distribuida (DU), que juntas componen lo que tradicionalmente se conoce como *Base Band Unit* (BBU), y una Unidad de Radio (RU). En la Figura 2.5 es posible observar dichas secciones y como se interconectan entre sí. Las separaciones funcionales se conocen como *splits* y pueden ser implementadas en diferentes formas según las necesidades del operador o de servicio.

El *3rd Generation Partnership Project* (3GPP), en su informe técnico TR 38.801, “*Study on new radio access technology: Radio access architecture and in-*

Capítulo 2. Marco teórico y soluciones de hardware y software utilizadas

terfaces” [37], analiza diferentes opciones de separación funcional entre estas componentes. Dichas opciones, numeradas del 1 al 8, varían según el punto de división en la pila de protocolos como se puede ver en la Figura 2.6. Cada opción ofrece un equilibrio distinto entre los beneficios de centralización y los requisitos de la red de transporte. Al dividir las funciones de la BBU, los operadores pueden optimizar la asignación de recursos, mejorar la coordinación entre celdas y facilitar la implementación de tecnologías avanzadas como MIMO masivo y *Coordinated Multi-point* (CoMP).

En este proyecto se trabajó con dos tipos de splits funcionales. El Split 8 estuvo presente desde el comienzo al instanciar la maqueta utilizando equipos *Software Defined Radio* (ver Sección 2.3), ya que es la separación más baja en la arquitectura de una estación base, dividiendo la Unidad de Radio (RU) del resto. En esta configuración, la Unidad Distribuida (DU) se encarga de la capa física, mientras que la Unidad Central (CU) gestiona las capas superiores del protocolo. La RU solo realiza la conversión de señales digitales a analógicas y viceversa, además de la transmisión y recepción de radiofrecuencia (RF). Esta separación exige un enlace de transporte de alta capacidad y baja latencia, ya que el tráfico entre la RU y la DU es intenso y altamente sensible al retardo.

Además, en una siguiente etapa la comunicación entre la CU y la DU se implementó con un *split 2*, donde la DU maneja la capa RLC (*Radio Link Control*), MAC (*Media Access Control*) y la capa física, mientras que la CU administra las capas superiores, incluyendo PDCP (*Packet Data Convergence Protocol*), SDAP (*Service Data Adaptation Protocol*) y RRC (*Radio Resource Control*). Esta división permite mayor flexibilidad en la gestión de la red, facilitando una arquitectura menos centralizada.

La separación funcional entre CU, DU y RU no solo introduce modularidad en el diseño de las radiobases, sino que habilita nuevas posibilidades operativas en redes de acceso. Al permitir que las funciones se distribuyan geográficamente, esta arquitectura hace posible desplegar componentes del gNB en entornos virtualizados, como centros de datos locales o nubes públicas y privadas. En particular, la DU puede instalarse próxima al borde de la red para reducir la latencia en aplicaciones críticas, mientras que la CU puede centralizarse y gestionarse como parte de un entorno en la nube escalable. Este enfoque descentralizado permite implementar estrategias de *edge computing*, mejorar la eficiencia del uso del espectro, facilitar el mantenimiento y abrir la puerta a esquemas *multi-vendor* interoperables, todo en línea con los principios de arquitecturas abiertas como O-RAN.

Equipo de Usuario

El componente final en la arquitectura de una red móvil es la terminal o equipo de usuario (*User Equipment, UE*), que representa el punto de entrada del usuario final a los servicios provistos por la red. Este puede tomar diferentes formas, desde los hoy omnipresentes dispositivos móviles personales, como teléfonos inteligentes, hasta PCs o terminales interactivas de otros equipos que hagan uso de la conectividad provista. El avance de las redes 5G en el desarrollo de la banda ancha móvil se debe en gran parte a este tipo de dispositivos y su adopción masiva a nivel global.

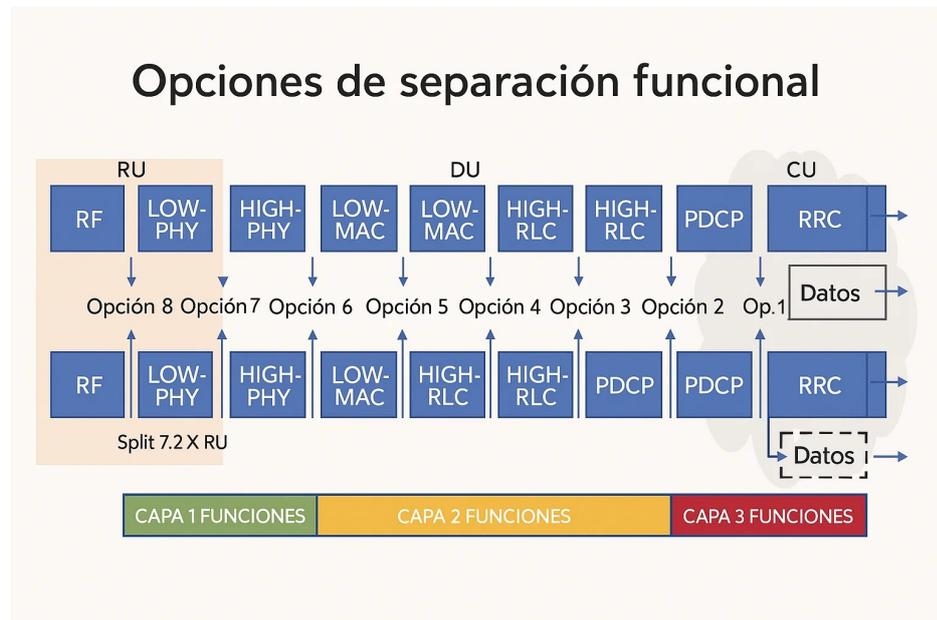


Figura 2.6: Posibles *splits* en una arquitectura 5G NR. Imágen obtenida de 5G Technology World.

A su vez, a esto se suma la incorporación de mMTC como pilar del desarrollo de nuevos servicios, lo que implica un conjunto nuevo de posibles UEs, con requerimientos sumamente variados. Estos incluyen sensores de baja potencia, cámaras de seguridad, electrodomésticos conectados, dispositivos médicos portátiles y medidores inteligentes de energía, como los que ya se han desplegado en Uruguay [38]. Este ecosistema diverso requiere que la red 5G sea capaz de adaptarse a distintos perfiles de tráfico, lo que representa uno de sus principales desafíos.

Desde el punto de vista técnico, la conexión de un UE a la red móvil comienza con su identificación y autenticación. Este proceso es posible gracias al uso de una tarjeta SIM (*Subscriber Identity Module*), que contiene la información necesaria para validar la identidad del abonado frente al núcleo de red. Esta tarjeta, que puede ser física (como ha venido siendo hasta ahora en los equipos celulares) o su reciente versión virtual (eSIM o integrada en firmware en algunos dispositivos embebidos), permite que el dispositivo se registre en la red, obtenga acceso a servicios e incluso sea facturado correctamente.

Esta tarjeta SIM contiene varios elementos, siendo el principal el IMSI (*International Mobile Subscriber Identity*), un número identificador único a nivel mundial para cada usuario. Este identificador consta típicamente de 15 dígitos, de los cuales los primeros 3 indican el código de país (*Mobile Country Code*, MCC) y los siguientes 3 indican un código de red móvil (*Mobile Network Code*, MNC), que en conjunto determinan el PLMN (*Public Land Mobile Network*) al que pertenece el abonado. Este parámetro es fundamental, ya que le indica al núcleo si el equipo tiene permiso para acceder a la red actual o debe considerarse como externo y por lo tanto aplicar políticas de *roaming*. Junto al IMSI, la SIM también contiene

Capítulo 2. Marco teórico y soluciones de hardware y software utilizadas

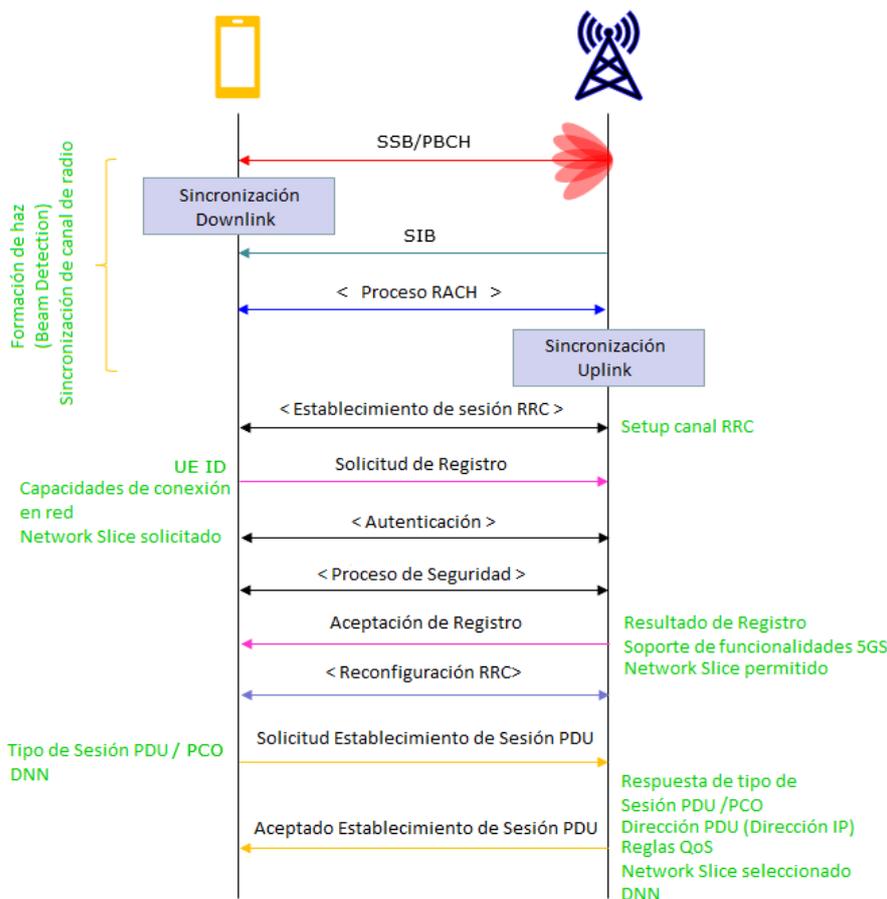


Figura 2.7: Proceso de registro de un UE en la red 5G, adaptado de [39].

parámetros necesarios para ejecutar algoritmos de autenticación y generación de claves criptográficas utilizadas en la comunicación segura. Estos se basan en el uso de un parámetro compuesto OPc, que se logra mediante la unión de un *Operator Code* (OP) asignado al operador de red que provee el servicio, y una clave *Key* específica al usuario que lo identifica de manera única. De esta forma, un OPc válido indica la identidad de un equipo a la vez que lo valida como usuario de la red. Todos estos parámetros se cotejan con los identificadores correspondientes al usuario cargados en la UDM del núcleo de red, lo que permite autenticar su identidad.

Además, al configurar un equipo de usuario para conectarlo a una red, se debe especificar el APN (*Access Point Name*). Este identifica el punto de acceso a la red de datos del operador e indica al núcleo el tipo de servicio solicitado por el usuario (por ejemplo, acceso a redes públicas o privadas), y en muchos casos debe ser configurado manualmente en el dispositivo o en los archivos de configuración.

En redes 5G, la autenticación puede seguir distintos esquemas, siendo el más moderno el 5G-AKMA (*Authentication and Key Management for Applications*) [40]. Este procedimiento implica un intercambio de mensajes cifrados entre el UE y el núcleo, donde se verifican las credenciales y se establecen claves de sesión

seguras para el resto de la comunicación.

En la Figura 2.7 se presenta de forma gráfica el proceso de autenticación e identificación en el contexto de cada capa. La primera etapa es la del descubrimiento entre antenas de cada dispositivo, iniciada por la radiobase en el canal de transmisión física (PBCH). Luego de solicitar la conexión en capa física mediante el proceso de *Radio Access Channel* (RACH), el cual se utiliza desde 3G para permitir al UE establecer una comunicación con la radiobase, se determinan los recursos físicos a destinar a esta conexión mediante el protocolo *Radio Resource Control* (RRC). Aquí es donde se aplican funciones desarrolladas en 5G para “personalizar” esta conexión a los requerimientos del UE, como *network slicing*. Al formar así una conexión estable a nivel de capa física, se puede realizar el intercambio de credenciales en capas superiores mediante los protocolos detallados anteriormente. Una vez autenticado y con una conexión estable, comienza la transmisión de datos para uso del usuario.

En el contexto del presente trabajo, estos procesos representaron un punto clave de la conexión de los diferentes equipos de usuario a la maqueta. El teléfono móvil comercial (COTS UE) accedió a la red a través de una SIM física programable, lo que permitió su integración inmediata al núcleo. En contraste, el dispositivo embebido *Sixfab Hat 5G*, basado en un módem Qualcomm, también requería la configuración de una SIM física, pero con una interfaz de consola más accesible para propósitos de testeo. Por último, el srsUE —una implementación puramente en software del UE— simula este proceso de autenticación mediante parámetros configurables en archivos, donde se definen valores como el IMSI, la Key y el OPc.

2.1.3. Transmisión y acceso al medio

La transmisión de datos en 5G NR se basa en el uso de la modulación OFDM (*Orthogonal Frequency Division Multiplexing*), una técnica que permite dividir el canal disponible en múltiples subportadoras ortogonales y paralelas. Esta característica posibilita una alta eficiencia espectral y una notable robustez frente a los efectos del canal, como el desvanecimiento selectivo en frecuencia [21]. En el enlace descendente (DL), 5G NR utiliza OFDM con CP (*Cyclic Prefix*), mientras que en el enlace ascendente (UL) puede utilizar tanto OFDM con CP como DFT-s-OFDM (*Discrete Fourier Transform spread OFDM*), una variante que reduce el PAPR (*Peak-to-Average Power Ratio*) y es especialmente útil en dispositivos con restricciones energéticas como los UEs móviles [41].

En 5G NR, la asignación de recursos físicos para la transmisión de datos se basa en unidades conocidas como PRB (*Physical Resource Blocks*). Este se define como un bloque de 12 subportadoras contiguas en el dominio de la frecuencia (ver Figura 2.8). Estos bloques representan la unidad mínima que puede ser asignada a un usuario en el canal compartido, tanto en enlace ascendente como descendente. La cantidad total de PRBs disponibles dentro de un sistema depende del ancho de banda operativo.

Una de las principales innovaciones que introduce 5G NR respecto a generaciones anteriores es la flexibilidad en la configuración del esquema OFDM, conocida como numerología. Este término hace referencia al espaciamiento entre subporta-

Capítulo 2. Marco teórico y soluciones de hardware y software utilizadas

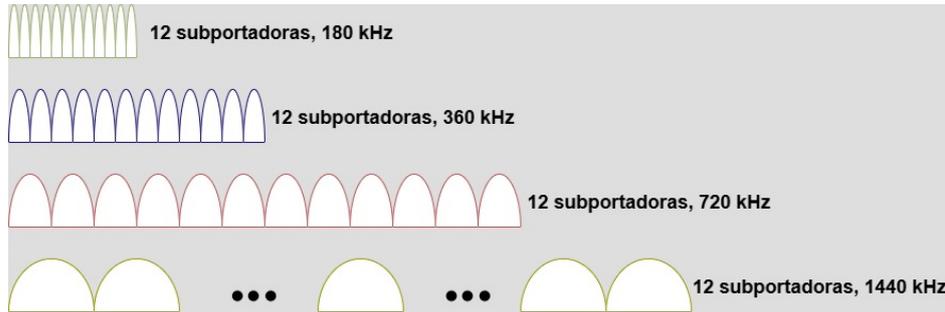


Figura 2.8: Variación del ancho de banda para diferentes numerologías.

doras (*Subcarrier Spacing*, SCS) y la duración de símbolo OFDM. El SCS puede tomar valores de $2^n \times 15$ kHz, siendo $n = \{0, 1, 2, 3, 4\}$. Esto habilita configuraciones de 15, 30, 60, 120 y 240 kHz, lo que permite adaptar la red a diferentes escenarios de operación y requisitos de servicio. Tomando en cuenta que se usan 12 subportadoras, esto resulta en anchos de banda de hasta 400 MHz. En la Figura 2.8 se puede observar de forma gráfica el impacto en el ancho de banda al variar estos valores.

En el aspecto temporal, la organización se basa en una estructura jerárquica que permite administrar recursos y sincronizar la comunicación entre UE y el resto de la red. La unidad de tiempo es llamada *frame*, y tiene una duración fija de 10 ms. Esta se subdivide en *subframes* de 1 ms cada uno, los cuales están compuestos por una cantidad variable de *slots*, dependiendo de la numerología utilizada. Por ejemplo, si se está utilizando 60 kHz de SCS, cada *subframe* es equivalente a cuatro *slots*. En el caso de trabajar con un SCS de 120 kHz, se tienen 8 *slots* por *subframe*, tal como se presenta en la Figura 2.9. Este nivel de granularidad es fundamental para adaptar la red a distintos tipos de servicio.

La asignación de recursos en 5G NR se coordina mediante canales físicos específicos que transportan la información de control necesaria para la transmisión de datos. En enlace de *downlink*, el *Physical Downlink Control Channel* (PDCCH) es el responsable de transmitir las *Downlink Control Information* (DCI), que indican al UE qué bloques de recursos han sido asignados para recibir datos, qué esquema de modulación utilizar (por ejemplo QPSK, 16-QAM, 64-QAM o 256-QAM), y cuál es la tasa de codificación correspondiente. Esta información permite que el equipo de usuario decodifique correctamente el canal de datos, denominado *Physical Downlink Shared Channel* (PDSCH). El PDCCH se monitorea al menos una vez por slot, garantizando que el UE reciba instrucciones actualizadas en tiempo real para adaptarse a las condiciones cambiantes del canal [42].

En el caso del *uplink*, el proceso comienza con el equipo de usuario informando a la red su intención de transmitir datos. Esto se realiza a través del envío de información de control denominada *Uplink Control Information* (UCI), que puede incluir indicadores como la calidad del canal, solicitudes de retransmisión o confirmaciones *Hybrid Automatic Repeat Request* (HARQ). Una vez recibida esta notificación, la red responde mediante el PDCCH, enviando una DCI que indica al UE en qué bloques de recursos puede transmitir, qué esquema de modulación utilizar y otros parámetros de transmisión. La información útil es entonces enviada a

2.2. Software utilizado

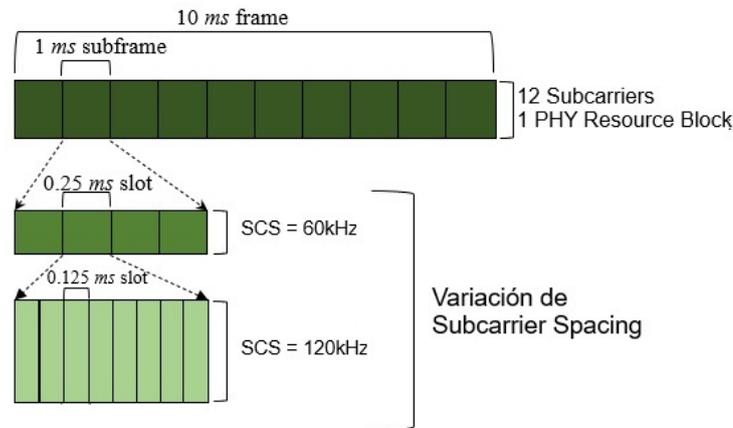


Figura 2.9: *Resource Blocks* e impacto de *Subcarrier Spacing* en tamaño de *subframes*.

través del *Physical Uplink Shared Channel* (PUSCH).

Durante el desarrollo del proyecto se realizaron pruebas variando el ancho de banda de operación y la numerología utilizada, lo cual implicó modificar el número de *Physical Resource Blocks* (PRB) disponibles y adaptar la estructura temporal de los cuadros de transmisión. Esto permitió explorar de forma experimental su impacto en la calidad de la conexión, la estabilidad del enlace y los bitrates alcanzables, así como la capacidad de cómputo necesaria. Además, el entendimiento del proceso de conexión del equipo de usuario fue clave para el *debugging* de diferentes problemas presentados al implementar el srsUE y el análisis de sus *logs*.

2.2. Software utilizado

A lo largo de este proyecto se utilizaron diferentes herramientas de software para implementar la maqueta de red 5G y evaluar su funcionamiento, en su mayor parte de proyectos *open-source*. A continuación se presentan los principales recursos utilizados para este proceso.

2.2.1. El proyecto srsRAN

srsRAN Project es una implementación de código abierto de una plataforma de emulación para un sistema de acceso de red móvil basado en las tecnologías 4G y 5G del 3GPP, publicado por la compañía *Software Radio Systems*. Se trata de una empresa de origen europeo que se ha dedicado al desarrollo y venta de licencias de uso para soluciones orientadas a redes de acceso móviles desde 2014 [43]. Opera proveyendo de forma abierta el código base de sus productos en el marco de licencias *open-source*, a la vez que obtiene sus ganancias mediante la venta de soluciones corporativas de redes de acceso personalizadas para otras empresas contratantes.

Capítulo 2. Marco teórico y soluciones de hardware y software utilizadas

El producto *srsRAN* es un proyecto en desarrollo constante, tanto por los avances incorporados por la propia empresa para mejorar la funcionalidad de sus productos, como por la colaboración de la comunidad *open-source*. Su implementación se realizó principalmente en el lenguaje C++, con un uso mínimo de librerías abiertas de terceros. En la Figura 2.10 se puede observar parte de su *roadmap*, específicamente lo relacionado a 5G.

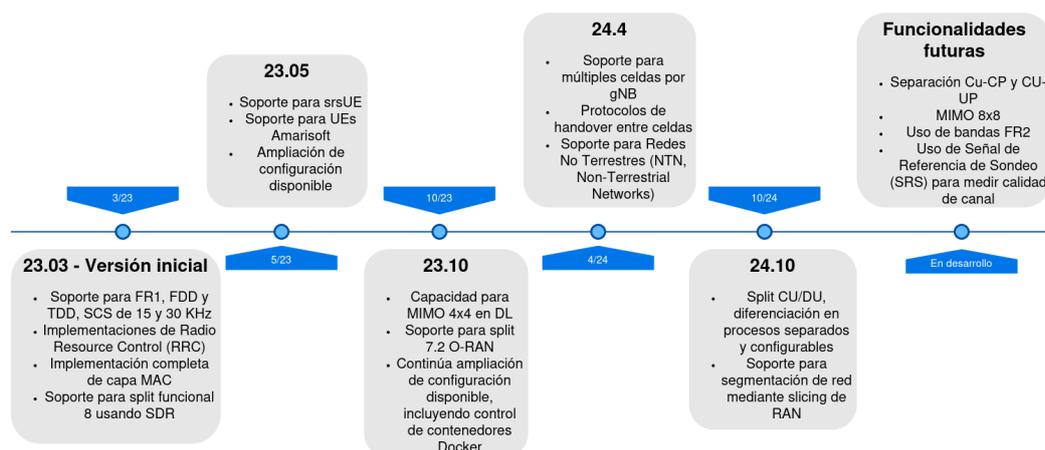


Figura 2.10: Diagrama de desarrollo del proyecto *srsRAN* a la fecha, incluyendo funcionalidades previstas para versiones futuras, en base a notas de versiones en [44].

Se trata de dos proyectos desarrollados y utilizados en conjunto, capaces de simular distintos elementos de la red. El proyecto *srsLTE* es el precursor de *srsRAN*, en el que se desarrollaron las funcionalidades necesarias para emular los componentes de una red LTE (eNodeB, EPC y UE 4G). A partir de la adaptación del módulo utilizado para simular el UE 4G, se incorporó la capacidad de emular un UE de quinta generación.

Por otro lado, el proyecto *srsRAN* 5G permite emular el funcionamiento de una gNB 5G, ofreciendo una variedad de parámetros de configuración disponibles. Se basa en un modelo de gNB compuesto por las partes RU, CU y DU, como fue estipulada en el “*Functional Split*” presentado en el *Release 15* [45] del 3GPP que sentó las bases de las redes 5G *Stand Alone*. Se simula entonces el funcionamiento de la RU y el par CU-DU como procesos separados pero interconectados (ver Figura 2.11).

Este proyecto ya es utilizado por diferentes compañías para desplegar sus propias redes 5G privadas, así como en el ámbito académico para desarrollo de redes y evaluación de equipos. Estos van desde ejemplos de *testbeds* de uso general, como [10] o [11] que presenta a su vez un conjunto de prácticas para el desarrollo de maquetas de este tipo; hasta varios estudios sobre sistemas particulares apoyados en *srsRAN* como por ejemplo *BeamArmor5G* [16]. Se trata de un sistema diseñado por el equipo *Wireless Communications Sensing and Networking* de la Universidad de California en San Diego que sirve de controlador de RAN inteligente (*RAN Intelligent Controller*, RIC), extendiendo las funcionalidades de capa física en la RAN para mitigar vulnerabilidades como el *jamming*. Se han realizado

2.2. Software utilizado

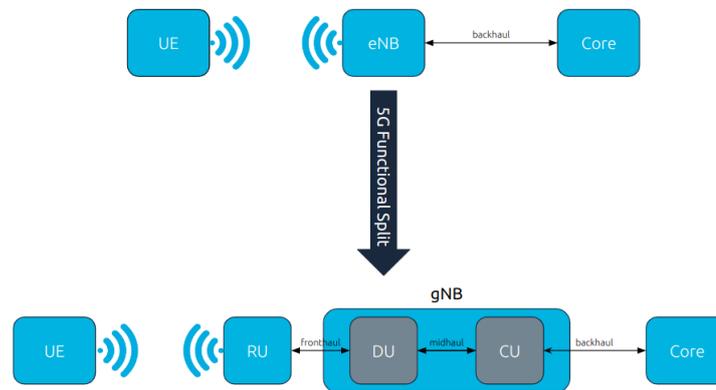


Figura 2.11: Cambios en arquitectura 5G implementados en *srsRAN* 5G según *Release 15*, vía documentación de *srsRAN* [44].

proyectos de desarrollo [16] y testeo [46] sobre este sistema apoyándose en el uso de *srsRAN* como una radiobase fácilmente desplegable.

2.2.2. Open5GS

Open5GS es una implementación de código abierto de un núcleo de red para sistemas de telecomunicaciones 4G LTE y 5G NR, siguiendo los estándares definidos por el 3GPP. Fue creada con el objetivo de ofrecer una alternativa accesible y flexible a los proveedores comerciales de infraestructura de red, permitiendo a los desarrolladores, investigadores y operadores experimentar con la implementación y prueba de servicios de una red móvil. La utilidad de *Open5GS* radica en su capacidad para simular y gestionar una red completa, tanto el plano de control como el plano de usuario, lo que facilita el desarrollo y la validación de nuevas tecnologías y servicios en un entorno de red controlado y configurable. Desde entonces, se ha utilizado en diversos centros de estudio y universidades alrededor del mundo, con fines que van desde el análisis de sus capacidades [47], implementaciones de redes privadas [48] e incluso se han realizado evaluaciones sobre posibles vulnerabilidades [49]. Al ser de código abierto, *Open5GS* fomenta la innovación y la colaboración dentro de la comunidad de investigación y desarrollo, permitiendo a los usuarios modificar y adaptar el núcleo de la red a sus necesidades específicas.

Dadas estas características, surge una intuitiva sinergia con *srsRAN*. Al ser ambas soluciones de código abierto, es posible integrarlas, utilizando *srsRAN* como la capa de acceso por radio (RAN). Por su parte, *Open5GS* se encarga del núcleo de la red, gestionando funciones críticas para el manejo de usuarios y datos. Esta integración logra una implementación de red móvil *end-to-end*, proporcionando un entorno de pruebas versátil y de bajo costo.

Es importante destacar en este punto que en el proyecto previo tomado como referencia directa para esta investigación [9], el núcleo de red fue implementado utilizando la solución de OAI. Esto introduce una diferencia significativa en la arquitectura del *testbed*, ya que si bien ambos núcleos siguen los lineamientos del

Capítulo 2. Marco teórico y soluciones de hardware y software utilizadas

3GPP, presentan variaciones en cuanto a implementación, estabilidad, facilidad de configuración y soporte comunitario. Si bien se observa en los resultados de estudios previos que estos presentan rendimientos similares (ver [12]), se percibe un mejor desempeño con *Open5GS* al escalar el número de UEs en la red. Dado que la comparación a realizar en este proyecto se mantiene en una cantidad limitada de UEs, se considera que el uso de *Open5GS* no representa un factor significativo al evaluar el rendimiento de la implementación del núcleo de red.

2.2.3. ZMQ

La librería ZeroMQ (ZMQ) es una implementación de un sistema asíncrono de colas de mensajes entre diferentes procesos, con la capacidad de intercomunicar procesos a través de diferentes protocolos de mensajería y de transporte. Se trata de un proyecto *open-source* implementado en C#, que incluye a su vez una librería de bajo nivel Libzmq para sistemas Unix.

Se utilizó durante la implementación inicial de una maqueta de red 5G (ver Subsección 3.2.1) por proveer una amplia variedad de protocolos de transporte, principalmente en este caso TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*) y mensajería *multicast*.

El principal uso de esta librería es mediante la generación de *sockets* en que los procesos involucrados establecen casillas de mensajes, y su administración para lograr una comunicación de baja latencia entre procesos. La implementación de *sockets* de ZMQ extiende la funcionalidad para permitir su uso en múltiples *end-points*, lo que amplía su capacidad de comunicar múltiples procesos mediante una única cola de mensajes.

2.3. Software Defined Radio

Los *Software Defined Radio* (SDR) o radios definidas por *software* son equipos que han propiciado un cambio significativo en el diseño y la implementación de sistemas de comunicación inalámbrica. A diferencia de los equipos de radio tradicionales, en los que los componentes de procesamiento de señal, como filtros, moduladores y demoduladores, están implementados en *hardware* dedicado, los SDR trasladan estas funciones a *software* ejecutado en *hardware* programable. Dicha traducción se logra mediante la digitalización temprana de la señal y el procesamiento posterior en unidades de cómputo, lo que permite una reconfiguración flexible sin necesidad de modificar físicamente los dispositivos. Basándose en el uso de sistemas *Field Programmable Gate Array* (FPGA), se emula la estructura de puertas de *hardware* de los dispositivos dedicados para cada caso [50].

La configuración de estos FPGA se especifica mediante un *Hardware Description Language* (HDL), que en el caso de los SDR Ettus usados en este estudio se aplican en el SDR mediante la programación de una “imagen FPGA” en forma de archivo “.bit”. Esta característica hace que los SDR sean ampliamente utilizados en aplicaciones de telecomunicaciones, redes móviles, e incluso comunicaciones es-

2.3. Software Defined Radio

paciales [51] y defensa [52], ya que pueden adaptarse a diferentes estándares y tecnologías mediante simples actualizaciones de *software*.

Sus orígenes datan de principios de los años 90, cuando avances en procesamiento digital de señales (DSP) y el aumento de la capacidad de cómputo permitieron trasladar funciones que históricamente se implementaban en *hardware* dedicado vía un entorno de *software*. Como en muchos otros campos, los primeros impulsos fueron dados por la industria militar, siendo un ejemplo el programa *SpeakEasy* [53] de la Fuerza Aérea de los Estados Unidos, cuyo objetivo era desarrollar un sistema de radio capaz de operar en múltiples bandas de frecuencia y protocolos sin requerir modificaciones físicas.

Actualmente son equipos de amplio uso, siendo particularmente útiles para el despliegue de redes móviles en entornos no industriales. Al soportar configuraciones variables y la posibilidad de cambiar parámetros en tiempo real, resultan prácticos para la experimentación e implementación de entornos de prueba. En este proyecto, la maqueta 5G se ha basado fuertemente en equipos SDR, específicamente los modelos Ettus X310 y Ettus B200, que se explorarán en la siguiente sección.

2.3.1. USRP Ettus X310

El SDR X310 es un dispositivo de radio definido por software de alto rendimiento desarrollado por Ettus Research, como parte de la serie *Universal Software Radio Peripheral* (USRP). La principal característica de su arquitectura es el uso de 2 *daughterboards* UBX de banda ancha que cubren el rango completo de frecuencias soportadas por la gNB de *srsRAN* a utilizar en este proyecto. A su vez, cuenta con una interfaz Ethernet 10 Gb y el conjunto de *USRP Hardware Drivers* (UHD) estandarizados, por lo que se pudo usar este equipo con una interfaz de alto rendimiento a los servidores implementados. Más características son detalladas en la Tabla 2.1. Por otra parte, las librerías UHD son de código abierto, por lo que presentan alta compatibilidad y varias herramientas para la configuración y puesta en marcha de los equipos. En particular, *srsRAN* es compatible y recomienda el uso de estas librerías.

Tabla 2.1: Especificaciones relevantes del USRP X310 [54].

Sistemas Operativos	Linux, Windows
Daughterboards RF	2 Tx, 2Rx
Rango de Frecuencias	70MHz a 6 GHz
BW	Hasta 160 MHz
Interfaces disponibles	1 y 10 Gbit Ethernet, PCIe
Alimentación	DC 12 V
Sincronización	Reloj configurable, GPSDO opcional

Por estas razones, el rendimiento del propio SDR no debería representar una limitante al medir el desempeño del resto de equipos de red utilizados en la maqueta. Sumado a esto, también fue el equipo utilizado en [9] durante su investigación

Capítulo 2. Marco teórico y soluciones de hardware y software utilizadas

y se quiere hacer una comparativa directa con dicho proyecto, por lo que se utilizó este SDR para la implementación inicial y la primera batería de pruebas realizadas.

2.3.2. USRP Ettus B200

El SDR B200 desarrollado por Ettus Research también fue utilizado a lo largo de este proyecto. A diferencia del USRP X310, el USRP B200 es un SDR de menores capacidades, diseñado para ser versátil y de bajo costo, operando con una única placa de menor porte que el X310 y una interfaz USB 3.0. Sin embargo, igualmente puede operar en un amplio rango de frecuencias y anchos de banda (ver Tabla 2.2), los cuales cumplen con las necesidades de este proyecto, siendo un buen equipo de transición a las implementaciones industriales.

Tabla 2.2: Especificaciones relevantes del USRP B200 [55].

Sistemas Operativos	Linux, Windows, MacOS
Daughterboards RF	1 Tx, 1Rx
BW	Hasta 56 MHz
Rango de Frecuencias	70MHz a 6 GHz
Interfaces disponibles	USB 3.0 SuperSpeed
Alimentación	USB Bus
Sincronización	Reloj configurable

En resumen

Siendo estos los dos equipos de radiofrecuencia utilizados como RU en la maqueta implementada, quedan así presentadas todas las componentes principales que permitieron el desarrollo de este proyecto de investigación así como los fundamentos técnicos necesarios para comprender el funcionamiento de una red 5G. Desde su evolución histórica y arquitectura hasta los aspectos clave de transmisión y acceso, incluyendo los componentes de *software* y *hardware* utilizados. Se procuró dar énfasis a aquellos elementos que constituyen la base teórica y práctica fundamental del *testbed*, tanto por su impacto funcional como por su relevancia en las decisiones tomadas durante la implementación.

No obstante, es importante destacar que se emplearon otros recursos que, si bien resultaron vitales en términos operativos —como antenas, atenuadores, mini computadoras del tipo NUC o *Raspberry Pi*, módulos de alimentación y componentes de interconexión—, no se incluyeron en detalle en este capítulo al salir del foco del análisis. Su estudio queda disponible para quienes deseen profundizar en los aspectos prácticos del montaje y despliegue físico del sistema, mediante recursos como el manual de usuario presentado en el Apéndice A.

Capítulo 3

Maqueta de red 5G

Este capítulo tiene como objetivo detallar la implementación de la maqueta de red 5G desarrollada en el marco del proyecto. Se pretende introducir el proceso de familiarización y posterior despliegue, describiendo claramente los componentes, equipos y configuraciones utilizadas, así como las diferentes variantes implementadas para evaluar el desempeño del sistema en distintos escenarios. Se busca de esta forma que el lector obtenga una visión global del funcionamiento y las diferentes decisiones tomadas durante la construcción de la maqueta, brindando además un marco sólido para la interpretación de los resultados que se presentarán más adelante.

3.1. Estructura base de la maqueta

El primer objetivo planteado fue lograr el despliegue de una maqueta funcional lo más sencilla posible, compuesta por los tres componentes fundamentales para lograr una conectividad de extremo a extremo: Núcleo de Red (5GCore), radiobase (gNB) y dispositivo de usuario (UE), tal y como se propone en el estándar diseñado por la 3GPP [28]. Esto permitiría en una primera instancia comparar su desempeño con la desarrollada anteriormente [9] basada en *OpenAirInterface*. Sin embargo, como se verá en el desarrollo del documento, dichos componentes (incluyendo configuración, *hardware* de base, etc) no permanecieron estáticos durante el desarrollo del proyecto, sino que por el contrario se planificó desde el inicio evaluar variantes, principalmente en el despliegue de la radiobase y los diferentes dispositivos de usuario a conectar.

A continuación se detallan las implementaciones logradas para cada uno de los elementos de la red que componen esta maqueta.

Núcleo de Red

Como se mencionó en la Subsección 2.2.1, *srsRAN* no dispone de una implementación de núcleo de red propia, por lo que se utiliza *Open5GS* para el despliegue del núcleo de red. Este puede instalarse directamente sobre el *hardware* utilizado,

Capítulo 3. Maqueta de red 5G

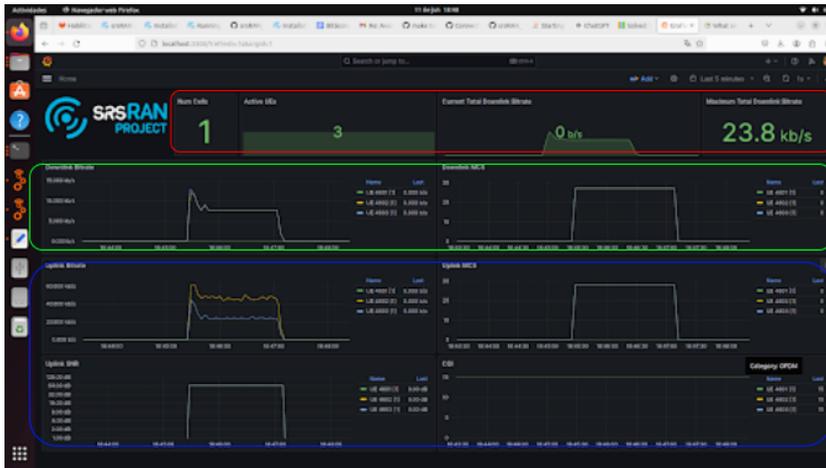


Figura 3.1: Panel de control de *Grafana*: enmarcado en rojo las estadísticas de tráfico total por la gNB, en verde estadísticas de tráfico en *downlink* con cada UE individual diferenciado por color, y en azul estadísticas de tráfico en *uplink* nuevamente diferenciado por UE.

en máquinas virtuales, o incluso se puede hacer un despliegue automático en contenedores Docker. Dado que la interacción directa con el núcleo no es el foco de trabajo en esta instancia, se optó por el despliegue automatizado utilizando contenedores. Para simplificar este proceso, se utilizó *Docker Compose* [56], que permitió coordinar y gestionar todos los servicios de manera eficiente. El repositorio del proyecto *srsRAN* [57] ofrece una solución preconfigurada que facilita el despliegue de este entorno, implementando no solo el contenedor de *Open5GS*, sino también dos servicios adicionales clave. El primero es *Grafana*, una plataforma de código abierto para la observabilidad, que permite un monitoreo en tiempo real del estado y rendimiento de la red. El *dashboard* de *Grafana*, presentado en la Figura 3.1, muestra métricas clave como el número de nodos y usuarios conectados, el uso de recursos de *hardware* y estadísticas detalladas del tráfico. Permite monitorear entonces el tráfico total y las características de la conexión con cada UE conectado, como *Channel Quality Indicator* (CQI) asignado o el *Modulation and Coding Scheme* (MCS). Esta información es proporcionada por *Metrics Server*, otro contenedor dedicado exclusivamente a la recolección y gestión de métricas provenientes de los diferentes contenedores en ejecución, lo que garantiza una supervisión eficiente y centralizada.

La configuración del núcleo de red se realiza mediante un conjunto de archivos y variables de entorno que son agregados al contenedor *Docker* de *Open5GS* durante su ejecución. Esto permite ajustar los parámetros con los que se quiere ejecutar mediante la configuración de protocolos y direcciones a utilizar, como, por ejemplo, las direcciones de los elementos del núcleo de red. A su vez, se configura el listado de usuarios disponibles en la red mediante una base de datos en formato *Comma Separated Values* (CSV), que contiene el par *Key/OPC* para cada uno, su identificador IMSI y una IP a asignarle cuando este se conecte a la red, entre otros. De esta manera, se autentica a los UE que se conecten a la red para controlar el tráfico que llega a estos.

3.1. Estructura base de la maqueta

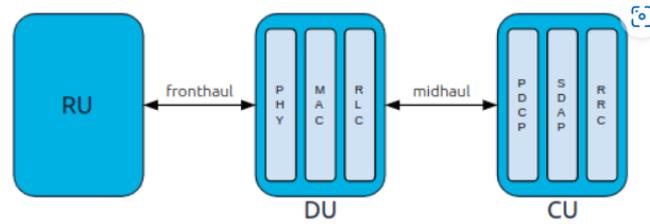


Figura 3.2: Separación CU/DU/RU implementada por *srsRAN* en *srsgNB*. Se detalla los protocolos delegados a cada unidad. Obtenida de [58].

La utilización de *Open5GS* como núcleo de red permitió validar la interoperabilidad entre componentes desarrollados por diferentes organizaciones, en este caso el *core* (*Open5GS*) y la RAN (*srsRAN*). A su vez, al ambos seguir las especificaciones del 3GPP, sería posible incluir al despliegue desarrollos de otros proveedores, siendo este un panorama auspicioso para crear un entorno de red desagregada.

Radiobase

Para el despliegue de la radiobase (gNB) se utilizó la implementación provista por *srsRAN Project*, que sigue los principios de la arquitectura O-RAN definida por la *Open-RAN Alliance* en su opción 8. Esta permite la implementación unificada o por separado de las componentes CU/DU de la gNB, como se puede apreciar en la Figura 2.11. Además, se delega la RU al SDR, cumpliendo con dicho *split*. En la Figura 3.2 se presentan en detalle los protocolos abarcados por cada unidad funcional. La radiobase de *srsRAN* gestiona las comunicaciones entre la red de acceso radioeléctrico (RAN) y el núcleo, y admite múltiples tecnologías de acceso, como 4G y 5G.

Es importante destacar que *srsRAN Project* permite ejecutar la gNB de dos formas diferentes. La primera opción es como un gNB monolítico, donde se combinan las funciones de la Unidad Central (CU) y la Unidad Distribuida (DU) en un único bloque. La segunda opción es utilizar un enfoque dividido, donde la CU y la DU se ejecutan por separado. Esto habilita una mayor flexibilidad en la red, al permitir que la CU asuma el control centralizado mientras que la DU gestiona las funciones asociadas al plano físico, pudiendo mejorar la eficiencia y escalabilidad del sistema. Es particularmente interesante en este caso ya que se tiene como fin la implementación de una versión portable de la maqueta, buscando una utilización mínima de recursos. Esta segunda opción puede resultar clave para este fin, permitiendo la ejecución de la gNB en dos equipos de menor porte.

Esta implementación, que permite configurar la gNB para adaptarse a distintos entornos de red como *edge computing* o despliegues en la nube, tiene el fin de promover redes de acceso radioeléctrico (RAN) virtualizadas y desagregadas. Según sus desarrolladores, esto se logra utilizando un enfoque altamente interoperable y compatible con múltiples proveedores. En la página oficial [44] es posible encontrar el *link* al repositorio de Github [57] donde se encuentran todos los archivos necesarios para compilar y ejecutar localmente la radiobase.

Capítulo 3. Maqueta de red 5G

Equipo de Usuario

Por último, se debió implementar un equipo de usuario. Como se explicó en el apartado anterior, *srsRAN Project* no provee una implementación específica para un UE 5G. Sin embargo, es posible utilizar su antecesor *srsRAN 4G* para testear la conectividad y sus funcionalidades. En la documentación de *srsRAN*, *srsUE* se describe como un software emulador de equipo de usuario (UE) para redes (inicialmente) 4G LTE. Este emulador ofrece una implementación completa del protocolo LTE, incluyendo el control de acceso de radio (RAC), la gestión de sesiones y la transferencia de datos. Este provee soporte para 5G NR en sus modos *Non-Standalone* y *Standalone*. Es compatible únicamente con esquemas *Frequency Division Duplex* (FDD) y maneja anchos de banda tales como 1.4, 3, 5, 10, 15 y 20 MHz. También soporta hasta MIMO 2x2 y modulación de hasta 256 QAM (solo en el enlace descendente). Otra limitante que surge es que solo puede utilizar un *Sub Carrier Spacing* de 15 kHz, valor que en 5G es variable y puede alcanzar hasta 120 kHz. Se decidió trabajar inicialmente con esta implementación ya que además se proveen varias configuraciones modelo para utilizar con el equipamiento disponible, como son los SDR USRP X310 y B200 de *Ettus Research*.

Esta no es la única opción que ofrece el grupo SRS para trabajar con una terminal de usuario, ya que también es posible trabajar con un *Commercial Off-The-Shelf User Equipment* (COTS UE). En su documentación es posible encontrar la lista de equipos probados hasta el momento, tanto por los desarrolladores como por los usuarios, permitiendo así la implementación de un caso de uso real, el cual se abordará más adelante. Trabajar con un dispositivo 5G nativo, como es el Samsung Galaxy A33 5G utilizado, permitió explotar a fondo funcionalidades propias de las redes móviles de quinta generación y evaluar su impacto cuantitativo en situaciones reales, sobre todo en contraste con el UE 4G modificado implementado anteriormente.

Finalmente, durante el desarrollo del presente trabajo se conoció la existencia en plaza del *Sixfab 5G Development Kit* para *Raspberry Pi*, dispositivo que generó interés por explorar una alternativa intermedia entre la solución *srsUE* y un dispositivo comercial (COTS UE). Dadas las limitaciones comentadas de *srsUE*, y que, si bien un COTS UE permite evaluar el sistema en condiciones más realistas, acarrea la desventaja de un control limitado sobre la configuración interna, un punto medio resulta atractivo. En este contexto, el kit de desarrollo *Sixfab 5G* para *Raspberry Pi* aporta un balance, combinando una interfaz de *hardware* modular con capacidades de configuración y control propias. Además, permite evaluar la viabilidad técnica y económica de implementar terminales de usuario en despliegues específicos, como pueden ser aplicaciones de IoT basadas en tecnología 5G.

Estas implementaciones también pusieron a prueba la interoperabilidad de la solución propuesta con una gama diversa de UEs, cada uno con características distintas en cuanto a sistema operativo, soporte de bandas, niveles de configuración accesibles y tipo de integración. Al lograr establecer conexiones funcionales en todos los casos, se evidencia capacidad de *srsRAN* para integrarse de forma flexible con diferentes implementaciones y se pone de manifiesto la madurez alcanzada por las soluciones open-source actuales en términos de interoperabilidad y adaptabili-

dad.

3.2. Procedimiento

En esta sección se detallan los procedimientos realizados para implementar la maqueta en dos entornos diferentes: primero uno simulado dentro de un único equipo, y posteriormente despliegue físico en laboratorio utilizando equipos de radiofrecuencia. El objetivo es mostrar las principales configuraciones necesarias para reproducir y entender el comportamiento de la red en ambas modalidades.

3.2.1. Entorno virtualizado

La primera aproximación al uso del proyecto *srsRAN* para la implementación de esta maqueta se realizó usando una red virtualizada dentro de una única PC. Esto permitió al equipo de trabajo familiarizarse con los archivos de configuración, tanto de *Open5GS* como de *srsRAN*. La interfaz entre los diferentes equipos se implementó usando ZMQ, que permitió comunicar los diferentes equipos simulados en una red virtual (ver Subsección 2.2.3).

Para comunicar los diferentes elementos de la red, se configuraron dos subredes principales:

- La subred 10.53.1.0/24 en la que se comunican el *core* implementado en *Open5GS* y la gNB implementada en *srsRAN*. Se utiliza ZMQ para implementar sockets en las direcciones 10.53.1.2 (dirección de lectura del *core Open5GS*, y de escritura de la gNB), y 10.53.1.1 (dirección de escritura del *core Open5GS*, y de lectura de la gNB).
- La subred 10.45.1.0/24 se utiliza para comunicar la gNB con los diferentes UEs. La gNB toma la dirección 10.45.1.1, y los UE se configuraron en las direcciones de 10.45.1.2 a 10.45.1.8 .

En la Figura 3.3 es posible observar de forma gráfica la red desplegada. Las direcciones del núcleo de red y de la gNB se designaron editando los archivos de configuración de cada aplicación antes de su ejecución. La dirección del núcleo de red se ingresó como una variable de entorno del contenedor *Docker*, mediante la edición del archivo `envvars` correspondiente. En el caso de la gNB, se configuró la subred 10.45.1.0/24 para comunicar la gNB con los posibles UEs . Esta también fue definida mediante una variable de entorno, y luego durante la ejecución al registrar la gNB en la red el núcleo le asigna la primera dirección posible (10.45.1.1).

Al trabajar con los UEs, inicialmente estos se agregaron en la base de datos de suscriptores del *core Open5GS*, para que al entrar en contacto con la gNB e intentar registrarse se les asigne una dirección. En esta base de datos se agregaron hasta 3 UEs, con las direcciones mencionadas anteriormente. Esto se hizo agregando un archivo `subscriber_db.csv` al contenedor de *Open5GS* antes de ejecutar. De todos modos, este también cuenta con una interfaz gráfica en la que se pueden agregar suscriptores adicionales en tiempo de ejecución. La misma se presenta en

Capítulo 3. Maqueta de red 5G

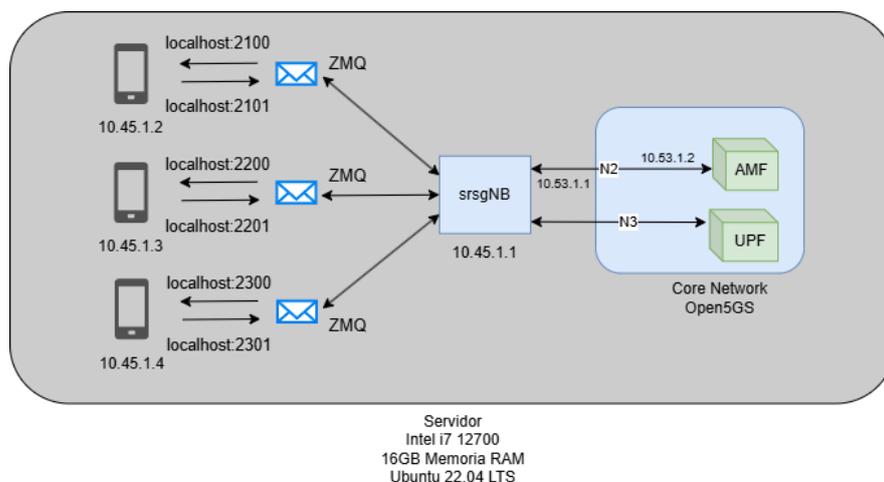


Figura 3.3: Diagrama de subredes configuradas en entorno virtualizado en una misma PC, con radiobase y 3 UEs conectados por interfaces locales levantadas mediante la librería ZMQ.

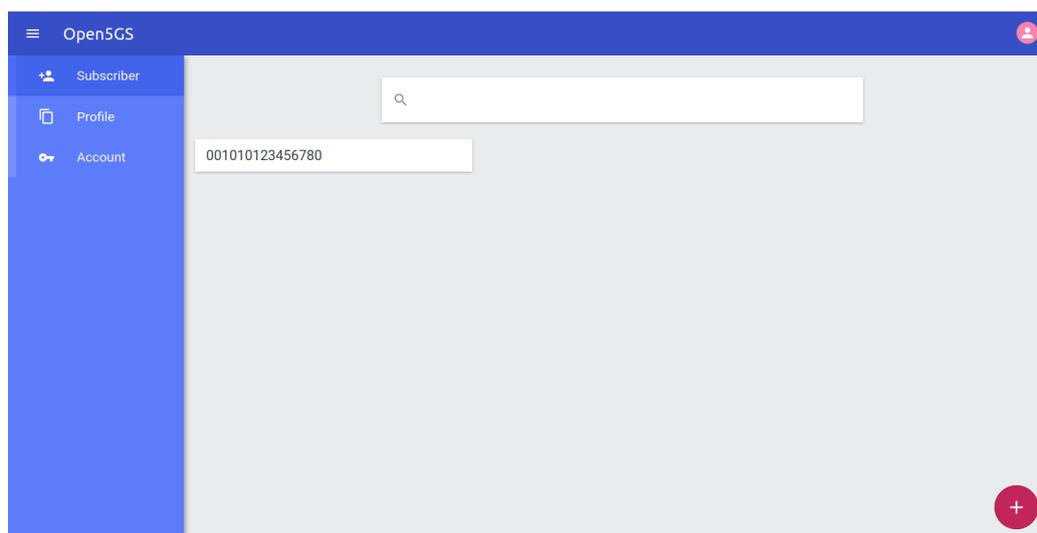


Figura 3.4: Interfaz del core *Open5GS*, con un UE registrado.

la figura Figura 3.4, donde puede observarse un único suscriptor cargado, de IMSI valor 00101123456780.

Se determinó el uso de ZMQ como interfaz en el archivo de configuración de la gNB y cada uno de los UEs al ejecutar. En la sección Subsección 3.2.2 se presentan a fondo los parámetros de interés. Una vez en ejecución, srsgNB y srsUE trasladan los paquetes transmitidos al administrador ZMQ que se encarga de la distribución de mensajes.

Se registraron y monitorearon las comunicaciones entre los equipos usando un contenedor con la aplicación *Kubernetes Metrics Server*, así como otro contenedor con *Grafana* (cuya interfaz gráfica se presentó en la Figura 3.1) para visualizar y graficar información de la red en tiempo real.

Es importante destacar en este punto que, en el *setup* provisto por *srsRAN*, es la radiobase la que envía la información del tráfico de la red a *Grafana*. Por lo tanto, en el archivo de configuración, se debe tener la variable `enable_json_metrics` en *True*, y se debe asignar correctamente una IP y un puerto hacia donde redirigir dichas métricas.

3.2.2. Entorno físico

Una vez verificado el funcionamiento de la red 5G en un entorno virtualizado dentro de un único equipo, se procedió al despliegue en un ambiente de laboratorio físico (utilizando servidores, equipos SDR, antenas, entre otros). En primera instancia el objetivo fue replicar la maqueta implementada en el proyecto antecedente [9]. Por lo tanto, se utilizaron los mismos equipos empleados en aquella oportunidad. Estos fueron dos servidores, los cuales a lo largo de este capítulo serán identificados como *Servidor A*, que alojó tanto el núcleo de red como la gNB, y *Servidor B*, encargado del procesamiento del UE. Ambos equipos cuentan con procesadores Intel i7 12700 con 16 GB de memoria RAM y utilizando Ubuntu 22.04 LTS.

Para esto, se utilizaron como base las configuraciones provistas en la sección *srsRAN gNB with srsUE* de la documentación oficial [44]. En esta se proveen configuraciones tanto para la gNB como para el UE. Antes de trabajar con estos archivos, se deben seguir los mismos pasos de ruteo en los equipos servidor que en la sección Subsección 3.2.1. Posteriormente, se deben editar los archivos de configuración. En el caso del archivo de configuración de la gNB, los principales parámetros a modificar y sus valores fueron:

- `device_driver`: El nombre del *driver* RF a utilizar. En este caso el valor asignado fue *uhd*, correspondiente a los *drivers* utilizados para manejar los equipos X310.
- `device_args`: Este parámetro puede incluir diferentes valores asociados al SDR a utilizar. En la documentación oficial se indica que se debe ingresar modelo de SDR, de la forma *type=X310*. Sin embargo, al tener un único SDR conectado, no fue necesario especificarlo. En este caso, lo único que se estableció fueron los tamaños de trama a enviar y recibir, indicando *send_frame_size=8000, recv_frame_size=8000*.
- `sync`, `clock`: Estos parámetros se utilizan para especificar la fuente de sincronización horaria y la fuente de referencia temporal del dispositivo RF respectivamente. Ambos son de vital importancia en caso de utilizar una fuente externa, lo cual es altamente recomendado por el equipo de desarrollo de *srsRAN*. En este caso, al no contar con una, se dejaron estos campos con valor *internal*.
- `srates`: Tasa de muestreo del dispositivo RF en MHz. El valor recomendado, y utilizado para el primer establecimiento de conexión fue *23.04*.
- `tx_gain`, `rx_gain`: Valores para ganancia en transmisión y recepción respectivamente, en dB. Estos deben estar entre 0 y el máximo soportado por

Capítulo 3. Maqueta de red 5G

el dispositivo RF. Su valor debió ser ajustado de forma manual hasta lograr una primera conexión, como se verá en la siguiente sección.

- **dl_arfcn:** ARFCN (*Absolute Radio-Frequency Channel Number*) de la portadora en *Downlink* (Frecuencia central). El valor del ARFCN en 5G indica el número absoluto de canal de radiofrecuencia. En el caso inicial se trabajó con el canal por defecto, de valor *368500*.
- **band:** La banda de NR en la cual se trabajará. En el caso inicial se trabajó con la banda por defecto, valor *3*.
- **channel_bandwidth_MHz:** Ancho de banda del canal, en MHz. El número de PRBs se deriva automáticamente de este valor. Se trabajó inicialmente con la configuración recomendada, de valor *20 MHz*, correspondiente en este caso a *106 PRBs*.
- **common_scs:** Valor para el *Subcarrier Spacing*, es decir, el espaciamiento entre subportadoras, en kHz. Se seteó en *15*, el valor sugerido inicialmente.
- **plmn:** El número de *Public Land Mobile Network* transmitido por la gNB. Se utilizó el valor por defecto “*00101*”.

Por otra parte, se debieron establecer las configuraciones principales para el UE.

- **srate:** Mismo parámetro que en gNB. Debe coincidir con el valor utilizado en el nodo. Se debe incluir “*e6*” luego del valor para indicar que se está trabajando en MHz.
- **tx_gain, rx_gain:** Mismos parámetros que los descritos para la gNB. Los valores indicados son propios del UE y se deben establecer independientemente de los de la gNB.
- **freq_offset:** Valor a compensar en caso de tener un *offset* en frecuencia. Se encontró que el *software* logra compensar valores de hasta *1 kHz* aproximadamente de forma automática. En caso de sufrir un *offset* mayor, se debería compensar utilizando este parámetro. Inicialmente se dejó establecido en *0*.
- **nof_antennas:** Número de antenas por *carrier*. Se estableció en *1*.
- **device_name:** Parámetro equivalente a *device_driver* en gNB.
- **bands:** Parámetro análogo a *band* en gNB. Debe coincidir con la banda establecida en el nodo.
- **max_nof_prb, nof_prb:** Valores de *Physical Resource Block* (PRB). Estos parámetros deben adaptarse según ancho de banda utilizado. Inicialmente, al trabajar con un ancho de banda de *20 MHz* y *15 kHz* de *Subcarrier Spacing*, se debe establecer en *106*.

3.2. Procedimiento

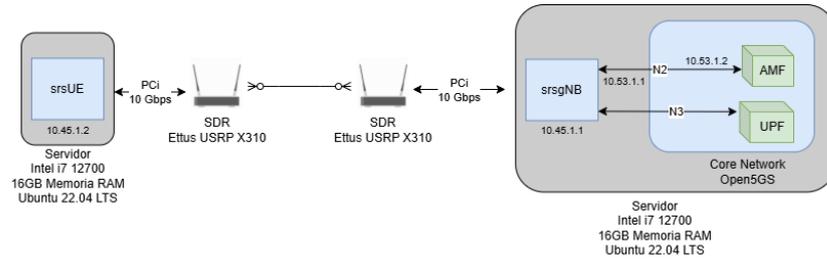


Figura 3.5: Diagrama de interconexión al trabajar con dos SDR X310, conectados por cable coaxial.

- `dl_ssb_arfcn`: Valor del ARFCN en *Downlink*, establecido automáticamente por la gNB al utilizar un ARFCN determinado. Inicialmente no fue necesario utilizar este parámetro en la configuración predeterminada, pero a medida que se cambiaron las bandas sí lo fue.
- `usim`: Esta es una subsección de parámetros que se encarga de establecer las credenciales que utilizará el UE para autenticarse contra el núcleo de red. Estas credenciales debieron ser previamente ingresadas como válidas en el *core*. Los subparámetros a establecer fueron *mode*, *algo*, *opc*, *k*, *imsi* e *imei*.

Es importante destacar que tanto para la gNB como para el UE existe una gran variedad de parámetros que fueron intencionalmente omitidos en esta sección ya que no resultaron relevantes para las configuraciones iniciales. Más adelante se profundizará en algunos al trabajar en otros escenarios.

Por otro lado, las interfaces de radio se implementaron inicialmente utilizando dos SDR modelo USRP X310, conectados mediante una interfaz 10 GbE a los servidores. Además, se utilizaron antenas *Log Periodic WA5VJB* (al igual que en el proyecto antecesor [9]), conectando dos de estas a cada SDR para transmisión y recepción, respectivamente.

En un primer paso, se realizaron pruebas conectando los SDR entre sí mediante cables coaxiales utilizando atenuadores de 30 dB, con el fin de comprobar el correcto funcionamiento al utilizar un medio físico de interconexión. En la Figura 3.5 se presenta un esquema lógico, con el fin de ilustrar con mayor claridad las conexiones realizadas, y en la Figura 3.6 se expone un registro fotográfico del despliegue físico. En este punto, al confirmar que los equipos podían “verse” entre sí, lo único que se debió ajustar manualmente fueron las ganancias para lograr una conexión estable, ya que el resto de las configuraciones se mantuvieron iguales al caso virtualizado. Posteriormente, una vez funcional el sistema cableado, se pasó a la conexión por interfaz inalámbrica.

En esta instancia, se dispusieron los equipos enfrentados, con línea de vista directa (ver Figura 3.7). Para las primeras pruebas, se configuró el uso de la banda `n3` del *Frequency Range 1 (FR1)*, especificando tanto el número de banda como el ARFCN. La elección de esta banda se debió a su compatibilidad con las implementaciones de gNB y UE ofrecidas por *srsRAN*. Como se verá en los siguientes capítulos, posteriormente se exploraron otras bandas.

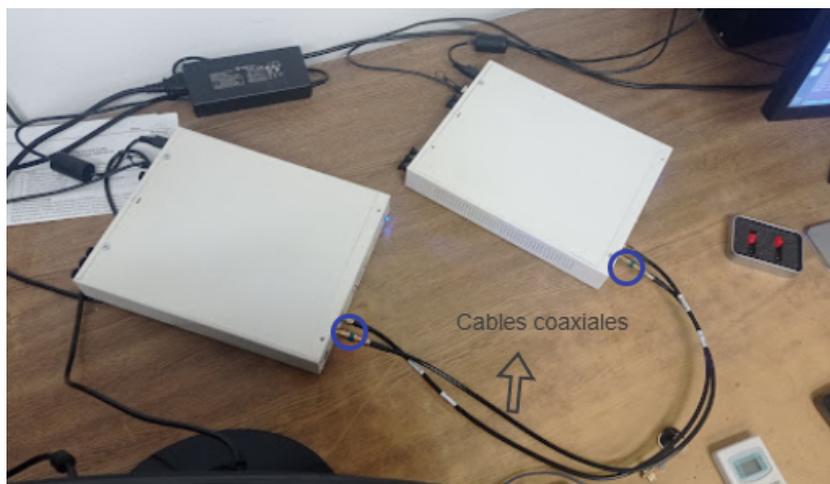


Figura 3.6: Interconexión de dos SDR X310 mediante cables coaxiales. En azul se circulan los atenuadores de 30 dB en transmisión.

Los parámetros principales evaluados fueron las ganancias y los ajustes del *offset* en frecuencia tanto en la gNB como en el UE. Para ello se revisaron los logs de ejecución, donde se pudo observar el valor *cfo* (*Central Frequency Offset*) registrado. Estos valores se variaron manualmente, con el fin de obtener una comunicación estable. Se notó que la estabilidad del enlace presentaba una alta sensibilidad a los cambios en las ganancias de las antenas. Una ganancia muy alta en transmisión, de cualquiera de las partes, saturaría el receptor. Este proceso también se realizó de forma manual hasta lograr una conexión exitosa.

3.3. Caracterización de la maqueta

Una vez lograda una maqueta funcional con enlace inalámbrico como la que se puede observar en la Figura 3.7, se procedió a caracterizar su funcionamiento con el objetivo de relevar los requerimientos de *hardware*. A partir de esta caracterización, se definió el *hardware* necesario para trasladar la radiobase a equipos de menor porte que las PCs utilizadas en la etapa inicial.

El principal foco de análisis fue el consumo de recursos computacionales, como memoria y capacidad de procesamiento. Para ello, se monitoreó en tiempo real el uso de estos recursos durante la ejecución de cada prueba, registrando el comportamiento de los procesos asociados a la radiobase. Se prestó especial atención a métricas como el porcentaje de uso de CPU y la cantidad de memoria RAM utilizada, ya que estos factores representan las principales limitantes al seleccionar el *hardware* más adecuado para la fase de portabilización.

En el caso del núcleo de red, al ejecutarlo en un contenedor *Docker*, se decidió hacer este relevamiento mediante la herramienta incorporada *docker stats*. Esta reporta el uso de recursos de todos los contenedores en ejecución, siendo en este caso el más relevante el contenedor de *Open5GS*. Se trata de una interfaz limitada,

3.4. Variantes en gNB

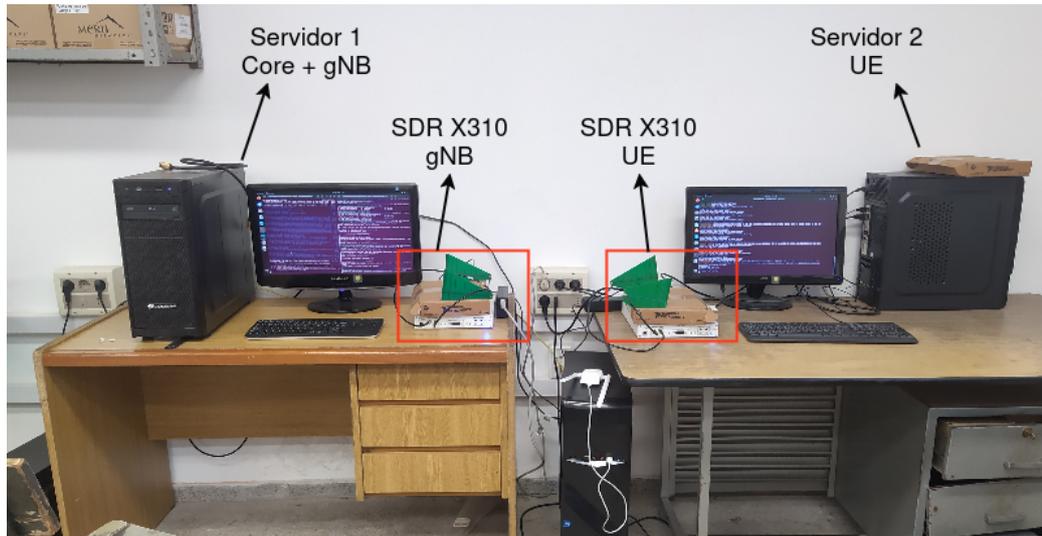


Figura 3.7: Maqueta final con SDRs USRP X310, interconectados por aire. Se señalan en la imagen los servidores y SDR, cada uno con su función. Sobre los SDR, en color verde, se pueden observar las antenas direccionales.

pero que de igual manera se puede utilizar para reportar confiablemente el uso de los recursos más relevantes como el uso de CPU y memoria RAM en cada momento.

Por otro lado, tanto el proceso usado para emular el gNB en la PC servidor como el srsUE en la PC UE se ejecutaron mediante procesos nativos en el *hardware*, por lo que para estos se decidió utilizar la herramienta *htop* de Linux. Esta provee una funcionalidad similar a la función *top* de Unix, pero con una interfaz gráfica más clara y configurable. A través de la misma se pueden observar los recursos utilizados por cada proceso del sistema, incluyendo no solamente las métricas de uso de RAM y CPU buscadas, sino incluso memoria virtual y residente en RAM o uso de memoria de swap, entre otros.

3.4. Variantes en gNB

En esta sección se describen las distintas variantes implementadas sobre la radiobase (gNB) durante el desarrollo del proyecto. El objetivo principal fue evaluar configuraciones alternativas al despliegue inicial, explorando desde la caracterización de su comportamiento en *hardware* convencional hasta implementaciones en dispositivos compactos, así como el uso de separaciones funcionales CU/DU propuestas por el proyecto *srsRAN*.

3.4.1. Implementación de gNB en NUC

Una vez obtenidos los valores aproximados del consumo de recursos de la radiobase (ver Sección 3.3), se procedió a seleccionar un nuevo equipo que pudiera funcionar como servidor de la gNB. De forma general (los detalles cuantitati-

Capítulo 3. Maqueta de red 5G

vos se presentan en secciones posteriores), se observó que el uso de CPU y RAM no representa una limitación crítica. Sin embargo, la interfaz de conexión con el SDR se identificó como un posible cuello de botella, dado el alto volumen de datos transmitidos. Esto resulta razonable según lo expuesto en el artículo [59], donde se mencionan los altos requerimientos para la interfaz de *fronthaul* de un *Split 8* como es este caso, en contraposición a otras variantes. Se observa que este despliegue puede requerir tasas hasta 100 veces mayores que, por ejemplo, un *Split 6*.

Los únicos equipos disponibles con interfaz Ethernet de 10 Gbps eran los servidores de escritorio utilizados en la fase inicial y los SDR USRP X310. Con el objetivo de reducir el tamaño y consumo del sistema, se optó por evaluar el funcionamiento con los SDR USRP B200, que utilizan una interfaz USB 3.0, cuya velocidad teórica máxima es de 4.8 Gbps. Esto trae como consecuencia la limitación de anchos de banda soportados o la posibilidad de implementar MIMO en configuraciones avanzadas.

Además de permitir utilizar equipos de cómputo de menor consumo y con la portabilidad como uno de los ejes del proyecto, se reduce naturalmente el tamaño de la maqueta en general. El SDR B200 tiene unas dimensiones de 9.7 cm x 15.5 cm x 1.5 cm y un peso de 350 g, mientras que el X310 ocupa 26.7 cm x 21.8 cm x 4.1 cm y pesa 1.6 kg.

En esta línea, se identificaron en Facultad equipos *Intel NUC (Next Unit of Computing)*, utilizados previamente en proyectos como computadoras a bordo de drones o barcos [5]. Esta característica los posicionó como candidatos adecuados para una implementación portátil de la radiobase, ya que cumplen con los criterios de tamaño, conectividad y consumo establecidos en la sección Subsección 1.2.1.

Las especificaciones del equipo NUC utilizado fueron las siguientes:

- Sistema Operativo: Ubuntu 22.04 LTS
- Procesador: Intel Core i3-7100U @ 2.40 GHz (4 hilos)
- Memoria RAM: 16 GB
- Gráficos: Intel HD Graphics 620
- Interfaces de Entrada/Salida: USB 3.0, Micro HDMI
- Dimensiones: 104.1 mm x 101.6 mm x 27 mm
- Peso: 225 g

Al momento de la ejecución de la gNB basada en el Intel NUC y SDR B200, se observó que el consumo de recursos se encontraba dentro de lo esperado. Sin embargo, se encontraron en los logs de la radiobase un alto porcentaje de *Underflows*. Este error proviene del SDR indicando que no está pudiendo ser alimentado con la cantidad de datos esperada para un funcionamiento correcto. Dado que el porcentaje de uso de CPU es relativamente bajo, manteniendo un promedio de 100%, un único núcleo (ver Figura 3.8), se señaló como posible cuello de botella a la interfaz USB 3.0, la cual se utiliza tanto para proveer energía como para transferir datos. Este tipo de inconvenientes son usuales, como se puede observar en [60]. Allí

3.4. Variantes en gNB

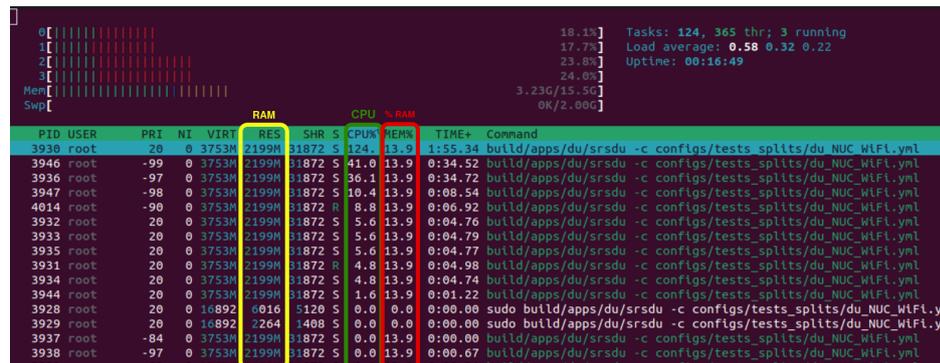


Figura 3.8: Captura de pantalla de comando *htop* en NUC corriendo srsDU a 20 MHz de ancho de banda. Se destaca el uso de CPU (en verde) y RAM (cantidad total residente en memoria en amarillo, y porcentaje de la memoria total en rojo).

los desarrolladores recomiendan incrementar los *buffers* de las interfaces de conexión al máximo posible. Incluso han desarrollado el *script* “*srsran_performance*” que lo hace automáticamente. Este fue adoptado por el equipo de trabajo y se ejecutó antes de comenzar cada prueba.

Si bien la interfaz teóricamente debería alimentar al SDR holgadamente, en equipos compactos como son el NUC de Intel estas pueden sobrecalentar y disminuir así su desempeño. No se realizó un estudio exhaustivo de este fenómeno ya que se entendió estaba por fuera del alcance de este proyecto. En lo práctico, estos *Underflows* resultaban en conexiones inestables que caían al comenzar la transferencia de paquetes, o directamente errores de sincronización en las conexiones.

Se procedió a disminuir el *Sample Rate* (y por ende el ancho de banda del canal), bajándolo gradualmente desde 20 MHz hasta 5 MHz, donde los *Overflows* disminuyeron significativamente y se alcanzaron conexiones estables con un UE. Se concluyó que a esta tasa, la menor posible, era la única viable para el correcto funcionamiento de la maqueta en esta configuración.

3.4.2. Split CU/DU

Dentro de las variantes ofrecidas por *srsRAN Project*, existe la posibilidad de implementar la radiobase en forma de un *Split* CU/DU, pudiendo hacerlo de diversas formas. Durante el desarrollo del proyecto, al utilizar equipos SDR, se trabajó en todo momento utilizando un *Split* 8 entre DU y RU. No se contó con equipos que hicieran posible una implementación en formato 7.2, dónde la separación ocurre dentro de la capa física, dividiendo el procesamiento en la transformación de la señal y la modulación. Para ellos se requieren equipos específicos como el *Benetel 550* [61] dónde el RU se encarga de la modulación y conversión a señales RF, mientras que la DU maneja las funciones de pre-procesamiento.

Sin embargo, sí ofrece la posibilidad de separar la CU y DU mediante la ejecución de dos *software* distintos. Se tuvo como fin lograr una separación física de estos componentes, implementando así un *Split* 2.

Capítulo 3. Maqueta de red 5G

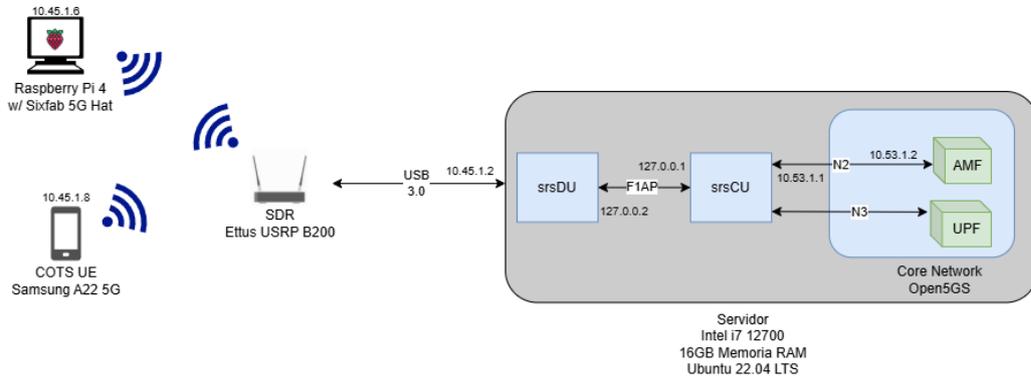


Figura 3.9: Esquema de Red con separación CU/DU en un mismo servidor.

Para implementar esta arquitectura en la maqueta, se siguieron las indicaciones proporcionadas por *srsRAN* en su página oficial [44]. Se instaló la versión más reciente de *srsRAN* en un PC con Ubuntu 22.04.1 LTS y se integró *Open5GS* como núcleo 5G, ejecutándose de manera nativa en la máquina *host*. Como *frontend* de RF, se empleó un dispositivo USRP B200, conectado al PC mediante USB 3.0. Esta decisión fue tomada con foco en la versión portable (este SDR no necesita una conexión dedicada de energía), y porque el USRP X310 presentó alta inestabilidad luego de uso prolongado.

Para la configuración, se generaron dos archivos esenciales: uno para *srsCU* y otro para *srsDU*. Estos archivos definieron parámetros clave, como direcciones IP, puertos de comunicación y configuraciones específicas de la red. La CU y la DU fueron desplegadas inicialmente en la misma PC servidor, con el fin de familiarizarse con las configuraciones, utilizando interfaces virtuales. Esa configuración se puede observar en la Figura 3.9

Si bien esta configuración no presenta mayores diferencias en el despliegue con la gNB anteriormente implementada, además de servir como primera experimentación, es útil para profundizar en el entendimiento de las interfaces que participan en la comunicación y el rol de cada una. En particular, se puede observar cómo la CU es la encargada de comunicarse con el AMF mediante el plano de control, y crea la interfaz F1AP donde escuchará tráfico proveniente de la(s) DU adyacente(s). Por otra parte, la DU es la sección que define la comunicación con la RU (en este caso un SDR) y además establece los parámetros de transmisión de la celda.

Para la ejecución fue suficiente con asignar correctamente las IP en los archivos de configuración. Los parámetros a modificar fueron el `f1ap: bind_addr` en *srsCU*, al cual se le dio el valor 127.0.0.1. Este asigna una IP local a la interfaz F1AP, la cual usarán el CU y DU para comunicarse. Análogamente, en *srsDU* se modificó el `f1ap: cu_cp_addr`, que apunta a la IP anterior, `f1ap: bind_addr` y `nru: bind_addr`, parámetro que indica a la sección RU en qué IP está la DU. Estos últimos dos coinciden en este caso, ya que nos encontramos en un *Split* 8. Se le asignó el valor 127.0.0.2. Una vez configurado, se procede a ejecutar primero el contenedor *Docker* con *Open5GS*, luego *srsCU* y por último *srsDU*.

3.4. Variantes en gNB

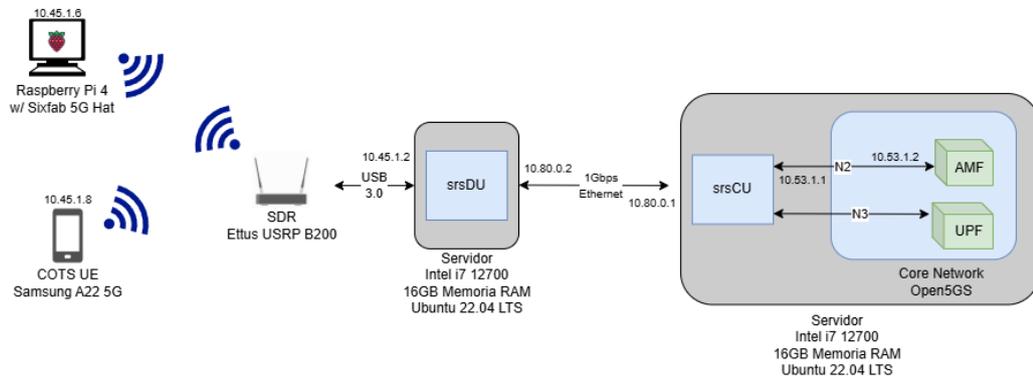


Figura 3.10: Esquema de Red con separación CU/DU conectada mediante cable Ethernet.

Posteriormente, se pasó esta versión a servidores separados como se presenta en la Figura 3.10, ambos con capacidades idénticas, como las mencionadas en la sección Subsección 3.2.2, con el fin de evitar cuellos de botella generados por el *hardware*. En esta instancia, se verificó que la utilización del procesamiento de los servidores era notoriamente bajo (ver Sección 3.3). Por lo tanto, abordando el objetivo de trabajar en la portabilización del sistema, se implementó la sección DU en una mini PC Intel NUC como la descrita en la sección anterior. Se repitieron entonces las pruebas realizadas con el ejecutable de la gNB completa, con el fin de comparar el desempeño en ambos casos.

La separación física entre CU/DU se implementó de dos formas: una conexión cableada mediante Ethernet 1 Gbps y otra conexión inalámbrica a través de Wi-Fi. Esta decisión fue motivada por la necesidad de evaluar el impacto del medio en el desempeño de la red y analizar las diferencias en términos de latencia, estabilidad y capacidad de transmisión.

La conexión mediante Gigabit Ethernet se estableció para garantizar una comunicación confiable y de baja latencia entre la CU y la DU. Ethernet proporciona un canal dedicado con alta capacidad de transmisión y menor interferencia en comparación con medios inalámbricos, fundamental en este tipo de arquitecturas desagregadas donde los paquetes de datos deben ser transportados con tiempos estrictos de entrega. Si bien en implementaciones reales se utilizan enlaces de fibra óptica, este tipo de conexión se alinea con los despliegues tradicionales de redes 5G centralizadas y permite validar el correcto funcionamiento del sistema sin mayores interferencias. En la Figura 3.10 se puede observar un esquema de la configuración implementada. Además, se debieron configurar las interfaces locales Ethernet en ambos servidores, con las IPs correspondientes en la red 10.80.0.0/24.

Por último, la conexión inalámbrica a través de Wi-Fi se implementó para explorar la viabilidad de establecer enlaces flexibles y reducir la dependencia de infraestructuras físicas en ciertos escenarios. Esta configuración permite evaluar el rendimiento en entornos donde no es viable un despliegue cableado. Sin embargo, Wi-Fi introduce variaciones en latencia y estabilidad debido a la congestión de la banda, interferencias y fluctuaciones en la calidad del enlace, lo que podría afectar

Capítulo 3. Maqueta de red 5G

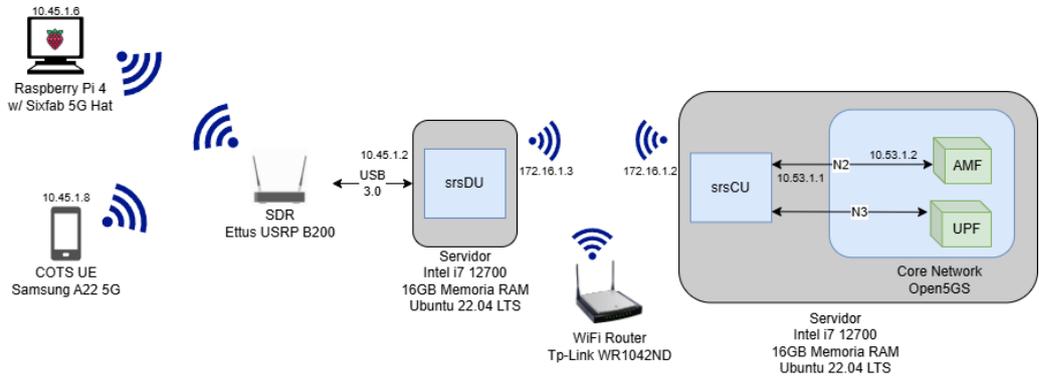


Figura 3.11: Esquema de Red con separación CU/DU conectada mediante Wi-Fi.

la sincronización entre la CU y la DU y propiciar desconexiones indeseadas. Para ello se utilizó un router *Tp-Link WR1042ND* [62]. En la figura Figura 3.11 se puede observar la configuración modificada para este caso.

Una vez completada la configuración, se iniciaron las aplicaciones srsCU y srsDU en sus respectivas máquinas. La DU se conectó al frontend de RF con el SDR B200, mientras que la CU gestionó las interacciones con el núcleo 5G y la DU. Esta configuración permitió establecer una red 5G de extremo a extremo, compatible con la arquitectura O-RAN y el *Split 2*. Se presenta en la figura Figura 3.12 la sección de la DU implementada sobre un NUC. Se resaltan los componentes y se evidencia la diferencia de tamaño requerida para este despliegue.

3.5. Variantes en UE

Como se mencionó en la Sección 3.1, la primera implementación de un UE en este proyecto fue utilizando el *software srsUE* provisto por *srsRAN*. Tanto la facilidad de uso como la compatibilidad con la gNB, como la existencia de documentación del propio proyecto *srsRAN* respecto a cómo utilizarlo en un *testbed* fueron de gran utilidad al usarlo como implementación inicial de un UE 5G.

Para configurar este tipo de UE, los valores usados para la identificación de este en la red, como el IMSI o el par Key/OPC, pueden especificarse directamente en el archivo de configuración con que se ejecuta el UE, dentro de la sección *usim* (ver Subsección 3.2.2). En vez de requerir el uso de una tarjeta SIM, srsUE simula el comportamiento con los identificadores ingresados. Recibe también parámetros que determinan el comportamiento del SDR, como la banda, ancho de banda de la señal y ganancias de transmisión y recepción. Esto permite aplicar diferentes configuraciones de forma ágil entre intentos de conexión consecutivos para hacer una variedad de pruebas.

Sin embargo, el UE presenta diferentes limitaciones por el hecho de haber sido creado inicialmente para su uso en redes LTE, siendo la compatibilidad 5G agregada como extensión posterior. Por esta razón, no se puede usar el srsUE para establecer conexiones en bandas del espectro que usen el esquema *Time Division*

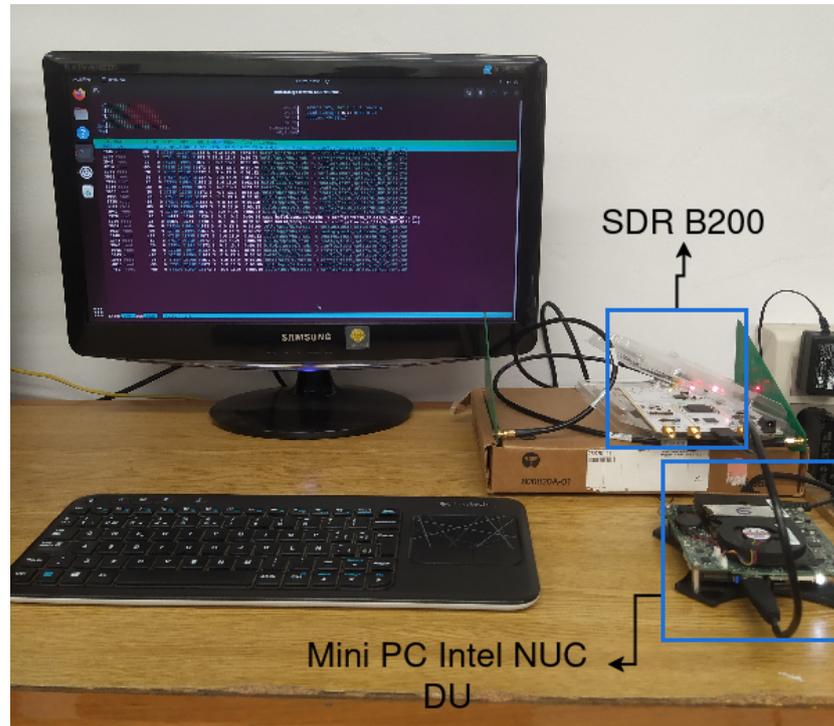


Figura 3.12: Imagen de DU en NUC, utilizando un SDR USRP B200.

Duplex (TDD) ya que es únicamente compatible con el esquema FDD. Tanto esto como otras limitantes observadas en el rendimiento (ver Sección 5.6) llevaron a incluir otros tipos de UEs en las pruebas realizadas.

3.5.1. Dispositivo COTS

Para probar el *testbed* implementado con un dispositivo comercial se utilizó un celular Samsung Galaxy A33 5G, que cuenta con una antena y receptor 5G incorporados. Esto permitió observar la compatibilidad de la maqueta con dispositivos de uso general y extendido.

Para incorporar este dispositivo a la red, se le colocó un SIM obtenido y programado específicamente para este fin (no comercial). El chip fue obtenido del proyecto *Open-Cells*, el cual provee SIMs programables y la herramienta “*UICC/SIM programming*” para configurarlos mediante un dongle USB [63]. Utilizando estas herramientas, se programó el SIM a utilizar en el celular con las credenciales para poder ingresar en la red. En el Apéndice E se detalla el proceso realizado usando esta herramienta. De forma similar al srsUE, se utilizan las credenciales precargadas en el núcleo de red y se puede controlar entonces qué IP se le asignará al UE COTS y qué protocolo IP va a utilizar.

Al ingresar el SIM en el celular, solo hace falta configurar el *Access Point Name* (APN) de la red en el UE, y luego este intentará conectarse a la red con las credenciales configuradas en la SIM. En la Figura 3.13 se puede observar el

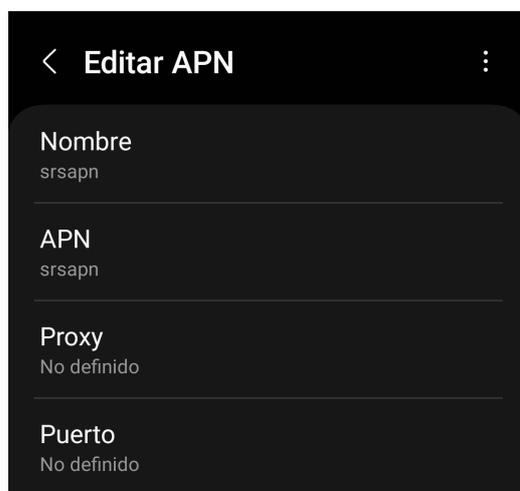


Figura 3.13: Configuración de APN en COTS UE.

panel de configuración que ofrece el equipo. Los valores de **Proxy** y **Puerto** se dejan indefinidos a propósito. Sin embargo, para asegurar que el UE se conecte a la red utilizando los protocolos 5G, se descargó la aplicación “*5G Switch - Force 5G Only*”, que fuerza la conexión a la red en modo 5G SA, evitando el uso de LTE. De esta forma, y luego de un reseteo de fábrica del equipo, se lograron conexiones automáticas a la red que presentaron alta robustez a las variaciones de ganancia y anchos de banda, pudiendo conectarse en todos los escenarios planteados más adelante (ver Capítulo 4).

3.5.2. Dispositivo Sixfab

El tercer tipo de UE utilizado fue el kit de desarrollo *Sixfab 5G*, ya que, como se mencionó anteriormente, este representa un punto medio entre la implementación simulada del srsUE y la implementación comercial y poco modificable del COTS UE.

Se desarrolló una búsqueda de un equipo con estas cualidades y una comparativa detallada (ver Apéndice B) de los dispositivos disponibles en plaza con funcionalidades similares. Dado que el presupuesto asignado al proyecto era limitado, aproximadamente 500 USD, se priorizó encontrar una solución que combinara bajo costo con facilidad de integración y compatibilidad con el entorno de pruebas. En este sentido, la disponibilidad previa de una *Raspberry Pi 4* por parte de un integrante del equipo permitió enfocar la búsqueda en módulos de comunicación 5G compatibles con dicha plataforma.

Además de ajustarse al presupuesto y a los requerimientos técnicos, el uso combinado de una *Raspberry Pi 4* y un módem 5G representa una opción altamente relevante en escenarios de IoT. Este tipo de plataformas embebidas permiten desarrollar dispositivos conectados, de bajo consumo y fácilmente desplegables, ideales para entornos donde la conectividad inalámbrica de alta velocidad, la portabilidad y la eficiencia energética son claves. En este punto, el dispositivo *Sixfab 5G*

3.6. Características y costos finales de la maqueta



Figura 3.14: Kit de desarrollo *Sixfab 5G*, instalado en *Raspberry Pi 4*.

Hat surgió como una alternativa natural.

Para realizar las pruebas diagramadas, se instaló Ubuntu 22.04 en la *Raspberry Pi 4* utilizada. De esta manera, se pudieron hacer tanto pruebas cuantitativas y cualitativas de uso del UE en la red.

En la Figura 3.14 puede verse la instalación del kit *Sixfab*, el cual incluye también una base con ventilador y antenas incorporadas. Estos se conectan al *hat*, el cual contiene también la entrada para ingresar un SIM programado de manera análoga al usado en el COTS UE. Este *hat* se conecta a la *Raspberry Pi* usada mediante los pines de la misma y por un puerto USB.

En este caso, la configuración del UE se realizó mediante la herramienta “*mini-com*”, la cual se usó para configurar el APN de la red usada y corroborar el correcto uso de los parámetros configurados en el SIM. Además, se debieron configurar también los servidores DNS en la interfaz establecida por el *hat*. Los detalles tanto de la instalación como de la configuración del módulo se pueden ver en el Apéndice C.

3.6. Características y costos finales de la maqueta

Dadas las diferentes arquitecturas exploradas en esta sección, se muestra a continuación un resumen de los diferentes equipos usados para implementar cada elemento de la red. Se agregó el costo de los equipos que se consideró pertinente en cada caso, obviando el de equipos de disponibilidad general como las PCs de escritorio. En este caso, el equipo utilizado tuvo un costo de mercado de unos USD 800, pero este puede ser muy variable según los componentes.

En cuanto al núcleo de red, se trabajó en todos los casos con la versión en contenedores *Docker* de *Open5GS*, la cual es de uso libre y gratuito, pudiendo desplegarse en una variedad de equipos de cómputo o incluso en la nube. Por lo tanto, no se asocia un costo a su utilización. Por otra parte, se muestran los costos

Capítulo 3. Maqueta de red 5G

asociados a la gNB y UE desplegados en las tablas 3.1 y 3.2.

Tabla 3.1: Variantes en *hardware* y *software* de radiobase y costos asociados.

Hardware	Software	Costo
PC de escritorio y SDR Ettus X310	srsgNB	USD 10.915 (Ettus X310)
Intel NUC	srsgNB	USD 750
SDR Ettus B200		USD 1.460

Tabla 3.2: Variantes en *hardware* y *software* de equipo de usuario y costos asociados.

Hardware	Software	Costo
PC de escritorio y SDR Ettus x310	srsUE	USD 10.915 (Ettus X310)
PC de escritorio y SDR Ettus b200	srsUE	USD 1.460 (Ettus B200)
Samsung Galaxy A33 5G	Android 11	USD 440
Sixfab 5G Development Kit	Minicom	USD 350
Raspberry Pi 4	Ubuntu 22.04	USD 50

En resumen

A lo largo de este capítulo se documentó detalladamente la implementación de la maqueta 5G desarrollada en el marco del proyecto, partiendo de una arquitectura mínima virtualizada y evolucionando hacia versiones optimizadas en términos de portabilidad, consumo de recursos y desempeño. Se abordaron las distintas configuraciones del núcleo de red, la radiobase y los dispositivos de usuario, así como los procedimientos técnicos para su despliegue en diferentes entornos. Se destacaron además las variantes exploradas, como el uso de mini-PCs, dispositivos comerciales (COTS) y desarrollos intermedios como el kit *Sixfab*.

La diversidad de escenarios probados no solo permitió validar la flexibilidad e interoperabilidad del sistema, sino también sentó las bases para una caracterización precisa de sus requerimientos y una eventual portabilización. Si bien se priorizó el análisis funcional y la estabilidad operativa, se incorporaron también consideraciones prácticas como costos, facilidad de configuración y compatibilidad con equipamiento disponible, aspectos clave para replicar o escalar esta solución en otros contextos. En los siguientes capítulos se presentan las pruebas realizadas y se analizan los resultados obtenidos.

Capítulo 4

Dispositivos, pruebas y resultados preliminares

A lo largo de este proyecto, se realizaron diferentes pruebas en una variedad de arquitecturas posibles para la maqueta implementada, luego de establecer los parámetros para lograr una conexión estable en cada una de ellas. En esta sección se detallan los equipos y procedimientos utilizados en cada etapa, así como las diferentes métricas que se relevaron en cada experiencia. Esto permitirá más adelante tener un conjunto de parámetros con los que se podrán comparar las soluciones implementadas, tanto en el rendimiento de la conexión establecida como en su uso de recursos para guiar el proceso de portabilización.

4.1. Tests de desempeño

Una vez logradas conexiones estables sobre la maqueta interconectada por aire (ver Figura 3.7 y Figura 4.1), se propuso investigar la incidencia de diferentes variables en un entorno controlado. De esta forma, se buscó explorar el desempeño y posibles limitantes de la maqueta basada puramente en *srsRAN* (y *Open5GS*). Para mantener un criterio uniforme de evaluación, se mantuvieron estables las siguientes premisas durante el desarrollo de esta sección de pruebas:

- Utilizar dos SDR USRP X310, tanto para gNB como UE, conectados mediante Ethernet 10 Gbps a los servidores (ver la Figura 4.1 para un ejemplo de esta configuración con los SDRs conectados por la interfaz de aire).
- Trabajar con los mismos servidores en todos los casos, siendo estos los descritos en la Subsección 3.2.2.
- Emplear antenas *Log Periodic Antenna* 850 MHz a 6.5 GHz, dos por SDR (Rx y Tx).
- Conectar un único UE en todos los casos presentados en esta subsección.

Capítulo 4. Dispositivos, pruebas y resultados preliminares

- Realizar los tests en un mismo entorno físico, siendo este el Laboratorio de Telecomunicaciones de la Facultad de Ingeniería (UdelaR), como se ve en la Figura 3.7.

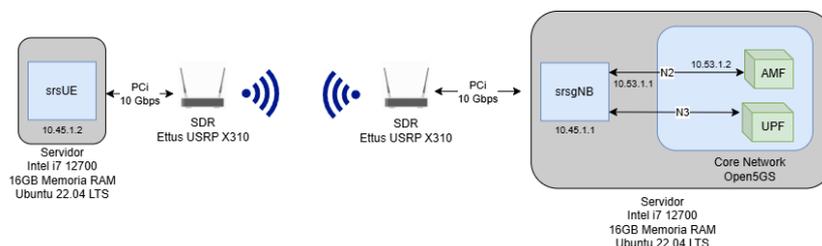


Figura 4.1: Esquema de red con conexión entre SDRs por interfaz de aire, usada para pruebas de rendimiento.

A la hora de evaluar el desempeño de una red móvil existen múltiples métricas que pueden utilizarse dependiendo del objetivo y el contexto de la implementación. Entre las métricas más relevantes se encuentran la latencia, que mide el tiempo de respuesta de la red ante una solicitud; el *jitter*, que evalúa la variabilidad en los tiempos de transmisión de los paquetes; la tasa de error de bits (BER, *Bit Error Rate*), que indica la proporción de bits erróneos respecto al total transmitido; la eficiencia espectral, que calcula la cantidad de datos transmitidos por unidad de ancho de banda; y el *throughput*, que representa la cantidad de datos efectivamente transmitidos por segundo a través de la red. Cada una de estas métricas ofrece información clave sobre distintos aspectos del rendimiento de la red, desde la fiabilidad hasta la capacidad de transmisión y la estabilidad del servicio.

En esta sección, se ha seleccionado el *throughput* como métrica principal para evaluar el desempeño ya que refleja la capacidad real de la red para transportar datos. Su importancia radica en que encapsula el impacto de otras métricas, como la latencia y el BER, en el rendimiento global de la red. Si, por ejemplo, se tiene una latencia alta o una gran cantidad de retransmisiones debido a un alto BER, el *throughput* se verá directamente impactado.

Durante cada uno de estos tests, se empleó la herramienta *Iperf3* para la medición de la tasa de transferencia de datos. Esta es una herramienta ampliamente utilizada en la evaluación del rendimiento de redes, que funciona generando y enviando un flujo continuo de tráfico de datos ficticio entre un servidor y un cliente. En este caso se utilizó *Iperf* para generar tráfico UDP, por lo que el flujo generado se transmite a la máxima velocidad posible y se determina un rango de tiempo durante el que testear una conexión dada. Al finalizar este intervalo la herramienta reporta los valores obtenidos a cada segundo y el promedio total del intervalo de tiempo, permitiendo medir con precisión y de forma real la capacidad de transmisión de datos. Además, su arquitectura ligera y su facilidad de uso permiten la integración en diferentes equipos sin comprometer el *hardware* donde se está ejecutando.

En este caso, se decidió utilizar *iPerf3* en intervalos de 60 segundos para cada experimento. Para esto se estableció un servidor en el equipo que contiene el núcleo,

4.1. Tests de desempeño

y se evaluó la conexión desde un cliente *iPerf3* en el equipo del UE por la subred 10.45.1.0/24.

En todos los casos se compararon los valores de *throughput* obtenidos con los valores teóricos resultantes de la Ecuación 4.1:

$$Data\ Rate = 10^{-6} \cdot \sum_{j=1}^J \left(v_{Layers}^{(j)} \cdot Q_m^{(j)} \cdot f^{(j)} \cdot R_{max} \cdot \frac{N_{PRB}^{BW(j),\mu} \cdot 12}{T_s^\mu} \cdot (1 - OH^{(j)}) \right) \quad (4.1)$$

Esta ecuación es descrita en la especificación [64], la cual calcula la tasa de datos (*data rate*) teórica en Mbps. A continuación se presentan los diferentes términos que la componen y qué representa cada uno:

- $\sum_{j=1}^J$: Esta sumatoria indica el número de portadoras utilizadas en caso de trabajar con *Carrier Aggregation*. En este caso se utiliza $J=1$, es decir, una única componente.
- $v_{Layers}^{(j)}$: Representa la cantidad de capas de transmisión (*layers*) utilizadas. Indica cómo se distribuyen los datos entre las capas disponibles para el usuario j . Al no trabajar con MIMO, este valor también se deja en 1.
- $Q_m^{(j)}$: Indica la modulación utilizada en la transmisión j (por ejemplo, 64-QAM o 256-QAM), lo que afecta directamente la cantidad de bits transmitidos por símbolo. Al trabajar con 64-QAM y un MCS Index de 28, se establece $Q_m^{(j)} = 6$.
- $f^{(j)}$: Corresponde al factor de codificación aplicado, que representa cuántos bits útiles hay respecto al total transmitido (incluyendo bits de corrección). Este se fija en 1.
- R_{max} : Es la tasa máxima de codificación LDPC disponible, límite teórico impuesto por el sistema. Este valor es fijo en 0.926.
- $N_{PRB}^{BW(j),\mu}$: Representa el número de bloques de recursos físicos (PRBs) disponibles en función del ancho de banda y de la numerología μ seleccionada.
- 12: Es la cantidad fija de subportadoras que contiene cada PRB.
- T_s^μ : Indica el tiempo de duración de un símbolo OFDM bajo la numerología μ , afectando así la tasa de transmisión.
- $1 - OH^{(j)}$: Es el factor que refleja la fracción de recursos útiles disponibles, luego de descontar el *overhead* de señalización. Dicho *overhead* valiendo 0.14.

A continuación se presentan los valores preliminares obtenidos, separados en cuatro categorías, en las que se varió un parámetro fundamental de la configuración en cada caso y se exploró su influencia sobre el *throughput* de la red. Es importante destacar que de aquí en más se utilizarán los términos *throughput*, *data rate* o *bitrate* de forma indistinta.

4.1.1. Ganancias en antenas

Se probaron diferentes niveles de ganancia en transmisión y recepción en ambos equipos durante el proceso de implementación para lograr una conexión estable entre el gNB y el UE, manteniendo el resto de los parámetros fijos. Este fue el primer conjunto de pruebas realizado, ya que era necesario determinar los rangos de ganancia en los que la conexión funcionaba correctamente antes de poder variar otros parámetros.

Identificados los rangos de valores en los que la conexión era suficientemente estable como para soportar varios minutos de tráfico, se procedió a iterar, variando de a 5 dB y registrando solo los valores en los que se lograba establecer conectividad, resultando en los valores expuestos en la Tabla 4.1. Se realizó un promedio de múltiples pruebas, cada una con un intervalo de 60 segundos. En cada caso, se utilizaron valores idénticos de ganancias en los equipos del UE y del gNB.

Además del análisis de *throughput* explicado anteriormente, se empleó un comando de monitoreo incluido en el proyecto *srsRAN* para analizar la potencia de las señales transmitidas, el cual permite evaluar la potencia promedio de la señal transmitida y recibida por el gNB, la potencia pico en ambos casos y el *Peak to Average Power Ratio* (PAPR) en transmisión y recepción. Este último se consideró una medida adecuada no solo de la potencia de las señales transmitidas, sino también de su relación señal a ruido (*Signal to Noise Ratio*, SNR) en relación con el ruido de fondo presente durante el experimento (ver Tabla 4.1).

4.1.2. Ancho de banda

En segunda instancia, se decidieron investigar posibles variaciones de *throughput* y de rendimiento en función del ancho de banda del canal utilizado. Se destaca que al modificar el ancho de banda establecido en la gNB, mediante el parámetro `channel_bandwidth_MHz`, también se deben modificar las tasas de transmisión `srate` y los valores de PRB en el UE para indicar a ambos equipos que se conecten en el nuevo ancho de banda establecido. Las frecuencias de muestreo a utilizar son determinadas por la implementación de *srsRAN*. El *software* establece qué `srate` se debe introducir para trabajar a un ancho de banda de canal determinado.

Si bien *srsRAN Project* permite utilizar anchos de banda mayores, en este caso se pudo trabajar hasta 20 MHz, siendo la limitante el srsUE. Este valor es el recomendado en el caso base de conexión propuesto en la documentación de *srsRAN* [44]. Para estos *test*, se propuso bajar estos anchos de banda, evaluando así

Tabla 4.1: Resultados de pruebas de ganancia por interfaz de aire utilizando SDR USRP X310.

Tx (dB)	Gain	Rx (dB)	Gain	Bitrate DL (Mb/s)	Bitrate UL (Mb/s)	Tx PAPR (dB)	Rx PAPR (dB)
10		5		3.31	3.39	23.5	13.7
20		5		3.08	3.17	23.0	12.0
20		10		3.20	3.32	25.8	10.0

4.1. Tests de desempeño

Tabla 4.2: Valores de los parámetros afectados al modificar el *Bandwidth*.

Bandwidth (MHz)	Sample Rate (MHz)	Number of PRBs
20	23.04	106
15	15.36	79
10	11.52	52
5	5.76	25

Tabla 4.3: Resultados de pruebas de ancho de banda por interfaz de aire utilizando SDR USRP X310.

Bandwidth (MHz)	\overline{Br}_{DL} (Mbits/s)	\overline{Br}_{UL} (Mbits/s)	Valor Teórico _{DL} (Mbits/s)	Valor Teórico _{UL} (Mbits/s)
20	5.61	5.69	85.07	91.00
15	4.88	4.97	63.40	67.82
10	4.04	4.11	41.73	44.64
5	1.78	1.84	20.06	21.46

el impacto en las velocidades de transmisión y el consumo de recursos de *hardware*. Los anchos de banda propuestos para la comparación fueron 5, 10, 15 y 20 MHz. En la Tabla 4.2 se establecen las configuraciones que se debieron adaptar de los parámetros afectados.

Durante estas pruebas se trabajó con ganancias de 10 dB en transmisión y 5 dB en recepción, tanto en gNB como en UE. Se transmitió en la banda 3, centrados en la frecuencia 1842.5 MHz, con un SCS de 15 kHz. A su vez, las antenas estuvieron ubicadas en posiciones idénticas, a 0.5 m de distancia. Se utilizó Iperf3 con UDP para saturar la red de tráfico y evaluar el *data rate* promedio que alcanzaban cada una de las configuraciones en 60 segundos de transmisión.

El uso de UDP radicó en que si se utilizara TCP, este introduce mecanismos de control de congestión y recuperación de errores que pueden influir en la estabilidad de la medición y en la detección de pérdidas. Con UDP es posible hacer una medición más directa del rendimiento bruto. Las pruebas se realizaron en modo *full-duplex*, es decir, con tráfico bidireccional simultáneo. Esta decisión permite simular saturación máxima en ambos sentidos y observar el comportamiento del *throughput* bajo diferentes niveles de exigencia en paquetes por segundo (pps).

En la Tabla 4.3 se pueden observar los resultados. En forma de comparación, se agregan los valores teóricos que deberían alcanzarse según la Ecuación 4.1. Como se puede observar, los *bitrates* alcanzados inicialmente por la maqueta se encuentran muy lejanos al límite teórico. Si bien no se espera alcanzar estos valores ya que son modelados en condiciones iniciales, los resultados están por debajo del 10% del ideal en todos los casos. Posibles causas de esto se analizarán más adelante en la Sección 5.3.

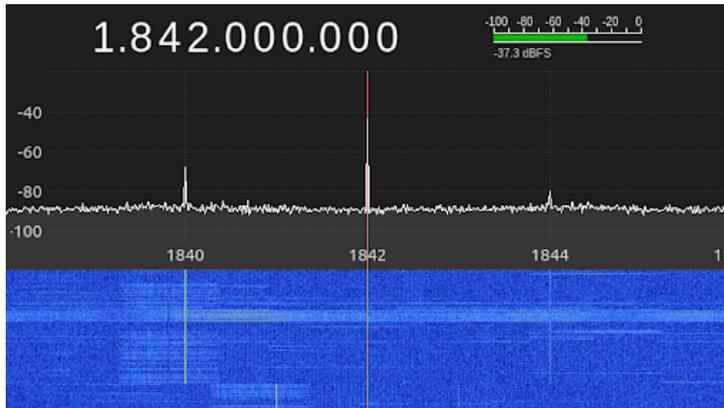


Figura 4.2: Espectro de banda $n3$, centrado en 1842 Mhz.

4.1.3. Bandas del espectro

Posteriormente, se exploró el comportamiento de la conexión entre el gNB y el UE en diferentes bandas del FR1, dentro de aquellas soportadas por *srsRAN*. Como se explicó anteriormente, el UE es compatible únicamente con bandas FDD y admite tasas de muestreo de hasta 20 MHz.

Para especificar la banda deseada en cada caso, se utilizó el parámetro `band` en ambos equipos. Además, mediante el parámetro `dl_arfcn` en la gNB, fue posible seleccionar una portadora específica para la transmisión en *Downlink*. Al ejecutar el programa asociado a la radiobase, este proporciona el valor de ARFCN correspondiente a la transmisión en *Uplink*, el cual, en el caso del UE implementado en *srsUE*, debía ser configurado mediante el parámetro `dl_ssb_arfcn` antes de iniciar la ejecución.

El primer conjunto de pruebas se realizó dentro de la banda $n3$, que ya había sido utilizada previamente. Antes de la ejecución en los equipos, se llevó a cabo un análisis espectral para evaluar el nivel de ruido de fondo en distintos puntos dentro de esta banda. En la Figura 4.2 se presenta una captura del espectro observado, centrado en 1842.5 MHz, equivalente a un valor de ARFCN 368400. En este caso, se tuvo como referencia un piso de ruido de aproximadamente -90 dBm, el cual se considera razonable para un entorno de laboratorio. Utilizando la fórmula de **Johnson-Nyquist** ($N = kTB$, donde k es la constante de Boltzmann (1.3810^{-23}), T la temperatura en grados Kelvin (290K), y B el ancho de banda 20 MHz), se obtiene un valor teórico de aproximadamente -101 dBm. La diferencia puede atribuirse a la presencia de ruido adicional de equipos eléctricos cercanos y la ganancia del receptor.

Se procedió entonces a variar la frecuencia de la portadora mediante el ajuste del valor de ARFCN, nuevamente haciendo pruebas separadas de longitud de 60 segundos y obteniendo así el rendimiento promedio en cada intervalo. Los resultados se presentan en la Tabla 4.4.

Adicionalmente, se realizaron pruebas en otras bandas además de la $n3$. Para ello, se seleccionó un conjunto representativo de bandas asignadas a distintos pro-

4.1. Tests de desempeño

Tabla 4.4: Resultados de pruebas en banda n3.

dl_arfcn / f_{DL} (MHz)	dl_ssb_arfcn / f_{UL} (MHz)	Ruido DL / UL (dB)	\overline{Br}_{DL} / \overline{Br}_{UL} (Mbits/s)
368500 / 1842.5	368410 / 1747.5	-65 / -100	3.34 / 3.61
364000 / 1820.0	363850 / 1725.0	-65 / -75	5.94 / 6.21
361200 / 1806.0	361230 / 1711.0	-110 / -100	5.11 / 5.37

Tabla 4.5: Resultados en bandas asignadas a proveedores.

Banda	dl_arfcn / f_{DL} (MHz)	dl_ssb_arfcn / f_{UL} (MHz)	Proveedor asignado	$\overline{Bitrate}_{DL}$ / $\overline{Bitrate}_{UL}$ (Mbits/s)
n3	368500 / 1842.5	368410 / 1747.5	Antel	3.34 / 3.61
n1	432000 / 2160.0	432030 / 1970.0	Claro	4.37 / 4.61
n5	390000 / 1950	390030 / 1870	Movistar	3.87 / 4.15
n8	187000 / 935	186970 / 890	Sin asignar	16.4 / 16.7

veedores de servicios de telefonía en Uruguay, de acuerdo con la asignación de la Unidad Reguladora de Servicios de Comunicaciones (URSEC) [65]. Estos resultados se presentan en la Tabla 4.5

4.1.4. Cobertura

Dado que el sistema presentó una notable sensibilidad a las ganancias de transmisión y recepción, resultó de interés evaluar su desempeño a diferentes distancias entre el gNB y el UE. Para ello, se definieron tres escenarios dentro del mismo entorno físico previamente detallado. El primero corresponde a la configuración utilizada en el resto de las pruebas, con las antenas enfrentadas a 0.5 m. Naturalmente, esta distancia es muy reducida y no representa un caso realista en escenarios prácticos. Por lo tanto, se decidió explorar el límite de cobertura posible.

Para analizar si dicha separación se encuentra dentro del campo lejano, se aplicó la fórmula de Rayleigh $R = \frac{2D^2}{\lambda}$ donde D representa la dimensión mayor de la antena transmisora y λ la longitud de onda correspondiente a la frecuencia central utilizada. En este caso, se utilizó una antena log-periódica WA5VJB (850 MHz a 6.5 GHz), con un largo aproximado de 16 cm ($D = 0.16$ m). A una frecuencia central de 3.5 GHz ($\lambda \approx 0.0857$ m), el límite inferior del campo lejano se estima como:

$$R = \frac{2 \cdot (0.16)^2}{0.0857} \approx 0.6 \text{ m} \quad (4.2)$$

Por tanto, los 0.5 metros utilizados se encuentran en el límite entre el campo cercano lejano y el campo lejano. Si bien no se alcanzó plenamente la región de propagación ideal del campo lejano, se consideró una distancia aceptable para pruebas funcionales de conectividad en entorno de laboratorio, manteniendo estabilidad en las condiciones de recepción.

Capítulo 4. Dispositivos, pruebas y resultados preliminares

Tabla 4.6: Valores de ganancias utilizados según la distancia.

Distancia (m)	gNB Tx Gain (dB)	gNB Rx Gain (dB)	UE Tx Gain (dB)	UE Rx Gain (dB)
0.5	10	5	10	5
1.0	10	10	10	5
3.0	20	10	20	10

Tabla 4.7: *Bitrates* según la distancia en 60 segundos de transmisión, utilizando SDR USRP X310.

Distancia (m)	$\overline{Bitrate}_{DL}$ (Mbits/s)	$\overline{Bitrate}_{UL}$ (Mbits/s)
0.5	3.39	3.31
1.0	3.05	2.94
3.0	3.49	3.16

En los siguientes experimentos se incrementó progresivamente la separación entre las antenas y se ajustaron las ganancias para mantener la conexión. Los radios de cobertura finales alcanzadas fueron entre 1 m y 3 m. Más allá de este punto, las conexiones resultaron altamente inestables (con duraciones de apenas unos segundos) o directamente inviables. En la Tabla 4.6 se presentan las ganancias mínimas de transmisión y recepción utilizadas tanto en el gNB como en el UE para cada uno de los escenarios. Estas configuraciones fueron determinadas mediante pruebas experimentales de conexión. Para todas las pruebas, se utilizó la configuración por defecto con un ancho de banda de 20 MHz, un sample rate de 23.04 MHz y una frecuencia central de 1842.5 MHz en la banda *n3*.

En esta instancia, la conexión presentó una mayor inestabilidad a medida que se incrementaba la distancia entre las antenas. Las pruebas de *Iperf* se realizaron con una duración de 60 segundos, como en todos los casos anteriores. Sin embargo, debido a la alta tasa de desconexiones, fue necesario repetir las varias veces para llegar a conexiones estables, tomando entonces varias mediciones por distancia y registrando en cada una un caso tipo de conexión exitosa. En la Tabla 4.7 se presentan los valores del *throughput* alcanzado en cada caso.

4.1.5. Variación de SDR

Luego de finalizar las diferentes pruebas de rendimiento, se propuso la implementación de la maqueta con otros equipos de RF, siendo los USRP B200 de *Ettus Research* los disponibles. Estos equipos se conectan mediante USB 3.0 al servidor, y trabajan utilizando las bibliotecas de *drivers* UHD al igual que los X310. En la página del fabricante [55] se pueden encontrar sus especificaciones, siendo las más relevantes para esta implementación su cobertura RF de entre 70 MHz y 6 GHz, y la capacidad de operar en *Full Duplex* con hasta 56 MHz de ancho de banda en tiempo real.

En esta etapa se logró implementar con éxito la maqueta utilizando los USRP

4.1. Tests de desempeño

Tabla 4.8: Ganancias utilizadas para las conexiones con SDR USRP B200.

gNB Tx Gain	gNB Rx Gain	UE Tx Gain	UE Rx Gain
70	30	60	40

Tabla 4.9: Resultados de pruebas con SDR USRP B200.

Medio	Bandwidth (MHz)	\overline{Br}_{DL} (Mbits/s)	\overline{Br}_{UL} (Mbits/s)	Valor Teórico _{DL} (Mbits/s)	Valor Teórico _{UL} (Mbits/s)
Coaxial	20	67.1	67.2	85.07	91.00
Aire	20	20.8	20.9	85.07	91.00
Aire	15	12.1	12.4	63.40	67.82
Aire	10	7.9	8.2	41.73	44.64
Aire	5	3.1	3.3	20.06	21.46

B200, con mínimas adaptaciones a los archivos de configuración. Los parámetros a modificar fueron:

- **device_args:** En gNB este parámetro debió recibir los siguientes argumentos para el correcto funcionamiento, modificando el tamaño de la trama a enviar y especificando el dispositivo en la gNB:
`type=b200, send_frame_size=1472, recv_frame_size=1472`
- **freq_offset:** En la UE se debió compensar el offset en frecuencia manualmente. Este valor puede variar, ya que depende de la sincronización entre ambos SDR. Al no utilizar un reloj de referencia, una vez inicializados los programas correspondientes a la gNB y UE e intentada una primera conexión, se observan los logs para encontrar el valor del *cfo* y luego compensarlo desde la configuración.
- **tx_gain, rx_gain:** Tanto en gNB como en UE se debieron cambiar las ganancias. Los valores a utilizar se hallaron experimentalmente, resultando los expuestos en la Tabla 4.8 los adecuados para transmisiones estables.

Para probar el *throughput* de las transmisiones con estos equipos, se utilizó *iPerf3*, al igual que con los USRP X310. Además, se trabajó en las bandas por defecto explicadas anteriormente. Con estas configuraciones, se obtuvieron los resultados expuestos en la Tabla 4.9 para los *tests de throughput* en diferentes anchos de banda. Se hicieron pruebas tanto por conexiones mediante aire como mediante cable coaxial conectados directamente, con el fin de observar cómo afecta el medio en este caso. Como era de esperarse, el valor alcanzado con la conexión mediante cable coaxial fue superior a la inalámbrica, evidenciando una cercanía a la capacidad máxima teórica. Se recuerda que este cálculo es independiente del medio.

Capítulo 4. Dispositivos, pruebas y resultados preliminares

Tabla 4.10: Resultados de *data rate* DL y UL para COTS UE y Sixfab en diferentes *Bandwidths*.

Dispositivo	BW(MHz)	\overline{Br}_{DL} (Mbits/s)	\overline{Br}_{UL} (Mbits/s)
COTS UE	20	68.6	69.1
	15	51.4	52.0
	10	32.0	32.5
	5	13.8	14.2
Sixfab	20	63.0	63.2
	15	39.7	40.1
	10	22.3	22.2
	5	11.0	10.8

4.2. Escenarios de prueba

En esta siguiente sección se explican las pruebas realizadas en cada una de las arquitecturas mencionadas en el capítulo anterior. Se buscó inicialmente lograr una conexión estable entre los diferentes elementos de la red de forma de poder pasar a una etapa de *tests*, como los que se detallaron en la última sección con srsUE. A partir de esto, se pudo sumar un conjunto de pruebas adicionales, como el uso de múltiples UEs o del *split* funcional 2. Una vez lograda la conectividad en cada uno de estos casos, se realizaron *tests* de desempeño individuales para medir el éxito obtenido en cada caso.

Antes de evaluar el desempeño en distintos escenarios, se realizó un registro del *throughput* alcanzado por cada uno de los nuevos UE con los que se está trabajando. Se estudió el desempeño de cada uno para los cuatro anchos de banda disponibles, en una misma banda libre de ARFCN de *Downlink* 187000. Los resultados registrados se presentan en la Tabla 4.10, sirviendo como punto de referencia para el análisis de desempeño en diferentes configuraciones.

En el capítulo posterior se analizarán estos resultados junto con los observados para el srsUE, en la Subsección 5.6.1. Sin embargo, ya es posible observar directamente con los resultados presentados por el COTS UE y el kit *Sixfab* que estos presentan un rendimiento considerablemente mejor. Este rendimiento a su vez es consistente con el comportamiento esperado al ir variando el ancho de banda.

Dadas estas observaciones, ya fue posible determinar que estos UEs serían una herramienta más confiable a utilizar en pruebas que evalúen el rendimiento de otros elementos de la red, como los detallados en la Subsección 4.2.2. De igual manera, se realizaron pruebas adicionales con los diferentes UEs para poder examinar su funcionamiento en simultáneo.

4.2.1. Tráfico concurrente

Luego de probar la funcionalidad de cada UE individual, se realizaron pruebas con diferentes permutaciones de equipos de usuarios conectados a la red en simultáneo. Así se logró evaluar el funcionamiento de la radiobase implementada en

4.2. Escenarios de prueba

una conexión con múltiples UEs, así como observar y comparar el comportamiento de cada uno al compartir una red con más equipos de usuario.

Se utilizó *Iperf3* para medir el *throughput* obtenido por cada UE en la red durante las pruebas con el fin de estudiar cómo se asignan los recursos disponibles a cada uno según sus requerimientos. Esto permite observar los efectos de los diferentes protocolos de manejo de tráfico aplicados sobre la red, evaluando la proporción entre el *throughput* destinado a cada UE y la capacidad neta de la red.

La configuración del núcleo de red y de la radiobase se mantuvo estable de forma de variar únicamente los UEs conectados. De esta forma, se utilizó el mismo servidor y los SDR B200, transmitiendo en la banda *n28* y manteniendo fijas las ganancias en 70 dB en transmisión y 30 dB en recepción. Se realizaron *tests* de *Iperf3* entre cada uno de los UE y la dirección del núcleo de red, en forma simultánea, con una duración de 60 segundos. Las permutaciones estudiadas fueron entonces:

- srsUE y COTS UE
- srsUE y UE Sixfab 5G
- COTS UE y UE Sixfab 5G
- Los 3 UEs en simultáneo

En la Tabla 4.11 se presentan los resultados para cada una de las combinaciones estudiadas. Puede verse que los UE COTS y *Sixfab Hat 5G* llegan a rendimientos similares entre sí en todos los casos estudiados, mientras que el srsUE parece “ocupar” una mayor proporción del canal respecto a los otros. Este comportamiento se analiza en profundidad en la Subsección 5.6.1, donde también se estudia el comportamiento del srsUE en estos casos en comparación a las instancias en que fue utilizado como único UE de la red, sumando los resultados ya presentados en la sección anterior.

4.2.2. Desempeño con Split 2

La siguiente batería de pruebas realizadas se utilizó para medir el rendimiento de la arquitectura desplegada separando la radiobase en procesos que ejecutan CU y DU de forma independiente.

Para esto se aplicó el proceso inverso a las pruebas de la sección anterior, fijando un único UE de prueba y variando la configuración de la gNB. El UE utilizado fue el COTS UE, considerando su facilidad de uso y configuración y el alto *bitrate* alcanzado. Este cuenta con herramientas suficientes para relevar el rendimiento de la nueva configuración. De esta forma el rendimiento del UE no sería una limitante en estas pruebas dedicadas a monitorear la gNB.

Como se observó en la Subsección 3.4.2, se presenta la limitante de que la interfaz entre el NUC y el SDR utilizado no es estable al superar los 5 MHz de ancho de banda, por lo que se restringieron las pruebas a este ancho de banda. Así,

Capítulo 4. Dispositivos, pruebas y resultados preliminares

Tabla 4.11: Resultados de *throughput* con múltiples equipos de usuario en simultáneo.

(a) COTS UE y Sixfab 5G.		
UE	\overline{Br}_{DL} (Mbps)	\overline{Br}_{UL} (Mbps)
COTS UE	29.3	28.5
Sixfab	27.9	27.5

(b) srsUE y Sixfab 5G.		
UE	\overline{Br}_{DL} (Mbps)	\overline{Br}_{UL} (Mbps)
srsUE	14.4	14.1
Sixfab	8.92	8.52

(c) COTS UE y srsUE.		
UE	\overline{Br}_{DL} (Mbps)	\overline{Br}_{UL} (Mbps)
COTS UE	15.6	14.9
srsUE	35.1	34.8

(d) Sixfab 5G, COTS UE y srsUE.		
UE	\overline{Br}_{DL} (Mbps)	\overline{Br}_{UL} (Mbps)
Sixfab	9.43	9.12
COTS UE	19.8	19.2
srsUE	29.1	19.9

se acotaron las comparaciones a los parámetros en que se pueden ejecutar todas las arquitecturas planteadas.

De esta manera, se pudieron establecer los diferentes escenarios descritos en la Subsección 3.4.2. En estos se fija la ejecución del CU en la PC servidor que ejecuta a su vez el núcleo de red, y se varía la ubicación del DU y la modalidad de la interfaz *F1AP* que comunica ambos procesos de las siguientes maneras:

1. Los dos procesos para ejecutar CU y DU corriendo en una misma PC, con la interfaz *F1AP* implementada de forma virtual en un puerto de la interfaz *localhost* de la PC.
2. Ejecutar el DU en una segunda PC, conectando CU y DU mediante un cable Ethernet de 2.5 Gbps.
3. Ejecutar el DU nuevamente en una segunda PC, conectando CU y DU mediante una red Wi-Fi local compartida.
4. El DU ejecutando en un Intel NUC, con la interfaz *F1AP* establecida sobre el cable de red Ethernet de 2.5 Gbps.

4.2. Escenarios de prueba

Tabla 4.12: *Throughput* relevado en las diferentes interfaces *F1AP*.

DU	interfaz <i>F1AP</i>	\overline{Br}_{DL} (Mbps)	\overline{Br}_{UL} (Mbps)
PC	virtual	13.1	12.9
PC	Ethernet	13.0	12.3
PC	Wi-Fi	10.7	9.8
NUC	Ethernet	11.3	11.1
NUC	Wi-Fi	13.2	12.4

5. El DU ejecutado nuevamente en un Intel NUC, pero estableciendo en este caso la interfaz *F1AP* por una conexión en una red Wi-Fi local.

Estas variaciones en la interfaz se realizaron con el fin de representar escalas cada vez más cercanas al grado de portabilización buscado. Además, al desplegar la CU y DU en dos equipos diferentes se está implementado un *Split 2*, el cual fue probado con diferentes enlaces.

Por otra parte, el despliegue de la DU en *hardware* de menor porte permitiría ejecutarlo en instancias más livianas y con menor costo. A su vez, al realizar estas pruebas usando una PC en vez del Intel NUC a modo de control permiten detectar la presencia de limitaciones creadas al introducir equipos de menor escala, pero no necesariamente asociadas a la interfaz *F1AP* elegida.

En la Tabla 4.12 se presentan los resultados obtenidos. En este caso no se observaron diferencias significativas en los *bitrate* obtenidos, evidenciando que una interconexión CU/DU inalámbrica es de hecho viable. Esto se analizará en profundidad en la Subsección 5.6.2.

4.2.3. Conexión a Internet

Luego de completar todas las pruebas de desempeño de la red, se evaluó la posibilidad de establecer una conexión a Internet utilizando la infraestructura implementada (es decir, que el UE se conecte a internet a través del núcleo de la red pasando por los diferentes segmentos de la red).

Para evaluar esta conexión, se realizaron mediciones de velocidad mediante *Speedtest by Ookla*, tanto desde su versión web como utilizando su servicio para la línea de comandos de Linux (CLI). En esta prueba, se busca evaluar el rendimiento en un escenario de acceso del usuario a servicios externos a la red, a través de las funciones provistas por el *core* 5G. Se observó que el funcionamiento de la red Wi-Fi (en este caso la red abierta de la Facultad de Ingeniería) a la que se conectaba la PC servidor y por consiguiente el núcleo de red podía llevar a variaciones en los resultados obtenidos entre diferentes días según el nivel de congestión, por lo que se realizaron diferentes instancias de pruebas con el fin de llegar a condiciones estables entre los diferentes equipos de usuario.

Como referencia, se realizó un *Speedtest* directamente desde la PC servidor, obteniendo un *throughput* de 25.3 Mbps en descarga y 28.9 Mbps en subida, con un ping de 40 ms. Nuevamente, estos valores presentaron grandes variaciones según el

Capítulo 4. Dispositivos, pruebas y resultados preliminares

Tabla 4.13: Resultados del Speedtest para diferentes dispositivos y anchos de banda.

Dispositivo	Ancho de Banda (MHz)	Throughput DL (Mbps)	Throughput UL (Mbps)	Ping (ms)
Sixfab	20	20.38	29.52	53
	15	11.18	9.61	39
	10	12.44	11.89	31
	5	10.84	6.98	39
COTS UE	20	17.60	37.60	8
	15	18.10	5.50	45
	10	10.91	30.90	8
	5	1.45	12.22	9
srsUE	20	3.83	8.18	117
	15	8.41	26.93	58
	10	11.93	27.18	39
	5	8.61	14.15	73

momento en que se realizaran, probablemente debido a la cantidad de tráfico que estuviese experimentando la red pública. En la Tabla 4.13, se presentan los valores alcanzados por los diferentes dispositivos de usuario en la misma instancia en que se tomaron los valores de referencia presentados. Se exponen los datos obtenidos para los diferentes anchos de banda a los que se pudo ejecutar la maqueta con el fin de observar el efecto de cambiar este parámetro y contrastarlo con las mediciones obtenidas con *Iperf3*.

Pruebas cualitativas

Habiendo estudiado la implicancia de diferentes variables en el *throughput* alcanzado tanto en tráfico interno como externo, como última instancia se exploró el desempeño de esta maqueta en aplicaciones reales. Para eso, tomando como premisa los comportamientos observados en etapas previas, se determinaron tres casos de uso básicos:

- Descarga de paquetes *apt* en Ubuntu
- Visualización de contenido multimedia en línea
- Establecimiento de videollamada entre dos usuarios

Estos ejemplos se seleccionaron ya que comprenden diferentes tipos de consumo de datos y una gama acotada pero diversa de posibles usuarios. El primer caso consistió en la descarga de paquetes desde los repositorios globales *apt*, lo que permitió observar el comportamiento de la conexión en la transferencia de datos tradicional, con solicitudes HTTP y flujos de descarga que no presentan grandes variaciones. Para el siguiente caso se trabajó con la plataforma *Youtube*, la que representó un escenario intermedio en términos de demanda de red. A diferencia de una descarga de paquetes APT, *YouTube* emplea técnicas de *buffering* y adaptación dinámica de calidad (ABR, *Adaptive Bitrate Streaming*), lo que le permite compensar fluctuaciones momentáneas en el *throughput*, pero con una demanda de *bitrate* mucho mayor. Por último, las videollamadas constituyen el escenario

Tabla 4.14: Resultados de pruebas cualitativas.

Descarga de paquetes en Ubuntu (apt-get install)	
Referencia: Servidor	160 kB/s
COTS UE	No fue posible*
Sixfab	39.9 kB/s
SrsUE	24.3 kB/s

*Sistema operativo no lo permite

más demandante, ya que requieren una comunicación en tiempo real, donde tanto la latencia como la estabilidad del ancho de banda son críticas. En estas el tráfico es bidireccional y debe mantenerse sin interrupciones para evitar cortes en la comunicación. Además, las videollamadas utilizan códecs de compresión con baja tolerancia a la pérdida de paquetes y requieren baja latencia para asegurar que la conversación fluya sin retardos perceptibles. Se observó que este último caso era límite, ya que la aplicación tuvo serios problemas a la hora de su uso.

Es importante destacar en este punto que las pruebas realizadas utilizaron aplicaciones con conexión a servidores externos, por lo que estas experiencias sirvieron además para evaluar el funcionamiento del *gateway* de conexión a Internet. De todos modos, al tratarse de una maqueta *End-To-End*, es posible utilizar servicios similares que no tengan necesidad de usar esta salida a Internet. A modo de ejemplo, si un servicio se implementa en el mismo equipo en que se ejecuta el núcleo de red, los UEs contarán entonces con el rendimiento observado en las pruebas anteriores que evalúan la conexión dicho servidor.

Si bien se tomaron datos en los casos en que fue posible, el análisis de estas aplicaciones se realizará de manera cualitativa, dado que su comportamiento es difícil de comparar numéricamente. Se optó por un enfoque subjetivo, centrado en evaluar la percepción del usuario durante el uso de cada aplicación, identificando dificultades evidentes, tiempos de respuesta y estabilidad general de la conexión. Esta aproximación permite tener una idea del desempeño de la red en casos de uso reales. En la Tabla 4.14 se presentan los valores obtenidos para los casos de la descarga de paquetes *apt*, ya que este es el caso en que se puede observar un valor más cuantitativo para analizar el rendimiento.

Con el fin de igualar las condiciones, en el caso de la descarga *apt*, se descargó el mismo paquete, y en el caso del contenido multimedia se visualizó el mismo video en la misma plataforma. Todas las pruebas fueron realizadas en un solo día. Es importante notar que con el COTS UE no era posible descargar el paquete *apt* ya que su sistema operativo (impuesto por el fabricante) no lo permite.

En el caso de la visualización de contenido en *Youtube*, se observó que el COTS UE no presentaba problemas para usar la aplicación correspondiente en el sistema Android y mantenía un flujo de datos constante que permitía utilizarla sin problemas. Por otro lado, el UE *Sixfab* y el *srsUE* presentaron diferentes problemas. En el caso del *srsUE*, este no fue capaz de mantener una conexión estable al des-

Capítulo 4. Dispositivos, pruebas y resultados preliminares

cargar los datos correspondientes, causando desconexiones cada ciertos intervalos que requerían una reconexión manual. El UE *Sixfab* presentó un problema inverso a este, ya que se observó que se mantenía un flujo de datos constante durante el uso pero no era posible usar la aplicación del navegador correctamente en la *Raspberry Pi*. Por lo tanto, no se presentarán los resultados cuantitativos obtenidos ya que en ninguno de los casos se logró utilizar la aplicación.

Este comportamiento se repitió en las videollamadas propuestas entre UEs, en las que se observó que el mejor funcionamiento se daba con el COTS UE. Nuevamente, el UE *Sixfab* mantuvo una conexión estable a la red, pero no se pudo usar correctamente la aplicación de *Zoom* en el navegador. Esto se observó claramente al hacer una videollamada entre el COTS UE y el UE *Sixfab*, en que la ejecución del navegador *Firefox* en la *Raspberry Pi* con Ubuntu 22.04 generaba retardos en la ejecución que impactaban en la calidad de la llamada vista por el usuario del UE *Sixfab*, mientras que en el COTS UE no se observaban problemas. En el caso del srsUE este no tuvo problemas de *hardware* que impidieran la ejecución, pero volvió a presentar desconexiones a la red durante la llamada.

En el siguiente capítulo se analiza este comportamiento y el funcionamiento de cada equipo en general, en base a los resultados presentados a lo largo de este.

Capítulo 5

Análisis de resultados

En este capítulo se presentan los resultados obtenidos a lo largo del desarrollo del proyecto. Durante el proceso de implementación y prueba de la maqueta, el equipo de trabajo fue tomando familiaridad con el proyecto *srsRAN*. A medida que se profundizó en el uso del *software*, se identificaron tanto sus capacidades como sus limitaciones. Se analizan las mediciones realizadas en distintos escenarios, evaluando el desempeño de la red en diferentes aspectos y configuraciones, y comparándolo con su competidor *OpenAirInterface*. Además, se incluyen observaciones cualitativas sobre el comportamiento de la maqueta en aplicaciones de uso real.

5.1. Comparativa teórica con *OpenAirInterface*

Inicialmente se comenzó por realizar un relevamiento teórico, creando un cuadro comparativo que presenta las principales diferencias en las prestaciones declaradas por *srsRAN* y *OAI*. En este se exponen tanto generalidades como aspectos técnicos de la *RAN* (ver Tabla 5.1). Esta comparación teórica facilitó la interpretación de los resultados experimentales, ya que permite contextualizar las limitaciones observadas en las pruebas con respecto a las características propias de cada solución.

De este cuadro es posible observar varios aspectos clave:

- *srsRAN* no cuenta con un UE 5G nativo, pero es posible utilizar *srsUE 4G* y conectarlo a la red de 5ta generación.
- *srsRAN* soporta una mayor variedad de anchos de banda. Sin embargo, al no contar con una solución de UE para 5G, no es posible aprovechar los mayores a 20 Mhz si no se utilizan equipos de terceros.
- *OAI* cuenta con un Core 5G propio, facilitando una implementación más autónoma, mientras que *srsRAN* utiliza el desarrollado por *Open5GS*.
- Ambos soportan tanto FDD y TDD. Sin embargo, nuevamente el *srsUE 4G* limita el uso de TDD en la solución estudiada.

Capítulo 5. Análisis de resultados

Tabla 5.1: Comparativa entre *srsRAN* y *OpenAirInterface*.

	srsRAN Project	OpenAirInterface
Generalidades		
Versión actual	24.10	2024.w49
gNB SA y NSA	Sí	Sí
UE SA y NSA	No. Utiliza srsUE 4G	Sí
Core Network	No. Utiliza Open5GS	Sí
RAN		
Subcarrier spacings	15/30 kHz*	15 kHz, 30 kHz, 60 kHz, 120 kHz
Bandwidths	Soporte de Ancho de Banda	
	5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100 MHz*	10, 20, 40, 60, 80, 100 MHz
Duplexing	FDD, TDD*	FDD, TDD, dynamic TDD
Split CU/DU	Sí, oficial	Sí, desarrollado por la comunidad

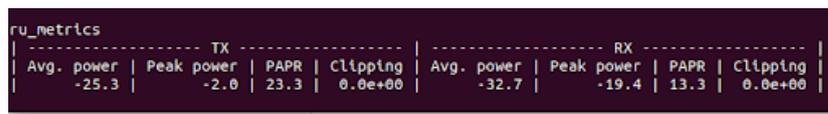
*Si bien *srsRAN* soporta SCS de 30 kHz, TDD y BW mayores a 20 MHz, estos no pueden ser usados por srsUE 4G.

- En *srsRAN*, el split CU/DU es oficialmente soportado, mientras que en *OAI* ha sido desarrollado por la comunidad, lo que podría afectar estabilidad y compatibilidad.

Puede observarse que las desventajas observadas en *srsRAN* frente a *OAI* se deben, en gran medida, a las limitaciones propias del srsUE. Esta diferencia de capacidades ya ha sido señalada en otros trabajos (ver, por ejemplo, la Tabla 3.1 en [9]), y al analizarla en profundidad se identifica que el origen radica en que el srsUE no fue diseñado específicamente para redes 5G SA, a diferencia del UE de *OpenAirInterface*. Al estar limitado a un *Subcarrier Spacing* de 15 kHz, no poder trabajar con modulaciones más allá de 64-QAM y no poder hacer uso de configuraciones avanzadas al usar el RRC de LTE, naturalmente su desempeño se verá limitado si se lo compara con un equipo de usuario que sí tenga esas capacidades.

Teniendo en cuenta los puntos destacados en la comparación teórica, se procede a describir los resultados obtenidos en el proceso de establecimiento de conexión y las observaciones de desempeño registradas, comparando cuando sea pertinente con lo expuesto en el proyecto de fin de carrera “Implementación de una Maqueta de Pruebas y Desarrollo de una Red 5G Stand Alone Completa” [9].

5.2. Establecimiento de conexión y análisis de desempeño



```
ru_metrics
----- TX ----- | ----- RX -----
| Avg. power | Peak power | PAPR | Clipping | Avg. power | Peak power | PAPR | Clipping |
| -25.3 | -2.0 | 23.3 | 0.0e+00 | -32.7 | -19.4 | 13.3 | 0.0e+00 |
```

Figura 5.1: Salida del comando `ru_metrics` en gNB.

5.2. Establecimiento de conexión y análisis de desempeño

A partir del proceso de experimentación se pudieron observar las capacidades del *software srsRAN*, así como del *setup* utilizado. El primer gran desafío fue lograr una conexión estable entre gNB y UE en un entorno de transmisión físico, luego de haber realizado una primera aproximación en un entorno virtual.

En este proceso, se fue adquiriendo una mayor maestría en el uso del *software*, profundizando en sus configuraciones y observando los *logs* que este iba generando. Se encontró que los segundos proveen información de logueo detallada si se establece la variable `log_level` en `info`, lo cual facilita el proceso de hallar y identificar y corregir errores, aunque requiere de un período de aprendizaje para comprender la información desplegada. A su vez, tanto la documentación del proyecto como la comunidad de desarrolladores mostraron una amplia disponibilidad de información y apoyo en la resolución de errores, reforzando su calidad de proyecto *Open Source*.

5.2.1. Establecimiento de conexión inicial y rango de ganancias

Una vez establecidas las configuraciones que teóricamente permitirían una conexión funcional, prestando especial atención a la frecuencia central de transmisión, ancho de banda, tasa de muestreo y tamaño de la trama, se pudo observar que en dichos casos la radiobase detectaba la presencia del equipo de usuario, pero sin lograr establecer la conexión. Esto se pudo detectar utilizando la función `ru_metrics` de la gNB. Al ejecutarlo en la línea de comandos, se presenta una tabla como la de la Figura 5.1. En esta se debió observar la sección RX, prestando especial atención al parámetro *Peak to Average Power Ratio* (PAPR). Para una conexión estable ese parámetro debería encontrarse en el rango entre 10 dB y 20 dB [66].

Utilizando esta herramienta fue posible sortear este primer obstáculo, llegando a la conclusión de que la estabilidad de la conexión era fuertemente dependiente de las ganancias de transmisión y recepción utilizadas. Particularmente, al estar utilizando ganancias mayores a 20 dB y observando un PAPR de unos 8 dB, se terminaba amplificando el ruido sensiblemente, lo que produjo errores en la recepción de la señal en ambos sentidos, impidiendo transmisiones coherentes de un equipo al otro.

Se fijó entonces el rango de ganancias evaluadas en la Subsección 4.1.1. En estas se observaron valores de PAPR que indicaban que ya no se estaba en condiciones de saturación. Fuera de este fenómeno particular, el rendimiento observado en la transmisión fue uniforme a lo largo del rango de ganancias trabajadas, en que se obtenía una conexión estable. Además, se observó que dentro de este rango, variar la ganancia no tuvo implicancias directas en el *bitrate* final (ver Tabla 4.1).

Registro del equipo de usuario: señalización en interfaces N1 y N2

Durante el procedimiento de registro de un equipo de usuario en el núcleo 5G, se verificó el intercambio correcto de señalización entre el UE, el gNB y el AMF. En particular, se analizaron los mensajes de la interfaz N1 (NAS) y N2 (NGAP) capturados durante el proceso de *attach* mediante Wireshark. El procedimiento comenzó con el envío de un *Registration Request (NAS)* desde el UE al AMF a través de la gNB. A continuación, se sucedieron los siguientes mensajes:

- **Initial UE Message** (NGAP, N2): encapsula el mensaje NAS enviado por el UE.
- **Authentication Request / Response** (NAS): procedimiento de autenticación.
- **Security Mode Command / Complete** (NAS): configuración de cifrado e integridad.
- **Registration Accept / Complete** (NAS): aceptación del registro y finalización.
- **Setup of PDU Session** (NAS + NGAP + GTP): establecimiento del túnel de datos.

Estos intercambios confirmaron que el plano de control estaba operativo y que el UE logró establecer una sesión PDU hacia el UPF, habilitando el tráfico de datos a través de la interfaz N3.

5.2.2. Evaluación de *throughput* por ancho de banda

Una vez se lograron conexiones estables, se verificó cuánto tiempo podían estar enlazadas la gNB y la UE sin perder la conexión. Al confirmar que estas lograban mantenerse sincronizadas por más de 10 minutos, se pasó a los test de ancho de banda. Como se puede observar en la Tabla 4.3, los valores de *bitrate* fueron similares para los casos de 20, 15 y 10 MHz, siendo notoriamente menores cuando se trabajó con 5 MHz. Si bien la relación es linealmente decreciente para los 3 valores menores, el *bitrate* para 20 MHz no la cumple. Aquí cabe destacar que con el fin de mantener rigurosidad en los *tests*, se utilizó este valor para la comparativa ya que fue obtenido el mismo día y de forma consecutiva con los otros. De todas formas, al ser 20 MHz el ancho de banda predeterminado utilizado para el resto de las pruebas, se ejecutaron muchas iteraciones de este en otras oportunidades, alcanzando tasas de hasta 20 Mbps en el mejor de los casos. Para una visualización más intuitiva de las diferencias obtenidas se presentan los resultados en formato gráfica de barras (Figura 5.2) donde se los expone junto a los valores teóricos.

5.3. Comparativa práctica con *OAI*

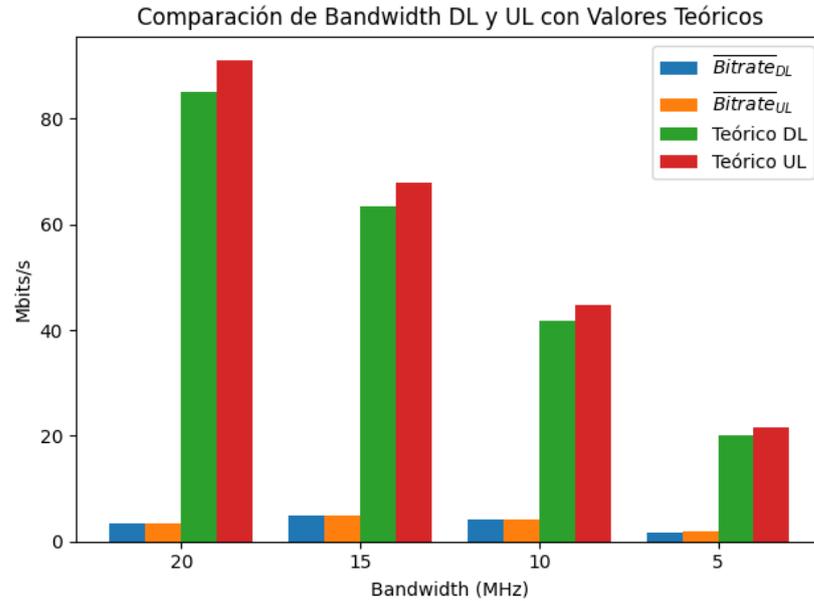


Figura 5.2: Gráfico de barras con valores obtenidos y teóricos para diferentes anchos de banda utilizando dos SDR USRP X310 (Tx y Rx).

5.3. Comparativa práctica con *OAI*

En este punto, se busca suplementar el análisis realizado en la Sección 5.1 con una comparativa práctica con los resultados obtenidos en el trabajo predecesor a este [9]. De allí, se obtuvieron los valores de *throughput* alcanzados con esa solución. Existieron solo dos configuraciones (para 106 y 24 PRBs) en los que las condiciones eran comparables, ya que para otras experiencias se usaron configuraciones que no era posible implementar con *srsUE 4G* modificado. Se presenta en la Figura 5.3 los valores porcentuales que alcanzó cada solución para la misma cantidad de PRBs, utilizando FDD, en la misma banda y utilizando SDR USRP X300. Esta comparativa sirve para entender la distancia en cada caso con el resultado óptimo.

Comparando los resultados obtenidos, se concluye que ambas implementaciones alcanzan rendimientos que permiten operar la maqueta en las configuraciones deseadas, garantizando la conectividad de extremo a extremo. No obstante, la implementación de *OpenAirInterface* logra valores de *bitrate* superiores en relación con el valor teórico esperado. Por otro lado, los valores obtenidos con la maqueta implementada usando *srsUE* se consideran extremadamente bajos, ya que en ningún caso superaron el 10 % del valor teórico.

Si se asocia la observación anterior con lo cercano de los valores resultantes en cada uno de los *tests*, los cuales a su vez fueron inferiores a los esperados, y la lejanía a los valores teóricos, surge la hipótesis de que puede existir algún cuello de botella en el sistema que esté limitando la capacidad de transmisión. Esta hipótesis se fortalece al observar resultados obtenidos por otras experiencias realizadas con *srsRAN*, como es el caso de [10]. En este caso, realizando experimentos con

Capítulo 5. Análisis de resultados

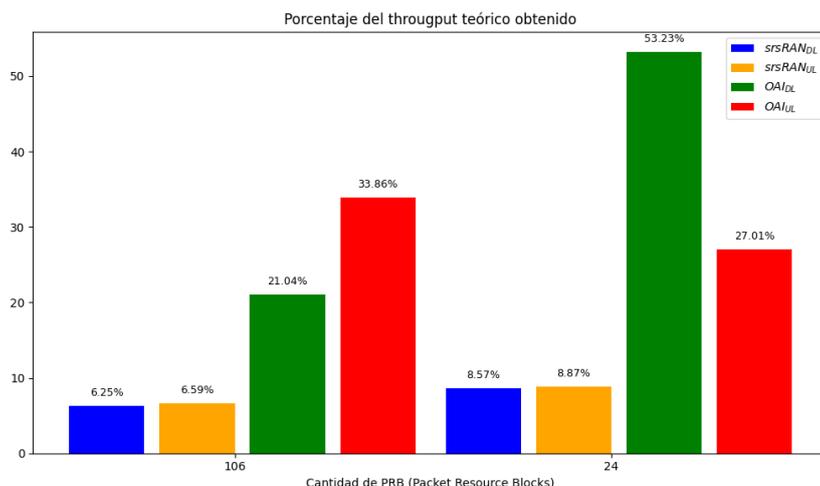


Figura 5.3: Comparativa del valor teórico alcanzado por cada solución.

dispositivos SDR USRP B210 a distancias similares, se obtuvieron valores por encima de los 15 Mbps.

5.3.1. Análisis de anomalías

A lo largo de las pruebas posteriores se pudo observar que la limitante observada anteriormente se presenta por los puertos SFP entre el SDR y la PC, para ambos modelos de SDR utilizados. Esto limita el *throughput* posible tanto en el caso del srsUE como en el uso de la gNB con otros UE, pero sin ser visible en los resultados de las pruebas de caracterización realizadas ya que no impacta directamente el uso de recursos de los respectivos procesos. Se pudo determinar entonces que se daba un efecto de saturación de esta interfaz, que requirió diferentes medidas para resolver en cada caso. Esto es respaldado por experiencias previas como [10] que observaron también que el srsUE presentaba este tipo de fenómenos al trabajar en *throughputs* altos como los generados en este caso por *Iperf*.

Para la interfaz del SDR B200, establecida en un bus USB 3.0, este presentaba errores al continuar las pruebas en el tiempo, llegando a un *throughput* mínimo como el observado en la Figura 5.2. Se pudo determinar que esta degradación se resolvía al reiniciar la interfaz con el SDR, simplemente desconectando y reconectándolo a la PC. Por lo tanto, las pruebas posteriores realizadas con este se hicieron con esta precaución y se obtuvieron mejores resultados como se mencionó anteriormente.

Esto se debe a que el SDR B200 recibe su alimentación por la misma interfaz USB, por lo que esta es la forma más conveniente de provocar un ciclo de reinicio. A su vez, la imagen FPGA de configuración de este SDR no persiste en la placa que realiza el procesamiento, si no que se programa desde la PC al inicializar la interfaz, lo que significa que este proceso causa un reinicio completo que permite volver a utilizar el equipo.

En el caso del Ettus X310, su configuración requirió medidas de mayor com-

5.4. Variación de parámetros en conexión física

plejidad para lograr este resultado. Este cuenta con 2 placas programables que son las que determinan la configuración usada al establecer la interfaz por cable Ethernet, y la alimentación se da por separado mediante una entrada DC de 12 V. Se observó que la placa correspondiente a la interfaz Ethernet de 10 Gbps (declarada por el proveedor) que estaba instalada de usos anteriores al de este proyecto requería actualización, ya que se basaba en *drivers* desactualizados de Ettus. A medida que se avanzaba en las pruebas, los *bitrate* eran más bajos y las desconexiones más frecuentes. Esto llegó a un punto límite cuando el SDR dejó de ser reconocido por la computadora y quedó inutilizable. Luego de una serie de pruebas e investigaciones, se llegó a la conclusión de que se debía reinstalar nuevamente la imagen en la placa para “revivirlo”.

Por lo tanto, se utilizó el puerto *Joint Test Action Group* (JTAG) disponible en la placa para reprogramar este SDR en modo de recuperación, siguiendo la documentación oficial. En el Apéndice D se detalla como se realizó este procedimiento usando el programa de diseño de *hardware* FPGA *Xilinx Vivado*.

5.4. Variación de parámetros en conexión física

Luego de probada la funcionalidad de la gNB estableciendo una conexión inicial, se pasó a estudiar la influencia de los aspectos físicos de la conexión. En particular, se estudió la influencia de la banda y portadora utilizada dentro del FR1, y de la distancia entre las antenas de los equipos correspondientes. Cada uno de estos cambios permite evaluar el impacto de fenómenos que influyen en el rendimiento de conexiones de este tipo, como el ruido presente o la interferencia de señales similares, como se verá a continuación.

5.4.1. Impacto del ruido presente en la conexión

El primer parámetro a estudiar fue la frecuencia central de las portadoras utilizadas durante la transmisión, evaluando no solo diferentes opciones dentro de la banda *n3* inicialmente usada, si no también diferentes bandas del FR1.

En primer lugar, dentro de la banda *n3* se observó que el rendimiento de las transmisiones realizadas a diferentes frecuencias no se correlacionaba necesariamente con el ruido presente. Esto se ve representado por el hecho de que los valores de *bitrate* obtenidos en transmisiones con más ruido como la dada 1820 MHz fueron mayores que los obtenidos en la banda usada inicialmente, a pesar de la presencia de un ruido de fondo mayor. Las frecuencias elegidas fueron seleccionadas por presentar diferentes patrones en el ruido, el cual en algunos casos era igual y uniforme tanto para *Uplink* como *Downlink* y en otros, por el contrario, dispar y no uniforme.

Como se mencionó, el rendimiento observado fue de igual manera independiente a la distribución de ruido en cada caso. Se observa entonces que una vez establecida la conexión, esta es robusta ante los efectos y errores que puede ocasionar el ruido blanco.

5.4.2. Evaluación de diferentes bandas

Por otro lado, al variar las bandas utilizadas, se buscó analizar el impacto de la interferencia generada al operar en espectro asignado a distintos proveedores de servicios de red. Para ello, se seleccionaron bandas representativas de los tres operadores de 5G en Uruguay: Antel, Movistar y Claro, así como un ejemplo en una banda no asignada. Como se puede observar en la Figura 5.4, el efecto fue una notoria mejoría en la banda sin interferencia, evidenciando que trabajar en bandas con tráfico disminuye considerablemente el desempeño de la red, como era de esperarse.

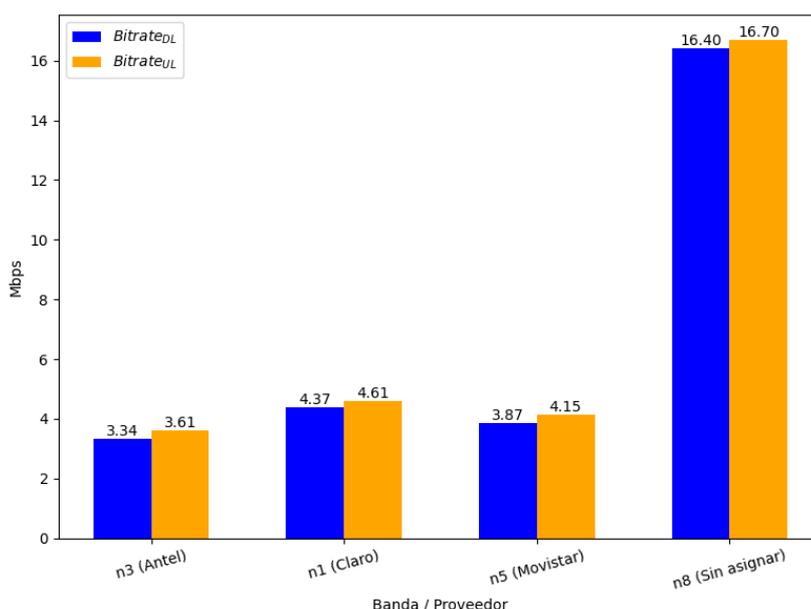


Figura 5.4: Comparación de *bitrate* obtenido según la banda de operación.

5.4.3. Impacto de la distancia física entre equipos

Resulta importante estudiar también las distancias a las cuales los equipos lograban una conexión estable, y corroborar si esto afectaba el desempeño del sistema. Al duplicar la distancia inicial de 0.5 m ya se debieron modificar las ganancias, como puede verse en la Tabla 4.6. En este caso, solo debió incrementarse la ganancia de recepción de la gNB, dado que se observó que el parámetro PAPR había aumentado, denotando una recepción más pobre. Solo bastó con aumentar 5 dB para compensar el aumento de distancia. Este patrón se repitió al pasar a trabajar a 3 m. Inicialmente, a esta distancia la recepción era muy pobre, tanto en gNB como en UE. Siguiendo la idea anterior, se aumentaron levemente los valores de todas las ganancias hasta lograr una conexión estable, compensando las pérdidas por distancia. Se observa entonces que la maqueta implementada es altamente sensible a las posiciones relativas entre gNB y UE y las ganancias deben ser ajustadas con gran precisión.

5.5. Caracterización de Hardware



Figura 5.5: Laboratorio en que se realizaron las pruebas, en este caso se representa la prueba de distancias máxima, señalando la ubicación de los SDR.

Esto se confirmó con un último experimento, en el que se colocaron ambas partes a 5m de distancia y sin línea de vista, como puede observarse en la Figura 5.5. Dada la disposición del laboratorio donde se hicieron las pruebas, esta fue la distancia máxima alcanzable, y en medio se encontraban objetos industriales de gran porte hechos de hierro y acero. En este caso, no se alcanzó una conexión estable para ningún conjunto de ganancias. Si estas se encontraban por debajo de los 20 dB, los elementos de la red no se detectaban. En el rango de 20 a 30 dB, los logs mostraban indicios de conexiones incipientes, indicando en el log del UE el mensaje “*Cell Search Found*”, pero en el intercambio de parámetros las respuestas no llegaban a tiempo y la conexión fallaba. Sin embargo, con ganancias superiores a 30 dB nos encontrábamos en zona de saturación, donde la conexión no era viable como se vio anteriormente. En definitiva, no se logró establecer una conexión en este caso. Se trabajó intentando corregir a su vez la frecuencia de *offset* del oscilador manualmente evaluando mediante los logs su valor inicial (ejercicio que se realizó con éxito a distancias menores) y pasándola como parámetro al UE, pero esto no resultó útil para este caso.

5.5. Caracterización de Hardware

Durante la ejecución de todos estos experimentos, se registró el consumo de recursos de las diferentes aplicaciones, como se explicó en la Sección 3.3. De esta manera, se pudo rastrear la exigencia en recursos de cada equipo simulado en diferentes instancias de ejecución.

Se observó que el uso de recursos de cada proceso era estable en los diferentes casos estudiados. El uso de memoria RAM se mantuvo constante durante la ejecución tanto en el entorno virtual como en el físico, y la principal variable fue el uso de procesador para cada proceso.

Capítulo 5. Análisis de resultados

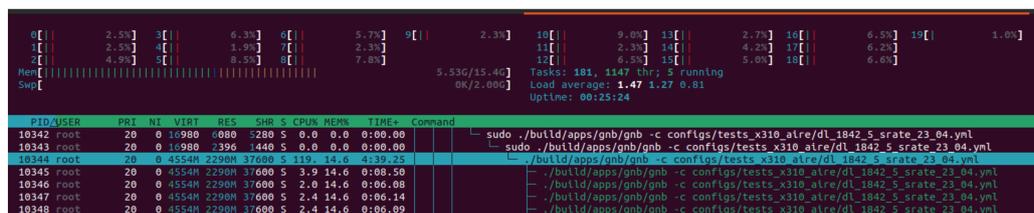


Figura 5.6: Software *htop* mostrando uso de recursos durante ejecución de gNB y Core.

En el caso del contenedor *Docker* usado para ejecutar el núcleo de red *Open5GS*, se observó que el contenedor tenía asignados de forma constante 465.5 MB de RAM, independientemente del uso que se le estuviera dando al proceso. El uso de procesador también fue notoriamente bajo, siendo en promedio un 10 % de uso. Se observó a su vez que al conectar múltiples UEs a una única gNB el uso de recursos del núcleo de red no aumentaba, por lo que el principal aumento de procesamiento en dichos casos se observó en la gNB.

Teniendo esto en cuenta, se deduce que la ejecución del núcleo de red se puede llevar a cabo en una variedad de equipos debido a la poca exigencia de recursos que este presenta, siendo incluso viable una implementación en que se ejecuta esta aplicación en el mismo equipo que la radiobase.

Por otro lado, la aplicación del proyecto *srsRAN* usada para simular la gNB fue la más computacionalmente intensa de las observadas. En este caso se rastreó el uso de CPU y memoria RAM del proceso usando *htop* mientras se aplicaban los diferentes *tests*, de forma de evaluar su rendimiento en una variedad de condiciones. En la Figura 5.6 se presenta la información mostrada por este *software* con el servidor ejecutando la radiobase y los contenedores *Docker* en simultáneo. En azul se resalta la línea del proceso de la gNB, evidenciando ser el proceso con mayor consumo.

Nuevamente se observó un uso constante de memoria RAM, en este caso de 2.3 GB. El uso del procesador fue más dinámico e intenso en este caso, oscilando entre en promedio un uso de 60 % de un núcleo de CPU en los periodos en que se estaba ejecutando sin conexión con un UE activo, hasta un máximo de 120 % de uso de CPU en promedio durante las pruebas de throughput ejecutadas ¹. Esto concuerda con lo esperado, ya que durante estas se utilizó *iperf3* para transmitir al máximo *throughput* posible. Sin embargo, no se trata de un consumo que se pueda considerar excesivo, al ser suficiente con un único núcleo de procesador y una porción de memoria residente de un tamaño fácilmente obtenible en múltiples de los posibles equipos de uso extendido para la etapa de implementación.

Por último, la evaluación de rendimiento del proceso usado para ejecutar el UE *srsUE* arrojó resultados similares a los obtenidos en la gNB. El consumo de recursos fue nuevamente estable durante la ejecución, aunque en una escala menor. Se observó un uso constante de 680 MB de memoria, y un uso de CPU que osciló entre un 30 % de uso de un núcleo durante periodos sin conexión o en estado de

¹*Htop* presenta los porcentajes en función de los núcleos. Una utilización del 100 % corresponde a un núcleo entero. El procesador utilizado cuenta con 12 núcleos

IDLE, a un 80 % de uso durante las pruebas de *throughput*.

Estos valores sugieren nuevamente la viabilidad de trasladar el srsUE a equipos de menor porte, sin tener un impacto significativo en el desempeño logrado. A su vez, se espera que si se logró tener un resultado aceptable con tan bajos requerimientos y una conexión estable, un equipo COTS lo logre sin problema, ya que cuenta con *hardware* dedicado creado para esta tarea.

De hecho, en las pruebas siguientes con un COTS UE se observó que este establecía fácilmente una conexión a la red en todas las configuraciones utilizadas, sin requerir mayor intervención del usuario. Esto entró dentro de lo esperado, ya que estos equipos están optimizados para realizar conexiones y desconexiones constantes y en diferentes configuraciones.

5.6. Análisis de Configuraciones

A partir de los resultados obtenidos en las pruebas (ver Sección 4.2), en esta subsección se analiza el impacto de distintas configuraciones en el desempeño de la red. Es importante destacar que en esta etapa los SDR utilizados fueron los USRP B200, ya que resultaron más estables que los USRP X310 en las mismas condiciones (ver Subsección 5.3.1).

5.6.1. Variantes en UEs

En una primera instancia, nuevamente se pudo observar una clara diferencia a favor de los equipos comerciales al comparar el rendimiento de cada tipo de UE utilizado. Como se expone en la Figura 5.7, donde se presentan los valores de *bitrate* en *Downlink*² para cada equipo, en todos los casos el COTS UE Samsung alcanzó los *bitrates* más altos. De todos modos, el *Hat Sixfab 5G* para *Raspberry Pi 4* alcanza valores cercanos, siendo estos dos ampliamente superiores a la solución con srsUE. Se observa que el COTS UE alcanza entre 70 % y 80 % del valor teórico (ver Tabla 4.2).

Al conectar varios equipos de usuario se observaron resultados variables. Por ejemplo, al enviar tráfico en simultáneo desde el COTS UE y el *Sixfab*, el resultado de ambos decrece a casi la mitad de lo que alcanzaron cada uno por separado. Si bien es un descenso grande, era de esperar que al traficar ambos en simultáneo utilizando el mismo medio, los *bitrate* descendieran. Sin embargo, al utilizar el srsUE con cualquiera de los otros dos equipos, los resultados fueron llamativos. Tanto el *Sixfab* como el COTS UE descendieron su *bitrate* más de un 75 %. Es decir, funcionaron mejor cuando estaban conectados estos dos, que cuando estaba presente el srsUE. Este acaparó la mayor cantidad de tráfico cuando estuvo presente, como se muestra en la Tabla 4.11.

Este fenómeno se adjudica principalmente a la asignación de recursos. En estos casos se debería hacer un ajuste dinámico de parámetros como el MCS, pe-

²Los valores de UL en todos los casos son muy cercanos. Se decide omitirlos ya que no aportan información comparativa en este caso.

Capítulo 5. Análisis de resultados

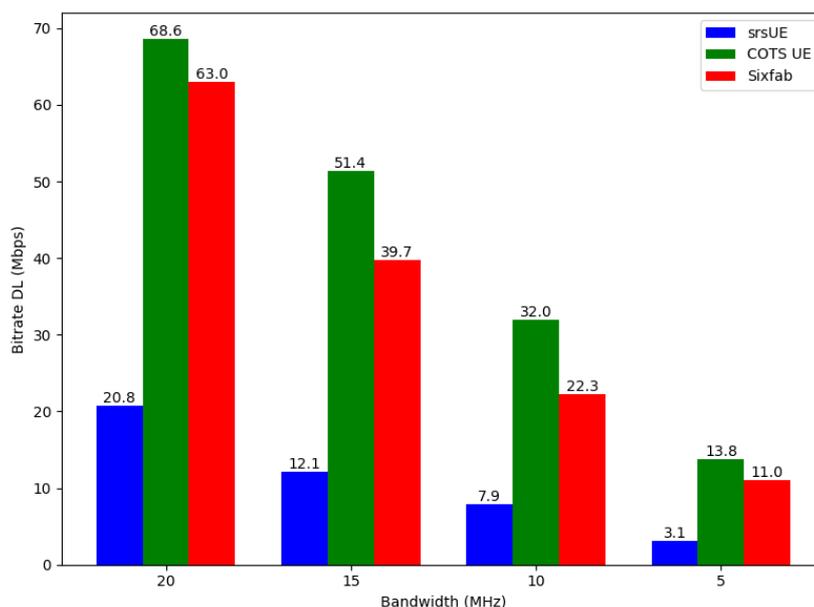


Figura 5.7: Comparación de *bitrate* DL para srsUE, COTS UE y Sixfab, alcanzados en pruebas internas en la red.

ro al no contar el srsUE con esta funcionalidad se observó que este se mantenía estático, sin permitir que se ajuste a los requerimientos de radio. A su vez, tampoco cuenta con potencia adaptativa. Esto explica entonces como en estos casos el srsUE acaparaba los recursos de la red que podrían haber sido destinados a otros UE (COTS UE y *Sixfab* UE) que “ceden” los recursos disponibles, dado que sí cuentan con implementaciones de asignación dinámica. Esto concuerda con otras investigaciones que observaron que el srsUE (y en menor grado también el UE implementado por OAI) presentaban irregularidades en escenarios de conexión con múltiples UEs adicionales, por no estar implementadas en estos las capacidades para asignación dinámica de recursos de un UE 5G estándar [11].

Además, se estima que las altas ganancias que manejan los SDR USRP B200 para lograr la conexión pueden haber potenciado esta situación. Al estar trabajando con valores de 60 o 70 dB en de ganancia en Tx, sus transmisiones predominan e interfieren con el resto de las señales del ambiente, limitando así a los otros dispositivos. Esto se pudo observar nuevamente al tener los tres equipos conectados (ver Tabla 4.11). Si bien el COTS UE mejoró levemente su rendimiento y el *Sixfab* también, se observan valores bajos mientras que srsUE sigue teniendo *bitrates* mucho mayores.

Sin embargo, lo más llamativo fue que el srsUE no solo opacó a los otros dos, sino que incluso mejoró sus propios *bitrate* (ver Tabla 4.9), alcanzando valores de hasta 35.1 Mbps (los más altos de todas las pruebas) al compartir el medio con el COTS UE. Aquí se debe puntualizar un aspecto clave. Durante el desarrollo de este trabajo se realizaron *tests* de *throughput* para una misma configuración en diferentes oportunidades. Puntualmente, para los casos en que se utilizó srsUE, los

5.6. Análisis de Configuraciones

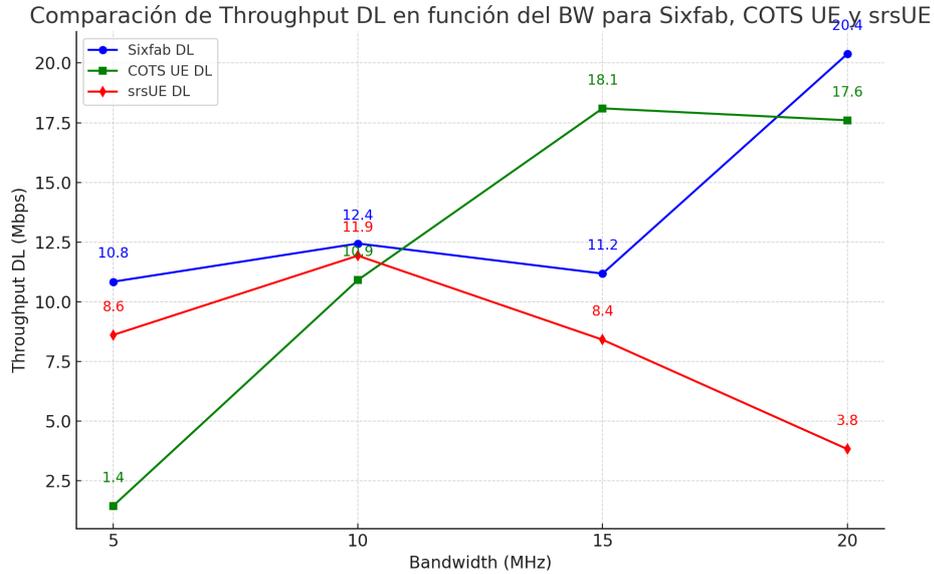


Figura 5.8: Comparativa de *Downlink* en *speedtests*.

resultados variaron significativamente (hasta un 100 %) según el día. Se asocian estas variaciones a que si bien el entorno fue idéntico, este no estaba aislado y se localizó en una zona céntrica, con mucho tráfico y por lo tanto con gran variabilidad. Las condiciones de interferencia no se pudieron garantizar idénticas para cada prueba. En este caso, las pruebas de srsUE con un SDR B200 solo no se realizó el mismo día que las de concurrencia, pudiendo este ser el factor principal en la inconsistencia. Además, como se mencionó en las secciones anteriores, la saturación de las interfaces jugó un papel fundamental en los *bitrate* alcanzados. Era muy diferente lo observado al realizar un *test* de *Iperf3* cuando recién se conectaba el SDR que si había estado conectado por un largo período.

Desde un punto de vista general, se entiende que esta etapa fue satisfactoria ya que si bien los *bitrate* descendieron, fue posible conectar hasta 3 equipos de usuario a una misma radiobase y testear su *throughput* en simultáneo sin grandes perjuicios como desconexiones o saturación de la radiobase. Se resalta la sensibilidad de srsUE, la alta variabilidad de los resultados y cómo este afectó a otros equipos en la red. Produjo mejores resultados y fue más sencillo trabajar solamente con equipos comerciales.

Por otra parte, al realizar *speedtests* hacia Internet, los resultados también presentaron variaciones interesantes. Como se muestra en la Figura 5.8, tanto el *Sixfab* como el COTS UE tuvieron resultados coherentes de conexión a Internet, ya que aunque no son estrictamente crecientes según el ancho de banda, se encuentra dentro del margen de variación esperado. Por otro lado, el srsUE presenta un pico en 10 MHz para luego descender notoriamente. Este comportamiento fue inesperado y no se encontró mayor fundamento más que una posible saturación de la interfaz USB 3.0 luego de reiteradas pruebas o un tráfico particularmente alto por la red Wi-Fi pública utilizada durante este período.

Capítulo 5. Análisis de resultados

Profundizando en los usos reales de la maqueta, se concluyó que era perfectamente viable descargar paquetes del repositorio *apt* de Ubuntu y consumir contenido multimedia con los dispositivos comerciales. Si bien con el srsUE se podía acceder a la página solicitada y el video comenzaba a cargar, a los pocos segundos la conexión se veía saturada debido al alto tráfico y este se desconectaba de la radiobase. Esto se repitió al utilizar la plataforma *Zoom* de videoconferencias, mientras que al establecer una llamada entre el COTS UE y *Sixfab* no había pérdida de conexión pero sí de paquetes, haciéndola inviable para un uso real de usuario.

5.6.2. Rendimiento de splits

Para estudiar las ventajas de la aplicación del *split 2* en esta arquitectura, se comparan los diferentes escenarios presentados en la Subsección 4.2.2. Como se observa en la Figura 5.9, el desempeño obtenido en el ancho de banda evaluado de 5 MHz es de orden similar en la mayoría de implementaciones. Los valores resultantes concuerdan a su vez con los observados previamente en esta banda con el mismo UE en la sección anterior, obteniendo solamente una leve reducción en el *throughput* final.

Esto representa por un lado que el uso de este tipo de *splits* no resulta en una disminución significativa de rendimiento en la comunicación establecida, dado que se observaron pocas pérdidas en el uso de cada uno respecto a la gNB “monolítica”. Además, las diferentes interfaces probadas no presentan una limitante en cuanto al rendimiento de estas arquitecturas, concluyendo que este es similar en todos los casos. Esto era esperado para la conexión por cable Ethernet de 2.5 Gbps, pero se corrobora entonces que incluso una interfaz más susceptible a interferencias como la establecida mediante Wi-Fi no genera impedimentos significativos ni presenta una limitante en sí.

A su vez, se determina entonces que el NUC cuenta con la capacidad suficiente para ejecutar la DU en esta banda sin observar impedimentos por rendimiento del *hardware* o falta de recursos. Esto es relevante porque el objetivo de esta evaluación es no solamente obtener un relevamiento de las capacidades de estas diferentes arquitecturas, sino también determinar su utilidad para aplicaciones que se beneficien trasladar esta funcionalidad a equipos de menor porte como el propio NUC.

Por ejemplo, estos mini PC son implementados como computadora a bordo para varios tipos de dron, usados en áreas tanto académicas como comerciales [67]. Dadas las capacidades observadas de la conexión estableciendo interfaz F1AP por Wi-Fi, esto abre puertas a establecer gNBs de procesamiento distribuido que cuenten con puntos de acceso móviles. Este proceso más liviano puede ejecutarse en este tipo de equipos junto con los procesos de navegación y telemetría de los drones, que suelen ser más demandantes computacionalmente.

Estudios anteriores como [68] han observado que los requerimientos de red para la transmisión de datos de telemetría en este tipo de drones se basa en paquetes de pocos bytes, lo que significa que una red de bajo ancho de banda pero baja latencia como la obtenida lograda en este proyecto cuenta con las prestaciones suficientes

5.6. Análisis de Configuraciones

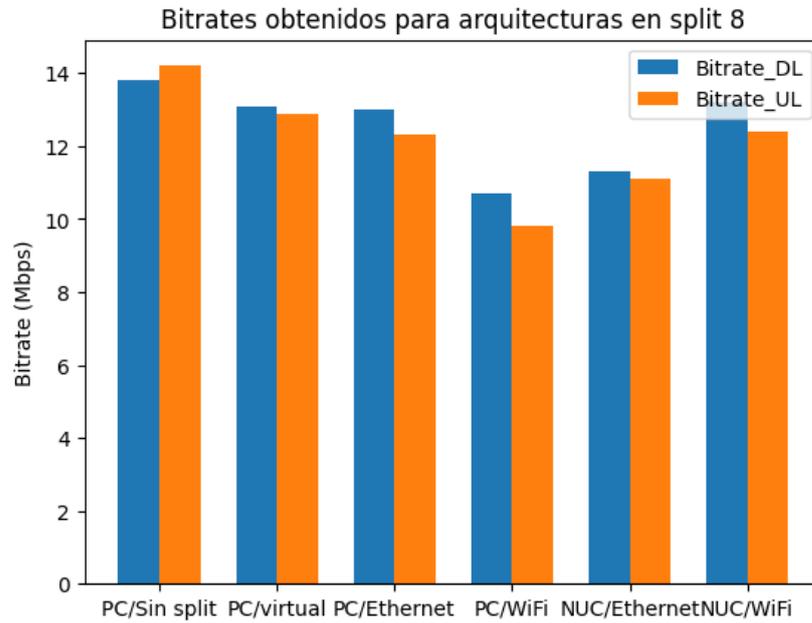


Figura 5.9: Comparación entre arquitecturas evaluadas en Split 8, según Subsección 4.2.2.

para servir de medio para la transmisión de telemetría en estos casos.

Por esta razón se han planteado ya trabajos futuros en el ámbito de la Facultad de Ingeniería que iteran sobre esta arquitectura para aprovechar estas ventajas en diferentes aplicaciones y ampliar la capacidad de esta conexión.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 6

Conclusiones y trabajo a futuro

El siguiente capítulo presenta una síntesis de los resultados alcanzados durante el desarrollo de este proyecto. A su vez, se analizan las principales conclusiones derivadas de la implementación de la maqueta y sus variantes, identificando fortalezas y limitaciones. Finalmente, se plantean posibles líneas de trabajo a futuro que permitirían extender y aprovechar la infraestructura desarrollada en nuevos escenarios.

6.1. Cumplimiento de objetivos

El primer objetivo propuesto para este proyecto fue la implementación exitosa de una maqueta 5G funcional basada en el proyecto *srsRAN*. Esta meta fue alcanzada mediante la puesta en marcha de una red en modo *Standalone*, con su correspondiente núcleo de red (basado en *Open5GS*) y una radiobase operativa. Se lograron establecer conexiones con múltiples equipos de usuario, tanto físicos como virtuales, y validar el intercambio de datos a través de la red. Esta maqueta fue utilizada como base para el resto de los objetivos.

En consecuencia, se logró cumplir con el segundo objetivo, realizando una comparativa con el proyecto antecesor “Implementación de una Maqueta de Pruebas y Desarrollo de una Red 5G Stand Alone Completa” [9]. Se logró utilizar exitosamente *hardware* base idéntico y replicar el entorno de pruebas en la medida de lo posible. Si bien existieron casos que no se pudieron repetir de forma idéntica dadas las limitaciones del *srsUE*, esta limitante aporta en sí misma a la comparación y es fundamental. Particularmente, las limitaciones principales fueron las siguientes:

- **Ausencia de MIMO:** el sistema no implementa múltiples flujos espaciales.
- **Ancho de banda limitado a 20 MHz:** no es posible configurar *srsUE* para anchos de banda mayores a 20 MHz.
- **Uso exclusivo de numerología 0:** se utilizó un spacing de subportadoras de 15 kHz, correspondiente a la numerología 0, por ser la única soportada en el entorno utilizado.

Capítulo 6. Conclusiones y trabajo a futuro

- **Operación en bandas FDD únicamente:** el *srsUE* sólo es compatible con bandas FDD, con una frecuencia de muestreo máxima de 30.72 MHz, lo que acota las bandas posibles.
- **Limitaciones en el uso de MCS adaptativo:** no fue posible variar el esquema de modulación y codificación de manera dinámica durante las pruebas, debido a restricciones del software del UE.

Se concluye que, de desearse implementar una maqueta *end-to-end* con una única solución, *OpenAirInterface* provee un entorno completamente desarrollado para 5G y mejores resultados prácticos que *srsRAN*. Sin embargo, en caso de solamente realizar un despliegue de núcleo y radiobase, la combinación de *Open5GS* y *srsngNB* brinda una solución completamente capaz de soportar tráfico con múltiples equipos de usuario conectados en simultáneo y mayor versatilidad para su implementación en equipos de menor porte, sorteando así las limitaciones del *srsUE*. La posibilidad de implementar la radiobase separando CU y DU acarrea flexibilidad y posibilidad de innovar en el despliegue, así como una opción práctica para la reducción de costos de *hardware*.

Siguiendo con los equipos de usuario, se planteó como objetivo explorar la conectividad con una mayor variedad de UEs que en trabajos anteriores. En este sentido, se logró establecer conexión con un UE comercial (COTS), que operó de forma estable utilizando una SIM física programable, logrando intercambiar tráfico real. Además, se adquirió y configuró exitosamente un dispositivo *Sixfab 5G Hat*, basado en un módem *Qualcomm*, que permitió validar la conectividad desde plataformas típicas de IoT. Esta experiencia resultó especialmente relevante para analizar escenarios de despliegue embebido y portabilidad, ya que el dispositivo se integra con facilidad a soluciones compactas y móviles.

El siguiente objetivo fue la caracterización funcional de la maqueta. En este punto se encontró que la ejecución del núcleo y la radiobase no representaron un consumo de recursos particularmente alto, siendo posible su despliegue en equipos de cómputos de gama media, como fue el caso de una PC con 16 GB de Memoria RAM y un procesador Intel Core i7. De todos modos, se observó que al intentar trasladar la radiobase a equipos de menor porte como el Intel NUC, la capacidad de transmitir datos se veía comprometida y se debía bajar el ancho de banda del canal al mínimo posible para lograr una conexión estable.

Finalmente, se abordó de forma explícita el objetivo de lograr una arquitectura portable para la maqueta implementada. Para ello, se trabajó en distintas variantes de despliegue, que incluyeron la ejecución de la gNB completa en un Intel NUC, así como la separación funcional CU/DU con la DU ejecutándose de forma autónoma en el NUC. Esta última variante representó un paso importante hacia una gNB modular y portable. Como fue planteado al inicio del proyecto (ver Subsección 1.2.1), la portabilidad se evaluó en tres dimensiones: espacial, de conectividad y energética. La ejecución de la DU en un NUC pequeño, alimentado por batería externa y con conectividad inalámbrica hacia la CU, permitió una implementación práctica y completamente funcional que puede considerarse portable en los tres sentidos.

6.2. Conclusiones sobre la maqueta y sus variantes

Profundizando en la implementación de la maqueta basada puramente en *srs-RAN*, se observó que su punto de mayor debilidad fue el módulo *srsUE*. Si bien permite realizar conexiones básicas y es útil para pruebas iniciales, no soporta muchas de las funcionalidades propias del 5G moderno ni ofrece confiabilidad suficiente para tareas exigentes o prolongadas, lo que se ve directamente reflejado en los *bitrate* máximos alcanzados. Además, su conexión a la radiobase debe realizarse de forma minuciosa, teniendo presentes múltiples aspectos de su configuración, siendo fundamentales las ganancias utilizadas. Estas presentaron una sensibilidad e inestabilidad muy altas. En adición, las interfaces de conexión con los SDR, tanto 10Gb Ethernet como USB 3.0, presentaron diferentes problemáticas que incidieron directamente en los *throughput* alcanzados.

Uno de los puntos más relevantes del desarrollo fue la buena estabilidad general de la maqueta para el tráfico de datos al trabajar con equipos comerciales como el COTS UE y el *Sixfab Hat 5G*. La experiencia con estos dispositivos fue satisfactoria y permitió comprobar que la maqueta es capaz de integrarse con *hardware* real. Se alcanzaron *bitrates* mayores a los esperados y cercanos a los valores teóricos, por lo que en ese sentido se considera que la implementación es confiable al trabajar con un único equipo de usuario. Esto se suma entonces a las observaciones de estudios anteriores como [10] que respaldan la preferencia por UEs especializados, en vez de las opciones simuladas de *srsRAN* y *OAI*. De esta manera, se aprovecha al máximo la funcionalidad de la radiobase basada en *srsRAN*.

De todos modos, se identificaron aspectos críticos como el tiempo de establecimiento de conexión, la estabilidad con múltiples UEs y una baja respuesta ante cambios dinámicos de configuración. Se detectaron, por ejemplo, irregularidades en el comportamiento de los UEs al conectarse simultáneamente, incluso si se trataba del mismo modelo de dispositivo, lo que coincide con observaciones en estudios recientes [11]. Esta caracterización permitió entender mejor los límites actuales de *srsRAN* en contextos de uso realista.

La arquitectura propuesta, compuesta por *srsgNB* y *Open5GS*, permitió validar múltiples configuraciones, incluyendo tanto ejecuciones monolíticas como divididas en CU/DU. Esto resultó fundamental para avanzar en la portabilización de la red, ya que no solo facilitó la distribución de funciones entre dispositivos de diferente capacidad, sino que también permitió trasladar parte de la red a plataformas de bajo consumo como el Intel NUC.

6.3. Trabajo a futuro

A partir de la maqueta desarrollada y sus variantes, se abren múltiples líneas de trabajo que permiten extender las capacidades del sistema. Durante este proyecto se creó un repositorio de *Gitlab* donde se pueden encontrar los archivos necesarios para recrear la maqueta en todas las diferentes versiones desarrolladas [18], permitiendo así la continuación en diferentes caminos. Una posibilidad concreta es evaluar la interoperabilidad cruzada entre distintos proyectos *open-source*, utili-

Capítulo 6. Conclusiones y trabajo a futuro

zando, por ejemplo, el módulo *srsNB* junto a un núcleo *Open5GS*, pero reemplazando el módulo de usuario *srsUE* por el UE implementado por *OpenAirInterface* que tanto en esta como en otras experiencias (ver [13]) ha demostrado ser superior. Queda planteada esta experimentación como posible solución a las limitantes impuestas por *srsUE*, mencionadas anteriormente.

A su vez, sería de gran interés explorar nuevos niveles de separación funcional (*splits*) altos, como el 6 o 7.2x, que distribuyen aún más la arquitectura de la radiobase. Esto permitiría investigar configuraciones híbridas en las que parte del procesamiento de la señal se traslade a la nube o a nodos de borde, en línea con los principios de virtualización y despliegue escalable que promueve la arquitectura O-RAN.

Otra línea de investigación interesante es el despliegue de la radiobase sobre plataformas móviles. La mini PC Intel NUC utilizada fue obtenida tras removerla de un dron, donde oficiaba de computadora a bordo. Este proyecto sienta las bases para el estudio de la viabilidad de incluir un SDR sobre estos drones, para ejecutar el control de vuelo del dron junto con la radiobase implementada.

Finalmente, se destaca que esta maqueta puede servir como base para pruebas más amplias, incluyendo algoritmos de asignación de recursos, simulaciones de tráfico o evaluaciones de nuevas funcionalidades del estándar 5G. También resulta de interés probar el módulo conformado por la *Raspberry Pi* y el *Hat Sixfab 5G*, teniendo potencial para el buen manejo del tráfico de tipo IoT. Incluso, dados los *splits* trabajados, es viable la implementación de múltiples DU trabajando en simultáneo [69]. La versatilidad de su arquitectura y la modularidad del entorno desarrollado hacen que pueda adaptarse a distintos proyectos de investigación, enseñanza o evaluación de tecnologías en fase experimental.

Apéndice A

Manual de instalación y configuraciones

En este apartado se deja constancia de cómo desplegar todas las componentes y configuraciones utilizadas durante este proyecto, a modo de guía para futuros usuarios.

Requerimientos iniciales

Previo a comenzar el despliegue de la RAN, es necesario instalar ciertas bibliotecas. Inicialmente, se debe contar con equipos Linux y acceso a `sudo` en estos equipos para correr los comandos. Este proyecto utilizó la distribución Ubuntu 22.04 LTS, pero es posible utilizar otras.

Como primer paso, se deben descargar las herramientas de *build*, dependencias y los *drivers* de UHD, tal y como se establece en la página oficial de Ettus [70]. En este caso se debió correr el comando en terminal:

```
sudo apt update
sudo apt-get install cmake make gcc g++ pkg-config libfftw3-dev
libmbedtls-dev libsctp-dev libyaml-cpp-dev libgtest-dev
sudo apt-get install libuhd-dev uhd-host
```

Además, es necesario contar con `git` en el equipo y clonar los repositorios de trabajo:

```
sudo apt-get install git
git clone https://gitlab.fing.edu.uy/portaran/portaran_
srsran
git clone https://gitlab.fing.edu.uy/portaran/srsran_4g_
portaran
```

El primer repositorio, *portaran_srsran* es un *fork* del git de *srsRAN Project* [57], correspondiente al proyecto 5G. De allí se ejecutarán los módulos del núcleo y gNB. Por otro lado, el segundo es un *fork* del git de *srsRAN 4G* [71], utilizado para ejecutar el UE. En caso de implementar las componentes en diferentes equipos, se debe clonar el repositorio correspondiente en cada uno, no es necesario descargar ambos.

Núcleo de red

El núcleo de red se ejecuta mediante el uso de un conjunto de contenedores *Docker*, orquestados utilizando la herramienta *docker-compose*. Los archivos de configuración se encuentran en la carpeta **docker** del repositorio *portaran_srsran*. Con *docker-compose* se ejecutan tanto el contenedor correspondiente al núcleo de red implementado por *Open5GS* como otras aplicaciones adyacentes que sirven para monitorear el tráfico por la gNB y mostrar una interfaz gráfica para el control de usuarios en el núcleo de red (ver Figura 3.4 en la Sección 3.1). Para desplegarlos se ejecuta el siguiente comando:

```
sudo docker compose up 5gc grafana metrics-server influxdb --build
```

Se recomienda generalmente levantar estos servicios adicionales junto con el núcleo de red ya que son útiles para medir el funcionamiento de la maqueta y su uso de recursos es negligible. A su vez, se recomienda usar la *flag* `--build` para volver a crear las imágenes desde las que se levantan estos contenedores en caso de realizar cambios en la configuración de los mismos.

La configuración de orquestado de los contenedores se encuentra también en la carpeta **docker**, en el archivo `docker-compose.yml`. Aquí, cada contenedor tiene una entrada en que se configura como se conecta con el resto en la subred generada por *Docker*, su uso de volúmenes y los comandos que ejecuta al iniciar, entre otros.

Luego, cada contenedor toma su configuración individual de una subcarpeta con su nombre en el directorio **docker**. En el caso particular del contenedor del núcleo de red, este se configura principalmente en el archivo `open5gs-5gc.yml`. Aquí se configuran las direcciones y parámetros correspondientes a funciones como la AMF y UPF, entre otros. Los valores modificados de este archivo fueron configurados de forma de poder tomar los mismos de un archivo de variables de entorno del contenedor (`open5gs.env`), de los cuales se detallaron los más relevantes en la Tabla A.1. De esta manera, es más fácil para el usuario encontrar y modificar parámetros desde las variables de entorno. Por último, se configuró un archivo `subscriber_db.csv` que contiene la base de datos de usuarios con que el núcleo de red inicia su ejecución.

Radiobase

Para ejecutar la gNB, el usuario debe antes crear la *code-base*, tal como indica la guía de instalación oficial [72]. Para eso, ejecutar:

```
cd portaran_srsran
mkdir build
cd build
cmake ../
make -j $(nproc)
make test -j $(nproc) (opcional)
```

Este proceso puede tardar varios minutos. Para facilitar los siguientes pasos,

Tabla A.1: Parámetros de acceso frecuente configurados en el archivo `open5gs.env`.

Parámetro	Uso	Valor
<code>OPEN5GS_IP</code>	Dirección IP del núcleo de red fuera del contenedor	10.53.1.1
<code>UE_IP_BASE</code>	Subred en la que se asignarán las direcciones de los UEs conectados a la red, con máscara /24	10.45.0
<code>UPF_ADVERTISE_IP</code>	Dirección a mostrar para ubicación de la UPF para otros servicios (por ejemplo la gNB)	10.53.1.2
<code>LOG_LEVEL</code>	Nivel de información a mostrar en los logs del contenedor del núcleo de red, trasladados a la terminal en que se ejecuta	info
<code>SUBSCRIBER_DB</code>	Nombre del archivo CSV en que se encuentra la base de datos de suscriptores	subscriber_db_sim.csv
<code>NETWORK_NAME_FULL</code>	Nombre completo de la red, usado como APN	PortaRAN
<code>TZ</code>	Huso horario a utilizar en el contenedor, determina el tiempo fijado por el núcleo para la red	America/Montevideo

el equipo de trabajo creó un *script* llamado `prepare_gnb.sh`, el cual ejecuta los comandos necesarios para optimizar el rendimiento del equipo y calibrar diferentes aspectos de configuración previos a la ejecución de la gNB:

- Parámetros de rendimiento, adaptados del procedimiento `srsran_performance.sh` del repositorio original [57], principalmente:
 - Configurar el regulador de escala del CPU en modo “performance”
 - Ajustar tamaños de *buffers* de red
- Agregar ruta a UEs mediante la gNB para dirigir tráfico a los mismos por la red
- Activar *forwarding* de paquetes
- Crear ruta en tablas IP para conectar los UEs de la red a Internet por la interfaz de red correspondiente en el equipo, especificada al ejecutar el *script*:

```
sudo prepare_gnb [nombre de interfaz wifi]
```

Apéndice A. Manual de instalación y configuraciones

Ahora, desde la raíz del repositorio *portaran_srsran*, es posible ejecutar la radiobase mediante el comando:

```
sudo ./build/apps/gnb/gnb -c ./configs/Archivo_de_configuracion.yml
```

Recordar que para que se inicie correctamente el *Core* debe estar en ejecución previamente. En la línea anterior el usuario deberá elegir el archivo de configuración base para la ejecución de la gNB. En la carpeta *configs* se pueden encontrar variedad de archivos, algunos generados por este proyecto, otros propios de *srsRAN Project*. Dentro de esta, se creó una carpeta *Ejemplos*, donde se encuentran ejemplos probado para la ejecución de la gNB tanto con el SDR B200 como con el SDR X310. A continuación se exponen los parámetros más importantes de los archivos de configuración de la radiobase:

- **device_driver**: El nombre del *driver* RF a utilizar. En este caso el valor asignado fue *uhd*, *drivers* utilizados para manejar los equipos X310.
- **device_args**: Este parámetros puede incluir diferentes valores asociados al SDR a utilizar. En la documentación oficial se indica que se debe ingresar modelo de SDR, de la forma *type=X310*. Sin embargo, al tener un único SDR conectado, no fue necesario especificar esto. En este caso, lo único que se estableció en esta sección fueron los tamaños de trama a enviar y recibir, indicando *send_frame_size=8000*, *recv_frame_size=8000*.
- **sync**, **clock**: Estos parámetros se utilizan para especificar la fuente de sincronización horaria y la fuente de referencia temporal del dispositivo RF respectivamente. Ambos son de vital importancia en caso de utilizar una fuente externa, lo cual es altamente recomendado por el equipo de desarrollo de *srsRAN*. En este caso, al no contar con una, se dejaron estos campos con valor *internal*.
- **srates**: Tasa de muestreo del dispositivo RF en MHz. El valor recomendado, y utilizado para el primer establecimiento de conexión fue *23.04*.
- **tx_gain**, **rx_gain**: Valores para ganancia en transmisión y recepción respectivamente, en dB. Estos deben estar entre 0 y el máximo soportado por el dispositivo RF. Su valor debió ser ajustado de forma manual hasta lograr una primera conexión, como se verá en la siguiente sección.
- **dl_arfcn**: ARFCN (*Absolute Radio-Frequency Channel Number*) de la portadora en *Downlink* (Frecuencia central). El valor del ARFCN en 5G indica el número absoluto de canal de radiofrecuencia. En el caso inicial se trabajó con el canal por defecto, de valor *368500*.
- **band**: La banda de NR en la cual se trabajará. En el caso inicial se trabajó con la banda por defecto, valor 3.
- **channel_bandwidth_MHz**: Ancho de banda del canal, en MHz. El número de PRBs se deriva automáticamente de este valor. Se trabajó inicialmente con la configuración recomendada, de valor 20 Mhz.

```

--== srsRAN gNB (commit 56e198bb9) ==--
Connecting to AMF on 10.53.1.2:38412
Available radio types: uhd.
[INFO] [UHD] linux; GNU C++ version 11.4.0; Boost_107400; UHD_4.7.0.0-84-gbdadaied
[LOGGING] Fastpath logging disabled at runtime.
Making USRP object with args 'type=b200,num_recv_frames=64,num_send_frames=64'
[INFO] [B200] Loading firmware image: /usr/local/share/uhd/images/usrp_b200_fw.hex...
[INFO] [B200] Detected device: B200
[INFO] [B200] Loading FPGA image: /usr/local/share/uhd/images/usrp_b200_fpga.bin...
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Detecting internal GPSDO....
[INFO] [GPS] No GPSDO found
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Setting master clock rate selection to 'automatic'.
[INFO] [B200] Asking for clock rate 16.000000 MHz...
[INFO] [B200] Actually got clock rate 16.000000 MHz.
[INFO] [MULTI_USRP] Setting master clock rate selection to 'manual'.
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
Cell pci=1, bw=20 MHz, IT1R, dl_arfcn=154000 (n28), dl_freq=770.0 MHz, dl_ssb_arfcn=153850, ul_freq=715.0 MHz

=== gNodeB started ===
Type <^> to view trace
t

```

		-----DL-----										-----UL-----									
pci	rnti	cqi	ri	mcs	brate	ok	nok	(%)	dl_bs	pusch	rsrp	mcs	brate	ok	nok	(%)	bsr	ta	phr		
1	460b	15	1	0	0	0	0	0%	0	n/a	n/a	0	0	0	0	0%	0	0us	n/a		
1	460b	15	1	0	0	0	0	0%	0	n/a	n/a	0	0	0	0	0%	0	0us	n/a		
1	460b	15	1	0	0	0	0	0%	0	4.7	-31.7	6	4.1k	1	0	0%	0	0us	n/a		
1	460b	15	1	0	0	0	0	0%	0	n/a	n/a	0	0	0	0	0%	0	0us	n/a		
1	460b	15	1	0	0	0	0	0%	0	n/a	n/a	0	0	0	0	0%	0	0us	n/a		

Figura A.1: Ejemplo de terminal de ejecución de gNB

- `common_scs`: Valor para el *Subcarrier Spacing*, es decir, el espaciado entre subportadoras, en kHz. Se seteó en 15, el valor sugerido inicialmente.
- `plmn`: El número de *Public Land Mobile Network* transmitido por la gNB. Se utilizó el valor por defecto “00101”.

Splits funcionales

Para ejecutar la gNB separando CU y DU, es necesario primero cambiar de la rama del repositorio. Para eso, ejecutar en terminal desde la raíz de `portaran_srsran`:

```
git checkout merge.actualizado_cu.du
```

Una vez en esa rama, crear la carpeta `build` nuevamente y repetir los pasos de la Sección A. En caso de ya haber creado la carpeta `build`, eliminarla antes. Posteriormente, una vez corriendo el `core`, se debe ejecutar primero la CU y luego la DU:

```
cd /portaran_srsran/build/apps/cu
sudo ./srscu -c ../../configs/tests_splits/nombre_de_archivo.yml
```

Dentro de la carpeta `tests_splits` están los archivos de configuración generados para cada los tres casos trabajados: ejecutados localmente en una misma PC, ejecutados en dos PCs diferentes conectados mediante Ethernet y mediante Wi-Fi. Allí el usuario podrá elegir, por ejemplo, el archivo `cu_interno.yml` y ejecutar con esa configuración. Es importante destacar que, en el caso de ejecutar CU y DU en diferentes equipos, es necesario asignar las IPs con las interfaces correspondientes, haciendo que estas esten en la misma subred. Posteriormente, para ejecutar la DU:

```
cd /portaran_srsran/build/apps/du
```

Apéndice A. Manual de instalación y configuraciones

```
sudo ./srsdu -c ../../configs/tests_splits/nombre_de_archivo.yml
```

Y allí elegir nuevamente el archivo deseado según el caso.

Equipo de usuario

En el caso del srsUE, se debe realizar un proceso similar a la ejecución del gNB implementado, pero partiendo en este caso del repositorio `srsran_4g_portaran`. Nuevamente se debe crear el código a ejecutar, generando el directorio `build`:

```
mkdir build
cd build
cmake ../
make
make test
```

De esta manera, queda creado en el directorio `build` el código necesario y se puede volver al directorio raíz del proyecto para continuar la ejecución.

Antes de ejecutar el proceso en sí, es necesario configurar ciertos parámetros en el equipo para que este pueda funcionar como UE, mediante los siguientes comandos:

- `sudo ip route add default via 10.45.1.1 dev tun_srsue`: Agrega una ruta por defecto a la dirección de la gNB
- `sudo resolvectl dns tun_srsue 8.8.8.8`: Agrega la dirección del DNS utilizado como DNS por defecto del UE, en este caso el DNS de Google 8.8.8.8

Ambos comandos se compactaron para la conveniencia del usuario en un *script* `prepare_srsUE.sh` disponible en la carpeta *scripts*, que realiza estas funciones junto con los tuneos de *performance* del *script* `srsran_performance.sh` detallados en la sección anterior.

Luego de estos pasos ya están preparadas todas las configuraciones necesarias para ejecutar el srsUE. Para esto, se llama al programa contenido en el directorio `build` desde la línea de comandos, con la dirección del archivo de configuración a utilizar. En este repositorio, se dejó una carpeta de ejemplos, por lo que una llamada a uno de estos ejemplos sería la siguiente:

```
sudo build/srsue/src/srsue Ejemplos/B210_AIR_ue_rf.conf
```

Estos archivos de configuración contienen diferentes categorías con un conjunto de parámetros a configurar, de los cuales se detallan los más relevantes a continuación:

- `[rf]`: Categoría que define los parámetros de radiofrecuencia a utilizar en el SDR
 - `freq_offset`: CFO aplicado sobre la señal, visible mediante *logs* de la ejecución

Tabla A.2: Correspondencia entre anchos de banda y cantidad de PRBs para determinar valor de los parámetros `nof_prb` y `max_nof_prb`

BW	PRBs
5	25
10	52
15	79
20	106

- `tx_gain`: Ganancia de transmisión
 - `rx_gain`: Ganancia de recepción
 - `srates`: Frecuencia de muestreo
 - `device_args`: Argumentos adicionales a pasar al SDR durante la ejecución, como tipo de sincronización o tamaño de paquete (ver Subsección 3.2.2)
- `[rat.nr]`: Parámetros de conexión en el acceso a la red NR
 - `bands`: Número de la banda utilizada
 - `nof_prb` y `max_nof_prb`: Cantidad de PRBs a utilizar, deben adaptarse al ancho de banda utilizado según la Tabla A.2 para funcionar correctamente en cada ancho de banda compatible con srsUE.
 - `dl_nr_arfcn` y `ssb_nr_arfcn`: ARFCN a utilizar para las señales en DL y UL respectivamente, visibles en la terminal al ejecutar la gNB como se observa en la Figura A.1
 - `[usim]`: Configuración del USIM simulado por srsUE
 - `opc`: OPC del móvil, correspondiente a un UE ingresado en el núcleo de red
 - `k`: Key correspondiente al par Key/OPC
 - `imsi`: IMSI configurado para ese mismo usuario, teniendo en cuenta que los primeros 5 dígitos corresponden al PLMN
 - `[log]`: Configuración de archivos de *logs*
 - `all_level`: Nivel de detalle de los *logs*
 - `filename`: Ubicación del archivo

Errores comunes y lecciones aprendidas

- Al conectar de forma inalámbrica el srsUE, lo más difícil es sintonizar correctamente las ganancias. En los archivos de configuración quedaron establecidos los valores que funcionaron, pero dependen mucho del entorno de

Apéndice A. Manual de instalación y configuraciones

despliegue. Incluso en el mismo entorno, en diferentes días pueden funcionar con ganancias levemente diferentes. Valores comunes para los SDR B200 rondaron los 70 dB en Tx y 30 dB en Rx, y para los SDR X310 generalmente funcionó al rededor de 10 dB en Tx y 5 dB en Rx.

- Los *logs*, tanto de UE como de gNB, aportan información muy útil a la hora de buscar errores de conexión. Un parámetro que hubo que sintonizar con frecuencia fue el *frequency offset*. Para ello se revisaban los logs del UE, buscando los valores de `cfo` iniciales, para compensarlos manualmente en el archivo de configuración del UE.
- Para la conexión a Internet se debió configurar correctamente el DNS en la interfaz del UE utilizada. Esto varía dependiendo del equipo. Se debe tener especial atención a como configurarlo al usar un SDR B200, que va conectado mediante USB. Allí, el comando `sudo resolvectl dns` puede ser de utilidad.
- En ocasiones, luego de un uso prolongado de los SDR, puede que estos comiencen a fallar. Esto se evidencia de varias formas, ya sea presentando *Overflows*, *Underflows* y *Lates* en la terminal o directamente acabando la ejecución. En estos casos, es conveniente desconectarlos y volverlos a conectar utilizando otra interfaz (USB o Gigabit Ethernet).

Apéndice B

Evaluación de compra de dispositivos

Una vez comenzada la etapa de exploración, se planteó la idea de experimentar con equipos nuevos que pudiesen incorporarse a la maqueta, teniendo siempre como referencia los parámetros de portabilización planteados. Inicialmente, se realizó un relevamiento orientado a Mini-PCs. Este abarcó tanto el mercado local como internacional, y se evaluaron características como precio, CPU, RAM, conectividad, fuente de energía, puertos disponibles y la posibilidad de soportar tanto srsUE como srsNB. Los datos de cada equipo se presentan en la Tabla B.1.

Si bien se evaluaron más equipos, estos fueron los que resultaron más atractivos. Otros casos resultaban en costos sobre el presupuesto o la necesidad de fuentes de energía más difíciles de transportar. Llegado este punto, se investigó qué equipos se podrían encontrar disponibles en Facultad de Ingeniería, y los candidatos fueron la *Raspberry Pi 4* y el *Intel NUC*, ambos utilizados por proyectos previos. Dado que el NUC podía llegar a officar como gNB y la *Raspberry* como UE, se optó por la incorporación de ambos al proyecto.

De todos modos, la *Raspberry Pi 4* no cuenta con conectividad 5G nativa, por lo que se procedió a buscar un *shield* o *hat* que cumpliera con este requerimiento. Se encontraron dos posibilidades en línea, una fabricada por *Waveshare Electronics* y otra por *Sixfab*. Si bien ambas ofrecían características muy similares, la segunda opción incluía un soporte físico y contaba con buenas reseñas, por lo que el equipo de trabajo se decantó por esta. Hasta la fecha, no se encontró ningún proyecto que hubiese incluido este tipo de kits en una maqueta basada en *srsRAN*.

El uso de este *shield* 5G permite la implementación de un UE sin la necesidad de utilizar SDR, acercándonos a lo que sucede en equipos comerciales. Además, su chip *Quectel* programable requirió al equipo entender cómo funcionan los equipos comerciales a bajo nivel, siendo que los equipos COTS no ofrecen al usuario la posibilidad de interactuar a bajo nivel. En el Apéndice C se presenta en profundidad el proceso de configuración.

Apéndice B. Evaluación de compra de dispositivos

Tabla B.1: Comparativa de equipos evaluados para implementación de testbed 5G portable

Características técnicas por equipo	
Raspberry Pi 4 Rol: UE Precio: USD 35 + 390 (Hat 5G) CPU: Broadcom BCM2711, Quad-core Cortex-A72 1.8 GHz Memoria: 2/4/8/16 GB Conectividad: WiFi 5, BT 5.0, Ethernet 1 Gbps Energía: 5V por USB-C o GPIO Puertos: 2x USB 3.0, 2x USB 2.0	Tinker Board S R2.0 Rol: UE Precio: USD 150 (Exterior) CPU: Quad-core ARM Cortex-A17 a 1.8 GHz Memoria: 2 GB LPDDR3 Conectividad: Ethernet Gbps, WiFi 5, BT 4.0 Energía: DC continua 5.5 V Puertos: 2x USB 3.2, 1x USB-C
Intel NUC Rol: gNB y UE Precio: USD 265 (Uruguay) CPU: Intel Celeron J4005 (4 MB cache) Memoria: Hasta 16 GB DDR4 Conectividad: WiFi 5, BT 5.0, Ethernet 1 Gbps Energía: Entrada DC continua 5.5 V Puertos: 2x HDMI 2.0, 4x USB 3.0	Tachyon 5G Rol: gNB y UE Precio: USD 200 (Exterior) CPU: Octa-core Qualcomm Kryo Memoria: 4 GB Conectividad: 5G hasta 2.5 Gbps, WiFi 6E Energía: USB-C o batería externa Puertos: GPU Qualcomm Adreno 643, USB-C

Apéndice C

Kit de desarrollo Sixfab 5G

El kit de desarrollo *Sixfab 5G* utilizado en este proyecto para la implementación de un UE es un *shield* desarrollado para la *Raspberry Pi 4*, que provee a la misma con conectividad 5G. Además del caso de red 5G SA en que fue utilizado en este proyecto, es compatible también con redes 5G NSA y LTE entre otros, y es compatible con esquemas TDD y FDD. Es desarrollado por la compañía *Sixfab*, que se especializa en soluciones IoT desde su fundación en 2016.

En el caso de este proyecto, este kit se instaló sobre una mini-PC *Raspberry Pi 4* como se ve en la Figura C.1, a la que se le instaló el sistema operativo Ubuntu 22.04.

El dispositivo *Sixfab 5G* adquirido en este caso cuenta con los siguientes elementos:

- La placa principal que contiene el procesador y controladores
- Entrada de corriente propia de DC 3.7 V por USB-C
- Entrada de pines de conexión para interfaz con *Raspberry Pi*
- Puerto USB 3.0 para conexión adicional a *Raspberry Pi*



Figura C.1: Kit Sixfab 5G usado en el laboratorio, ensamblado en una Raspberry Pi 4.

Apéndice C. Kit de desarrollo Sixfab 5G

- Adaptador USB a USB horizontal para facilitar la conexión entre ambas placas
- Lector dual de tarjeta SIM
- 4 antenas “*Sixfab LTE/5G High-Performance Antenna SMA*”, con cobertura en el rango sub-6GHz
- Carcasa para ambos dispositivos con ventilador incorporado, con conexión a placa *Sixfab*
- Tapa de acrílico transparente y tornillos

Armado del dispositivo

Para ensamblar y conectar este dispositivo a la *Raspberry Pi* utilizada se siguieron los siguientes pasos:

1. Fijar la *Raspberry Pi* al fondo de la carcasa, colocando tornillos en las ranuras designadas
2. Conectar las entradas de corriente, teclado inalámbrico y video por HDMI usados en la *Raspberry Pi*
3. Conectar el *hat Sixfab 5G* a la *Raspberry Pi*, tomando cuidado de conectar correctamente los pines a la entrada correspondiente del *hat* sin forzar o estresar demás el material de los mismos
4. Usar el adaptador USB a USB para conectar ambas placas entre sí
5. Conectar el ventilador incorporado a la carcasa a las entradas correspondientes en el *hat Sixfab*
6. Conectar las 4 antenas en las ranuras correspondientes en la parte exterior de la carcasa
7. Conectar la entrada de corriente de la placa *Sixfab*
8. Insertar un SIM en la ranura, programado previamente con valores correspondientes a un usuario registrado en el núcleo usando el proyecto *Open Cells* (ver Apéndice E)
9. Encender la *Raspberry Pi* de la manera estándar
10. Encender el *hat Sixfab* usando el interruptor presente en la parte superior de la misma
11. Opcionalmente, se puede colocar la tapa de acrílico



Figura C.2: Raspberry Pi 4, con los pines de conexión destacados en amarillo, y la sección más dañada de estos destacada en rojo.

Durante este proceso, en el caso de la *Raspberry Pi* utilizada se observó que los pines a conectar en el tercer paso estaban dañados por su uso en experiencias anteriores, con restos de soldadura y algunos pines torcidos. Se recurrió al apoyo de Horacio Duffau en el laboratorio de potencia, quien contribuyó reparando esta pieza de manera que se pudieran conectar ambas placas. En la Figura C.2 puede verse la sección particular de estos que se encontraba más dañada.

Una de las conexiones más relevantes que se encontraban en esta sección fue la utilizada para configurar el funcionamiento del ventilador incorporado al *hat* Sixfab, por lo que afortunadamente fue posible usar esta conexión correctamente para regular la temperatura de ambos dispositivos. Esto se logró usando la herramienta `gpio` de Linux para setear el valor del pin número 13 que se utilizó para indicar el encendido del ventilador.

Configuración

Para comenzar a usar el kit ensamblado, se debe instalar la herramienta de configuración *minicom* desde el repositorio *apt* de Linux. A su vez, en la documentación se advierte de desinstalar la utilidad *modemmanager* en caso de que pueda ocasionar un conflicto de *drivers*. Ambas acciones pueden realizarse con el uso de *apt*:

```
sudo apt purge modemmanager -y sudo apt install minicom -y
```

Aquí ya se puede iniciar el dispositivo con el interruptor en la parte superior, y luego iniciar la interfaz con el mismo en la PC por USB:

```
sudo minicom -D /dev/ttyUSB2 -b 115200
```

Esto inicia una terminal alternativa específica para configurar el UE, como puede verse en la Figura C.3. Esta cuenta con sus propios comandos específicos, disponibles en el manual [73]. A continuación se listan los comandos iniciales realizados en esta interfaz para configurar parámetros adicionales de conexión a la red, más allá de los ya configurados en el SIM utilizado en este caso.

- AT: Comando de inicio
- AT+QCFG="usbnet",1: Reiniciar en Modo de Control Ethernet (ECM)

Apéndice C. Kit de desarrollo Sixfab 5G

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyUSB2

Press CTRL-A Z for help on special keys

AT
OK

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB2
```

Figura C.3: Pantalla inicial de terminal del UE Sixfab, usando la herramienta *minicom*

- `AT+CGDCONT=1,"IPV4V6","srsapn"` Configurar APN de la red en el UE
- `AT+CFUN=1,1`: Habilitar todas las funcionalidades del UE, y luego reiniciar para aplicar

Luego de esto, el UE va a estar configurado para conectarse a la red que corresponde a su SIM. La única configuración pendiente es en caso de querer conectarse a Internet mediante esta red, se recomienda configurar el DNS utilizado (en este caso el DNS 8.8.8.8 de Google) en la interfaz USB correspondiente de la Raspberry Pi. Para esto se utilizó en este caso la utilidad `nmtui` de Linux en este caso ya que cuenta con interfaz gráfica, pero existen otras herramientas que logran el mismo resultado.

Para apagar el UE es necesario primero apagar el *hat* usando el interruptor antes de apagar la Raspberry Pi, evitando así diferencias de corriente entre las interfaces de los mismos que produzcan descargas no deseadas. La configuración realizada se guarda en la placa correspondiente al *hat Sixfab 5G*, por lo que en usos posteriores solo hace falta prender el dispositivo y este comenzará a buscar la conexión ya configurada.

Apéndice D

Reprogramación de SDR mediante puerto JTAG

En este apéndice se describe el proceso realizado para reprogramar el SDR Ettus X300, basado en la documentación disponible de la compañía *Ettus Research* [74].

Esto fue necesario debido a que mientras se lo utilizaba, se observó que no funcionaba correctamente durante las pruebas de rendimiento realizadas (ver Subsección 5.3.1). En estos casos el SDR dejaba de ser reconocido por la PC luego de cierto tiempo de realizar pruebas, requiriendo entonces reiniciar el sistema para volver a utilizarlo correctamente. Con el paso del tiempo, se notó que el tiempo en que el SDR era utilizable antes de que sucediera este fenómeno se iba reduciendo, y llegado cierto punto, este directamente dejó de responder. Al no funcionar incluso después de ser reiniciado, se debió proceder a su reprogramación.

Para reprogramar el SDR, la documentación indica que este se debe iniciar en modo de “recuperación” usando el *software Xilinx Vivado Design Suite* de la compañía *Advanced Micro Devices* (AMD). Dicho programa se utiliza para la programación de un amplio rango de dispositivos FPGA. Se corrió la versión 2024.2 del *software*, siendo la más reciente al momento. Esta instalación se realizó mediante un programa ejecutable, aunque también existe la posibilidad de descomprimir e instalarlo mediante la línea de comandos.

Durante la instalación, se agregó también el *driver* necesario para poder conectarse a la placa del SDR mediante el puerto JTAG, ya incluido en el paquete de instalación. Al terminar la instalación del programa *Vivado*, este descarga un *script install_digilent.sh*. Para comprobar que este *driver* fue cargado correctamente, se recargó la configuración de *drivers* del sistema con el comando

```
sudo udevadm control --reload
```

Para comenzar el proceso de reprogramación, primero se verificó la integridad de las imágenes disponibles. Estos son archivos en formato *.bit* que determinan el *hardware* simulado en la placa FPGA, como se mencionó en la Sección 2.3. En el

Apéndice D. Reprogramación de SDR mediante puerto JTAG



Figura D.1: Vista frontal del Ettus X300, con el puerto JTAG destacado en azul

caso de una instalación Linux como la usada en esta práctica, estos archivos se encuentran en el directorio `/usr/local/share/uhd`. Aquí, se verificó mediante las utilidades de la biblioteca *uhd* que estos archivos estaban correctamente actualizados a su versión más reciente.

Luego, se conectó el SDR mediante un cable JTAG, en el puerto marcado en la Figura D.1. Para poder intercambiar información con el SDR mediante este puerto, se debió también configurar la conexión al mismo en la pestaña de conexiones de red de la herramienta de configuración del sistema, generando una nueva conexión cableada “local” en la dirección 192.168.10.1. El único parámetro adicional configurado en esta conexión fue el tamaño de la *Maximum Transmission Unit* (MTU), establecido en un valor de 1500 para permitir mayor velocidad de transmisión entre la PC y el SDR.

Una vez conectado el SDR e iniciado el programa *Vivado*, los pasos realizados fueron los siguientes:

1. Abrir la ventana *Hardware Manager*.
2. Conectar al SDR utilizando la herramienta *Auto Connect*, en la ventana de herramientas.
3. Seleccionar el SDR y abrir el menú de programación *Program Device* mediante clic derecho.
4. Ingresar la nueva imagen `.bit` a programar en el SDR. En este caso, se eligió usar la imagen `usrp_x300_fpga_XG.bit`, disponible como se mencionó anteriormente en el directorio `/usr/local/share/uhd`.
5. Cuando el programa terminó de cargar esta imagen, se cerró el mismo y luego se cargó la imagen `usrp_x300_fpga_XG.bit` programada anteriormente usando la biblioteca *uhd*, mediante el comando

```
sudo uhd_image_loader --args "type=x300,addr=192.168.10.2,fpga=XG"
```



Figura D.2: Vista trasera del Ettus X300, con las entradas Ethernet de 10 Gbit/s destacadas en rojo

Se decidió utilizar la imagen XG en vez de la imagen HG comúnmente usada debido a la diferencia en como estos utilizan los dos puertos Ethernet disponibles en el SDR (visibles en la Figura D.2). En las imágenes de tipo HG, se programa el tráfico posible en estos puertos de forma que el puerto 0 tenga una capacidad de 1 Gbit/s, y el puerto 1 de 10 Gbit/s. Mientras tanto, en las imágenes de tipo XG ambos puertos están configurados con un máximo de 10 Gbit/s. Como se observó que el uso continuo de una única interfaz degradaba su funcionamiento con el tiempo, esta configuración permitiría alternar el uso de las interfaces y extender así su vida útil.

Se observó finalmente que el SDR volvía a funcionar correctamente luego de terminar el ciclo en modo de recuperación. A su vez, se pudieron utilizar ambas interfaces Ethernet de manera indistinta, por lo que este proceso se consideró exitoso.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice E

Programación de SIMs mediante herramienta OpenCells

Para la utilización de equipos COTS o el propio *Sixfab Hat 5G* para *Raspberry Pi* fue necesario contar con tarjetas SIM programables. Estas fueron heredadas del proyecto antecesor [9], donde se las adquirió al grupo *Open Cells Project* [63].

Este es un grupo cuyo fin es promover el uso de redes móviles de código abierto para investigación y educación, sin depender de soluciones propietarias, sirviendo como plataforma de aprendizaje práctico. Si bien sus tutoriales están mayormente orientados al uso de *Open Air Interface*, fue posible utilizar las SIMs programables que proveen con *srsRAN*.

Para esto fue necesario programar dichos módulos siguiendo los pasos publicados en su web [75]. Luego de descargar el software e instalar el software provisto, se conecta el lector de tarjetas SIM al computador mediante un puerto USB y desde una terminal situada en la carpeta de instalación se ejecutó el siguiente comando:

```
sudo ./program_uicc --adm 12345678 --imsi 001010000000001
--isdn 00000001 --acc 0001 --key fec86ba6eb707ed08905757b1bb44b8f
--opc C42449363BBAD02B66D116BC975D77CC1 -spn ‘‘srsapn’’
```

Allí, se pueden observar las siguientes *flags*:

- **adm**: Clave de administración de la tarjeta SIM. Desde la versión 4 de UICC, y con SIMs adquiridas a partir de 2021, se puede utilizar cualquier número de 8 dígitos ya que este parámetro no está establecido de fábrica.
- **imsi**: Valor de IMSI deseado. Es posible utilizar cualquier valor deseado, siempre que sea de 15 dígitos.
- **isdn**: Valor de la *Integrated Service Digital Network*. Se deja en 00000001 por defecto.
- **acc**: Determina la *Access Control Class*. Se deja en 0001 por defecto.

Apéndice E. Programación de SIMs mediante herramienta OpenCells

- **key**: Clave de autenticación a setear. Se utiliza el valor recomendado por el operador.
- **opc**: Clave de operador. Se utiliza el valor recomendado por el operador.
- **spn**: Nombre del proveedor de servicio.

Al programar diferentes tarjetas, es necesario variar el IMSI. Se dejaron fijos los primeros 14 dígitos, variando el último, utilizando el formato 00101000000000X. Cada uno debió ser dado de alta en la interfaz de administración de *Open5GS* para su correcta autenticación.

Estos módulos se pueden reconfigurar las veces que sea necesario. A modo de verificación, es posible ver la información presente en la tarjeta ejecutando:

```
sudo DEBUG=y ./program_uicc.
```

Glosario

G.1. Acrónimos

3GPP	3rd Generation Partnership Project
5G NR	5th Generation New Radio
5GC	5G Core
ABR	Available Bitrate
AKMA	Authentication and Key Management for Applications
AM	Amplitude Modulation
AMC	Adaptive Modulation and Coding
AMD	Advanced Micro Devices
AMF	Access and Mobility Management Function
APN	Access Point Name
APT	Advanced Package Tool
ARFCN	Absolute Radio Frequency Channel Number
BBU	Baseband Unit
BTS	Base Transceiver Station
BW	Bandwidth
Br	Bitrate
CFO	Carrier Frequency Offset
CLI	Command Line Interface
COTS	Commercial Off-The-Shelf
CP	Control Plane

Apéndice E. Programación de SIMs mediante herramienta OpenCells

CPF	Core Plane Function
CPU	Central Processing Unit
CQI	Channel Quality Indicator
CSV	Comma-Separated Values
CU	Central Unit
CUPS	Control and User Plane Separation
CoMP	Coordinated Multipoint
dB	Decibeles
DCI	Downlink Control Information
DFTs	Discrete Fourier Transforms
DL	Downlink
DNS	Domain Name System
DSP	Digital Signal Processing
DU	Distributed Unit
eMBB	Enhanced Mobile Broadband
ECM	Ethernet Control Model
eNodeB	Evolved Node B
EPC	Evolved Packet Core
eSIM	Electronic Subscriber Identity Module
F1AP	F1 Application Protocol
FDD	Frequency Division Duplexing
FM	Frequency Modulation
FPGA	Field-Programmable Gate Array
FR	Frequency Range
FWA	Fixed Wireless Access
gBE	Gigabit Ethernet
gNB	Next Generation Node B
GSM	Global System for Mobile Communications

HARQ	Hybrid Automatic Repeat reQuest
HD	High Definition
HDL	Hardware Description Language
HTTP	HyperText Transfer Protocol
Hz	Hertz
IMSI	International Mobile Subscriber Identity
IMT	International Mobile Telecommunications
IoT	Internet of Things
ITU	International Telecommunication Union
JTAG	Joint Test Action Group
LTS	Long Term Support
LTE	Long Term Evolution
MAC	Medium Access Control
MAQ5G	Maqueta 5G (nombre de proyecto antecesor)
MCC	Mobile Country Code
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
mMTC	Massive Machine Type Communications
MNC	Mobile Network Code
MS	Mobile Subscriber
MSC	Mobile Switching Center
MTU	Maximum Transmission Unit
NFV	Network Functions Virtualization
NSA	Non-Standalone
NSSF	Network Slice Selection Function
NUC	Next Unit of Computing
O-RAN	Open Random Access Network
OAI	OpenAirInterface

Apéndice E. Programación de SIMs mediante herramienta OpenCells

OFDM	Orthogonal Frequency Division Multiplexing
OH	Overhead
OPc	Operator Code (combined with Key)
OSA	Open Systems Architecture
PAPR	Peak to Average Power Ratio
PBCH	Physical Broadcast Channel
PC	Personal Computer
PCDP	Physical Channel Data Processing
PCF	Policy Control Function
PDCCH	Physical Downlink Control Channel
PDCP	Packet Data Convergence Protocol
PDSCH	Physical Downlink Shared Channel
PHY	Physical Layer
PLMN	Public Land Mobile Network
PRB	Physical Resource Block
PUSCH	Physical Uplink Shared Channel
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RAC	Radio Access Capability
RAM	Random Access Memory
RAN	Radio Access Network
RACH	Random Access Channel
RF	Radio Frequency
RRC	Radio Resource Control
RU	Radio Unit
RX	Receiver
SA	Stand Alone

SBA	Service-Based Architecture
SCA	Software Communications Architecture
SCS	Subcarrier Spacing
SDAP	Service Data Adaptation Protocol
SDN	Software Defined Networking
SDR	Software Defined Radio
SIM	Subscriber Identity Module
SNR	Signal to Noise Ratio
SMF	Session Management Function
SRS	Sounding Reference Signal
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TR	Technical Report (ej. 3GPP TR 38.801)
TX	Transmitter
UBX	Universal Baseband eXtension
UCI	Uplink Control Information
UDM	Unified Data Management
UDP	User Datagram Protocol
UE	User Equipment
UHD	USRP Hardware Drivers
UL	Uplink
UPF	User Plane Function
URLLC	Ultra Reliable Low Latency Communications
URSEC	Unidad Reguladora de Servicios de Comunicaciones (Uruguay)
USIM	Universal Subscriber Identity Module
USRP	Universal Software Radio Peripheral
V2X	Vehicle-to-Everything
ZMQ	ZeroMQ (librería de mensajería)

Esta página ha sido intencionalmente dejada en blanco.

Referencias

- [1] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "On challenges of sixth-generation (6g) wireless networks: A comprehensive survey of requirements, applications, and security issues," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 6–36, 2021.
- [2] J. I. Mitola, *Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering*. Wiley-Interscience, 2000.
- [3] F. Larroca, P. Flores-Guridi, D. Artenstein, L. Inglés, and G. Morales, "Ultra-cheap sdr digital television transmission: Isdb-t with an osmo-fl2k and an rtl-sdr," in *Proceedings of the 10th GNU Radio Conference, 2020*. [Online]. Available: <https://github.com/git-artes/gr-isdbt>
- [4] G. Belcredi, M. Randall, C. Rattaro, and P. Belzarena, "Satellite and aircraft communications through sdr as an introduction to telecommunications and electrical engineering," in *2020 IEEE Global Engineering Education Conference (EDUCON)*. Montevideo, Uruguay: IEEE, 2020, available online: Facultad de Ingeniería, Universidad de la República. [Online]. Available: <https://iie.fing.edu.uy/publicaciones/2020/BRRB20/BRRB20.pdf>
- [5] M. Pirri and R. García, "Implementación de un transmisor ais de bajo costo basado en sdr," Proyecto de Fin de Carrera, 2024, disponible en línea en IIE-Fing.
- [6] srsRAN Project, "srsran: Open source 4g and 5g radio access network," <https://www.srsran.com>, 2024, accessed: 2024-12-15.
- [7] OpenAirInterface Software Alliance, "Openairinterface: 5g software alliance for democratising wireless innovation," <https://www.openairinterface.org>, 2024, accessed: 2024-12-15.
- [8] O-RAN Alliance, "O-RAN Specifications," <https://specifications.o-ran.org/specifications>, Último acceso: 2 de abril de 2025.
- [9] W. Peña, W. Piastri, and P. Vázquez, "Implementación de una maqueta de pruebas y desarrollo de una red 5g stand alone completa," Master's thesis, Facultad de Ingeniería, Universidad de la República (UdelaR), Montevideo, Uruguay, Diciembre 2023.

Referencias

- [10] R. P. Alves, J. G. A. da S. Alves, M. R. Camelo, W. O. de Feitosa, and V. F. Monteiro, “Experimental comparison of 5g sdr platforms: srsran x openairinterface,” Master’s thesis, Federal University of Ceará (UFC), Fortaleza, Ceará, Brazil, 2024.
- [11] L. Mamushiane, A. Lysko, H. Kobo, and J. Mwangama, “Deploying a stable 5g sa testbed using srsran and open5gs: Ue integration and troubleshooting towards network slicing,” in *2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, 2023, pp. 1–10.
- [12] T. Mukute, L. Mamushiane, A. A. Lysko, E.-R. Modroiu, T. Magedanz, and J. Mwangama, “Control plane performance benchmarking and feature analysis of popular open-source 5g core networks: Openairinterface, open5gs, and free5gc,” *IEEE Access*, vol. 12, pp. 113 336–113 360, 2024.
- [13] M. Amini and C. Rosenberg, “Performance analysis and comparison of full-fledged 5g standalone experimental tdd testbeds in single & multi-ue scenarios,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.02341>
- [14] J. Garcia, A. Prasad, H. Tan, K. Rao, P. Khurana, and H. Schulzrinne, “Multi-ue 5g ran measurements: A gamut of architectural options,” in *Proceedings of the 24th International Workshop on Mobile Computing Systems and Applications (HotMobile ’23)*. ACM, 2023, pp. 73–79. [Online]. Available: <https://doi.org/10.1145/3572864.3580307>
- [15] L. M. P. Larsen, A. Checko, and H. L. Christiansen, “A survey of the functional splits proposed for 5g mobile crosshaul networks,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 146–172, 2019.
- [16] S. S. R. Jonnavithula, I. K. Jain, and D. Bharadia, “Beamarmor5g: Demonstrating mimo anti-jamming and localization with srsran 5g stack,” in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1817–1819. [Online]. Available: <https://doi.org/10.1145/3636534.3698870>
- [17] S. Nesmachnow and P. Moreno-Bernal, Eds., *Actas de las Jornadas Uruguayas de Ciencias de la Computación 2024*. Universidad de la República, Uruguay: Serie Reportes Técnicos InCo-PEDECIBA, 2025, disponible en: <https://iie.fing.edu.uy/publicaciones/2024/JUCC2024/>.
- [18] Facundo Pedreira y Bruno Tió, “Gitlab PortaRAN,” https://gitlab.fing.edu.uy/portaran/portaran_srsran.
- [19] Facundo Pedreira y Bruno Tió , “Gitlab srsRAN 4G PortaRAN,” https://gitlab.fing.edu.uy/portaran/srsran_4g_portaran.

- [20] 3GPP, “Release 8 - 3rd generation partnership project,” 3GPP, Tech. Rep., 2008. [Online]. Available: <https://www.3gpp.org/specifications-technologies/releases/release-8>
- [21] F. Larroca and P. Belzarena, *Comunicaciones Inalámbricas*. Instituto de Ingeniería Eléctrica - Facultad de Ingeniería - Udelar, 2017. [Online]. Available: <https://iie.fing.edu.uy/investigacion/grupos/artes/wp-content/uploads/sites/13/2017/10/principal.pdf>
- [22] Ericsson, “Ericsson mobility report q4 2024 update,” Ericsson, Tech. Rep., February 2024. [Online]. Available: <https://www.ericsson.com/492af1/assets/local/reports-papers/mobility-report/documents/2024/ericsson-mobility-report-q4-2024-update.pdf>
- [23] “Study paper on 5g core network,” Telecommunication Engineering Centre (TEC), Khurshid Lal Bhawan, Janpath, New Delhi - 110001, India, Tech. Rep., 2021, available at: https://www.tec.gov.in/public/pdf/Studypaper/5G%20Core%20Network_Study%20Paper_v8.pdf.
- [24] International Telecommunication Union (ITU), “Imt-2020: Framework and overall objectives of the future development of imt for 2020 and beyond,” ITU, Tech. Rep. ITU-R M.2083-0, 2020. [Online]. Available: <https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2020/Pages/default.aspx>
- [25] I. Futura, “Las comunicaciones 5g en el internet de las cosas,” <https://iotfutura.com/2018/10/iot-y-comunicaciones-5g/>, 2018.
- [26] J. García-Morales, M. C. Lucas-Estañ, and J. Gozalvez, “Latency-sensitive 5g ran slicing for industry 4.0,” *IEEE Access*, vol. 7, pp. 143 139–143 159, 2019.
- [27] Q. W. Solutions, “How fixed wireless access is erasing the digital divide,” Quectel Wireless Solutions, Tech. Rep., January 2024. [Online]. Available: <https://www.quectel.com/library/fixed-wireless-access-white-paper/>
- [28] 3GPP Official Site. [Online]. Available: <https://www.3gpp.org/technologies/5g-system-overview>
- [29] Microsoft, “What is the 5g service-based architecture (sba)?” <https://techcommunity.microsoft.com/t5/telecommunications-industry-blog/what-is-the-5g-service-based-architecture-sba/ba-p/3831367>, 2024, consultado el 1 abril de 2025.
- [30] Misha, “¿cuál es la diferencia entre nsa y sa? ¿cuál es mejor?” 2021, consultado el 1 de abril de 2025. [Online]. Available: <https://blog.router-switch.com/2021/08/what-is-the-difference-between-nsa-and-sa-which-is-better/>
- [31] ShareTechnote, “5g network slicing,” 2024. [Online]. Available: https://www.sharetechnote.com/html/5G/5G_NetworkSlicing.html

Referencias

- [32] S. Lee, “Quickstart,” <https://open5gs.org/open5gs/docs/guide/01-quickstart/>, 2025, consultado el 17 de Enero de 2025.
- [33] Motorola, “Adaptive modulation and coding (amc),” 3GPP TSG-RAN WG1, Tech. Rep. R1-00-1395, October 2000, tSG-RAN WG1 Meeting #17, Stockholm, Sweden, 20th-24th Oct 2000. [Online]. Available: https://www.3gpp.org/ftp/tsg_ran/wg1_rl1/TSGR1_17/Docs/PDFs/R1-00-1395.pdf
- [34] ShareTechnote, “Numerology / scs (subcarrier spacing) in detail,” consultado el 2 de abril de 2025. [Online]. Available: https://www.sharetechnote.com/html/5G/5G_Phy_Numerology.html
- [35] J. Wannstrom, “Carrier Aggregation explained,” <https://www.3gpp.org/technologies/101-carrier-aggregation-explained>, 2013, consultado del 3 de febrero de 2025.
- [36] K. Sthankiya, N. Saeed, G. McSorley, M. Jaber, and R. Clegg, “A survey on ai-driven energy optimization in terrestrial next generation radio access networks,” *IEEE Access*, vol. PP, pp. 1–1, 01 2024.
- [37] 3rd Generation Partnership Project (3GPP), “3gpp ts 38.801: Study on new radio access technology: Radio access architecture and interfaces,” 3GPP, Tech. Rep., 2016, available online: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3056>.
- [38] Presidencia de la República Oriental del Uruguay, “Ute alcanzó instalación de medidores inteligentes en 80 % de los hogares uruguayos,” Feb. 2024. [Online]. Available: <https://www.gub.uy/presidencia/comunicacion/noticias/ute-alcanzo-instalacion-medidores-inteligentes-80-hogares-uruguayos>
- [39] ShareTechnote, “5g/nr - initial attach,” consultado el 3 de abril de 2025. [Online]. Available: https://www.sharetechnote.com/html/5G/5G_CallProcess_InitialAttach.html
- [40] S. Nair, S. Khare, and J. Ping, “Authentication and key management for applications (akma) in 5g,” 3GPP, 2022. [Online]. Available: <https://www.3gpp.org/technologies/akma>
- [41] 3rd Generation Partnership Project (3GPP), “3gpp ts 38.211: Physical channels and modulation,” 3GPP, Tech. Rep., 2017, available online: https://www.3gpp.org/ftp/Specs/archive/38_series/38.211.
- [42] “ETSI TS 138 213 V16.2.0 (2020-07): 5G; NR; Physical layer procedures for control (3GPP TS 38.213 version 16.2.0 Release 16),” ETSI, Tech. Rep. ETSI TS 138 213 V16.2.0, 2020, especificación técnica. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138200_138299/138213/16.02.00_60/ts_138213v160200p.pdf
- [43] S. R. Systems, “About srs: Who we are,” <https://srs.io/about-us/>, 2024, accessed: 2025-3-29.

- [44] Documentación Oficial de srsRAN Project. [Online]. Available: <https://docs.srsran.com/projects/project/en/latest/#>
- [45] 3rd Generation Partnership Project (3GPP), “(3gpp tr 21.915 version 15.0.0 release 15: Digital cellular telecommunications system (phase 2+) (gsm); universal mobile telecommunications system (umts);” 3GPP, Tech. Rep., 2019, available online: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3389>.
- [46] F. J. Zumegen, I. K. Jain, and D. Bharadia, “Beamarmor demo: Anti-jamming system in cellular networks with srsran software radios,” in *MILCOM 2023 - 2023 IEEE Military Communications Conference (MILCOM)*, 2023, pp. 245–246.
- [47] G. Lando, L. Augusto, F. Schierholt, M. P. Milesi, and J. A. Wickboldt, “Evaluating the performance of open source software implementations of the 5g network core,” Master’s thesis, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Rio Grande do Sul, Brazil, 2023.
- [48] C. S. Choudhari, R. Patil, and S. Saraf, “Deployment of 5g core for 5g private networks,” in *2022 International Conference on Industry 4.0 Technology (I4Tech)*, 2022, pp. 1–6.
- [49] F. Dolente, R. G. Garroppo, and M. Pagano, “A vulnerability assessment of open-source implementations of fifth-generation core network functions,” *Multidisciplinary Digital Publishing Institute (MDPI)*, 2023.
- [50] X. Cai, M. Zhou, and X. Huang, “Model-based design for software defined radio on an fpga,” *IEEE Access*, vol. 5, pp. 8276–8283, 2017.
- [51] T. J. Kacpura, W. M. Eddy, and J. L. Carl R. Smith, “Software defined radio architecture contributions to next generation space communications,” *National Aeronautics and Space Administration - IEEE*, 2015.
- [52] C. Bergstrom, S. Chuprun, S. Gifford, and G. Maalouli, “Software defined radio (sdr) special military applications,” in *MILCOM 2002. Proceedings*, vol. 1, 2002, pp. 383–388 vol.1.
- [53] R. Lackey and D. Upmal, “Speakeasy: the military software radio,” *IEEE Communications Magazine*, vol. 33, no. 5, pp. 56–61, 1995.
- [54] *Ettus Research, USRP X300 and X310 X Series Reference Sheet*. [Online]. Available: <https://www.ettus.com/all-products/x310-kit/>
- [55] *Ettus Research, USRP B200 and B210 X Series Reference Sheet*. [Online]. Available: <https://www.ettus.com/all-products/ub200-kit/>
- [56] Docker Inc., “Docker compose documentation,” <https://docs.docker.com/compose/>, 2024.

Referencias

- [57] Github Oficial srsRAN Project. [Online]. Available: https://github.com/srsran/srsRAN_Project
- [58] srsRAN Project, “O-ran gnb functional split in srsran 5g,” 2024, accedido el 11 de abril de 2025. [Online]. Available: https://docs.srsran.com/projects/project/en/latest/knowledge_base/source/oran_gnb/source/index.html
- [59] B. Khan, Nidhi, H. OdetAlla, A. Flizikowski, A. Mihovska, J.-F. Wagen, and F. J. Velez, “Survey on 5g second phase ran architectures and functional splits,” *IEEE Communications Standards Magazine*, 2023, submitted/preprint, affiliation info from image. [Online]. Available: https://www.researchgate.net/publication/364556951_Survey_on_5G_Second_Phase_RAN_Architectures_and_Functional_splits
- [60] Github srsRAN Forums, “Real-time failure in RF,” https://github.com/srsran/srsRAN_Project/issues/1141.
- [61] Benetel, “RAN550 - 5G RU Indoor,” <https://benetel.com/ran550/>, 2024.
- [62] TP-Link Technologies Co., Ltd., “TL-WR1042ND - Router inalámbrico N Gigabit de 300Mbps,” 2024. [Online]. Available: <https://www.tp-link.com/es/home-networking/wifi-router/tl-wr1042nd/>
- [63] Open Cells Project. [Online]. Available: <https://open-cells.com/>
- [64] 3rd Generation Partnership Project (3GPP), “3gpp ts 38.306: Nr user equipment (ue) radio access capabilities,” 3GPP, Tech. Rep., 2020, available online: https://www.3gpp.org/ftp/Specs/archive/38_series/38.306.
- [65] Spectrum Tracker. [Online]. Available: <https://www.spectrum-tracker.com/Uruguay>
- [66] T. Electronics, “Optimizing peak to average power ratio for wireless systems,” in *MicroWave Journal*, 2020.
- [67] N. Aliane, “A survey of open-source uav autopilots,” *Electronics*, vol. 13, p. 4785, 12 2024.
- [68] P. Brach Del Prever, “Cellular networks support for real-time control of uavs,” Ph.D. dissertation, Politecnico di Torino, 2022.
- [69] Github srsRAN Forums, “CU/DU Split Deployment 827,” https://github.com/srsran/srsRAN_Project/discussions/827.
- [70] “Ettus research binary installation.” [Online]. Available: https://files-ettus-com.translate.goog/manual/page_install.html?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- [71] Github Oficial srsRAN 4G. [Online]. Available: https://github.com/srsran/srsRAN_4G

- [72] Guía de Instalación Oficial de srsRAN Project. [Online]. Available: <https://docs.srsran.com/projects/project/en/latest/user-manuals/source/installation.html>
- [73] L. Quectel Wireless Solutions Co., *RG50xQRM5xxQ Series AT Commands Manual*.
- [74] Nate Temple, Michael Dickens, Ettus Research, “X300/X310 Device Recovery,” https://kb.ettus.com/X300/X310_Device_Recovery, Último acceso: 24 de abril de 2025.
- [75] Open Cells Project - UICC Programming. [Online]. Available: <https://open-cells.com/index.php/uiccsim-programing/>

Esta página ha sido intencionalmente dejada en blanco.

Índice de tablas

1.1.	“Implementación de una Maqueta de Pruebas y Desarrollo de una Red 5G Stand Alone Completa” [9].	6
1.2.	“Experimental Comparison of 5G SDR Platforms: srsRAN x OpenAirInterface” [10].	6
1.3.	“Deploying a Stable 5G SA Testbed using srsRAN and Open5GS: UE Integration and Troubleshooting towards Network Slicing” [11].	7
2.1.	Especificaciones relevantes del USRP X310 [54].	27
2.2.	Especificaciones relevantes del USRP B200 [55].	28
3.1.	Variantes en <i>hardware</i> y <i>software</i> de radiobase y costos asociados.	48
3.2.	Variantes en <i>hardware</i> y <i>software</i> de equipo de usuario y costos asociados.	48
4.1.	Resultados de pruebas de ganancia por interfaz de aire utilizando SDR USRP X310.	52
4.2.	Valores de los parámetros afectados al modificar el <i>Bandwidth</i>	53
4.3.	Resultados de pruebas de ancho de banda por interfaz de aire utilizando SDR USRP X310.	53
4.4.	Resultados de pruebas en banda n3.	55
4.5.	Resultados en bandas asignadas a proveedores.	55
4.6.	Valores de ganancias utilizados según la distancia.	56
4.7.	<i>Bitrates</i> según la distancia en 60 segundos de transmisión, utilizando SDR USRP X310.	56
4.8.	Ganancias utilizadas para las conexiones con SDR USRP B200.	57
4.9.	Resultados de pruebas con SDR USRP B200.	57
4.10.	Resultados de <i>data rate</i> DL y UL para COTS UE y Sixfab en diferentes <i>Bandwidths</i>	58
4.11.	Resultados de <i>throughput</i> con múltiples equipos de usuario en simultáneo.	60
4.12.	<i>Throughput</i> relevado en las diferentes interfaces <i>F1AP</i>	61
4.13.	Resultados del Speedtest para diferentes dispositivos y anchos de banda.	62
4.14.	Resultados de pruebas cualitativas.	63
5.1.	Comparativa entre <i>srsRAN</i> y <i>OpenAirInterface</i>	66

Índice de tablas

A.1. Parámetros de acceso frecuente configurados en el archivo <code>open5gs.env</code> .	87
A.2. Correspondencia entre anchos de banda y cantidad de PRBs para determinar valor de los parámetros <code>nof_prb</code> y <code>max_nof_prb</code>	91
B.1. Comparativa de equipos evaluados para implementación de testbed 5G portable	94

Índice de figuras

2.1.	Tráfico de datos de la red móvil global de Ericsson y crecimiento interanual (EB por mes). Imagen adaptada de [22].	12
2.2.	Áreas de aplicación de redes 5G y ejemplos de servicios. Imagen obtenida de [25].	13
2.3.	Visión general de arquitectura de red. Presenta componentes, interfaces y tipo de conexión. Obtenida de [28].	14
2.4.	Implementación SA vs NSA. Imagen adaptada de [30].	16
2.5.	Secciones de una radiobase 5G (gNB). En gris se enmarcan las diferentes componentes. Imagen obtenida de [36].	17
2.6.	Posibles <i>splits</i> en una arquitectura 5G NR. Imágen obtenida de 5G Technology World.	19
2.7.	Proceso de registro de un UE en la red 5G, adaptado de [39].	20
2.8.	Variación del ancho de banda para diferentes numerologías.	22
2.9.	<i>Resource Blocks</i> e impacto de <i>Subcarrier Spacing</i> en tamaño de <i>subframes</i>	23
2.10.	Diagrama de desarrollo del proyecto srsRAN a la fecha, incluyendo funcionalidades previstas para versiones futuras, en base a notas de versiones en [44].	24
2.11.	Cambios en arquitectura 5G implementados en <i>srsRAN</i> 5G según <i>Release 15</i> , vía documentación de <i>srsRAN</i> [44].	25
3.1.	Panel de control de <i>Grafana</i> : enmarcado en rojo las estadísticas de tráfico total por la gNB, en verde estadísticas de tráfico en <i>downlink</i> con cada UE individual diferenciado por color, y en azul estadísticas de tráfico en <i>uplink</i> nuevamente diferenciado por UE.	30
3.2.	Separación CU/DU/RU implementada por <i>srsRAN</i> en srsgNB. Se detalla los protocolos delegados a cada unidad. Obtenida de [58].	31
3.3.	Diagrama de subredes configuradas en entorno virtualizado en una misma PC, con radiobase y 3 UEs conectados por interfaces locales levantadas mediante la librería ZMQ.	34
3.4.	Interfaz del core <i>Open5GS</i> , con un UE registrado.	34
3.5.	Diagrama de interconexión al trabajar con dos SDR X310, conectados por cable coaxial.	37
3.6.	Interconexión de dos SDR X310 mediante cables coaxiales. En azul se circulan los atenuadores de 30 dB en transmisión.	38

Índice de figuras

3.7. Maqueta final con SDRs USRP X310, interconectados por aire. Se señalan en la imagen los servidores y SDR, cada uno con su función. Sobre los SDR, en color verde, se pueden observar las antenas direccionales.	39
3.8. Captura de pantalla de comando <i>htop</i> en NUC corriendo srsDU a 20 MHz de ancho de banda. Se destaca el uso de CPU (en verde) y RAM (cantidad total residente en memoria en amarillo, y porcentaje de la memoria total en rojo).	41
3.9. Esquema de Red con separación CU/DU en un mismo servidor. . .	42
3.10. Esquema de Red con separación CU/DU conectada mediante cable Ethernet.	43
3.11. Esquema de Red con separación CU/DU conectada mediante Wi-Fi.	44
3.12. Imagen de DU en NUC, utilizando un SDR USRP B200.	45
3.13. Configuración de APN en COTS UE.	46
3.14. Kit de desarrollo <i>Sixfab 5G</i> , instalado en <i>Raspberry Pi 4</i>	47
4.1. Esquema de red con conexión entre SDRs por interfaz de aire, usada para pruebas de rendimiento.	50
4.2. Espectro de banda <i>n3</i> , centrado en 1842 Mhz.	54
5.1. Salida del comando <i>ru_metrics</i> en gNB.	67
5.2. Gráfico de barras con valores obtenidos y teóricos para diferentes anchos de banda utilizando dos SDR USRP X310 (Tx y Rx). . . .	69
5.3. Comparativa del valor teórico alcanzado por cada solución.	70
5.4. Comparación de <i>bitrate</i> obtenido según la banda de operación. . .	72
5.5. Laboratorio en que se realizaron las pruebas, en este caso se representa la prueba de distancias máxima, señalando la ubicación de los SDR.	73
5.6. Software <i>htop</i> mostrando uso de recursos durante ejecución de gNB y Core.	74
5.7. Comparación de <i>bitrate</i> DL para srsUE, COTS UE y Sixfab, alcanzados en pruebas internas en la red.	76
5.8. Comparativa de <i>Downlink</i> en <i>speedtests</i>	77
5.9. Comparación entre arquitecturas evaluadas en Split 8, según Subsección 4.2.2.	79
A.1. Ejemplo de terminal de ejecución de gNB	89
C.1. Kit Sixfab 5G usado en el laboratorio, ensamblado en una Raspberry Pi 4.	95
C.2. Raspberry Pi 4, con los pines de conexión destacados en amarillo, y la sección más dañada de estos destacada en rojo.	97
C.3. Pantalla inicial de terminal del UE Sixfab, usando la herramienta <i>minicom</i>	98
D.1. Vista frontal del Ettus X300, con el puerto JTAG destacado en azul	100

D.2. Vista trasera del Ettus X300, con las entradas Ethernet de 10 Gbit/s
destacadas en rojo 101

Esta es la última página.
Compilado el viernes 8 agosto, 2025.
<http://iie.fing.edu.uy/>